Fast and Energy-Efficient Watermarking of HEVC-Compressed Video Bitstreams

Jakub Stankowski
Institute of Multimedia Telecommunications
Poznań University of Technology
Poznań, Poland
ORCID: 0000-0002-5105-5090

Marek Domański
Institute of Multimedia Telecommunications
Poznań University of Technology
Poznań, Poland
ORCID: 0000-0002-9381-0293

Tomasz Grajek
Institute of Multimedia Telecommunications
Poznań University of Technology
Poznań, Poland
ORCID: 0000-0002-7142-9416

***Abstract*— Most research results in video watermarking refer to the embedding of messages into uncompressed video, which needs full decoding and then full encoding of the content. In some applications, such a method is too complex, and watermarking of compressed bitstreams is required. Watermarking of compressed bitstreams is more difficult as the number of bits is very limited, and the compressed video is very sensitive to all modifications that may yield visible artifacts in the decoded watermarked video. In the paper, an original watermarking method is described for HEVC-compressed video. The main idea of the technique is to embed a message into nonzero residual transform coefficients in such a way that potential drift is limited. The watermark detection is based on reference, unwatermarked video. The experimental verification demonstrates the strong robustness of the embedded watermarks.**

***Keywords*— robust watermarking, HEVC, transform-domain watermarking, bitstream watermarking**

I. Introduction

We are witnessing continuous development in video compression methods. About every 9-10 years new compression standard is released by ISO/IEC [1]. Typically video compression standards do not have support for functionalities such as authentification, copyright protection, and fingerprinting. Therefore, whenever a new standard is deployed on the market, a great effort is made to develop the above-mentioned functionalities. They are very crucial because with improvements in flat-panel displays and Internet streaming platforms, creating high-quality copies of video data is getting easier and easier every year.

One of the most popular solutions for illegal copy protection is watermarking [2, 3]. Generally, watermarks may be visible or invisible, fragile or robust. For illegal copy protection, a watermark should be robust (it has to survive after strong image processing including camcording and recompression) and invisible/unnoticeable by potential viewers [4-7].

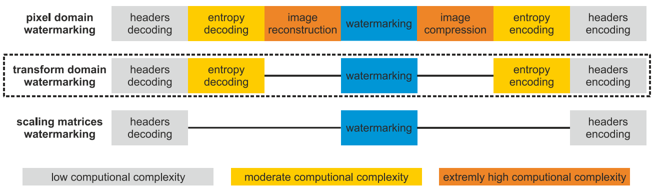

Fig. 1. Comparison of most significant classes of compressed domain watermarking. The method proposed in this paper belongs to the transform domain watermarking class [5].

In order to survive after recompression and/or camcording, a watermark should be hidden in image samples. Generally, three main paths to modify image samples according to watermark may be distinguished (Fig. 1): pixel domain, transform domain, and scaling matrices domain. The scaling matrices domain is the less complex approach because only entropy decoding and re-encoding are needed. However, it allows only for very weak control of the watermark embedding process as there is no direct access to image samples. At the other end, we have the pixel domain watermarking. This is the most complex approach (full decoding and re-encoding are needed), but it offers full control of the watermark embedding process including drift compensation. Between these two mentioned approaches, the transform domain watermarking is placed. It is characterized by moderate complexity (partial decoding and re-encoding are needed) and gives indirect access to image samples through transform coefficients [2, 3].

This paper focuses on HEVC (MPEG-H part 2 and H.265) [8], as it is observed to replace AVC (MPEH-4 Part 10 and H.264) [9]. The HEVC watermarking is not new but is still a very active research area [6, 7, 10-35]. Mainly because most of the methods designed for previous video compression standards cannot be directly applied to HEVC, except that requiring full decoding and reencoding (i.e. pixel domain watermarking). Moreover, most of the proposed methods are fragile against full decoding and further processing of video samples, so they will be useless to track illegal camcording and distribution on the internet.

II. General Idea

The idea of the described solution is to propose a technology that allows hiding some watermark data in HEVC-compressed bitstream. The user of the technology should have the possibility to mark each stream with a dedicated watermark. Therefore, the watermark embedding should have relatively low computational complexity, and memory bandwidth requirements and generate low energy consumption. This leads us to only one reasonable approach which is the transform domain watermarking (Fig. 1). Moreover, the authors previously developed a set of watermarking solutions for AVC [9] based on scaling matrices watermarking [4] and transform domain watermarking [5].

In the proposed approach the watermarking technique is treated as a way of creating a reliable communication channel hidden in video content that is robust against common attacks. The general idea of the proposed system has been shown in Fig. 2.

In general, the watermark is embedded by altering some of the low-frequency DCT coefficients transmitted as residual data. This leads to modification of reconstructed frame frequency characteristics i.e. a slight increase of energy of selected components. These changes are unnoticeable for the viewer but can be detected and decoded into watermark bits.

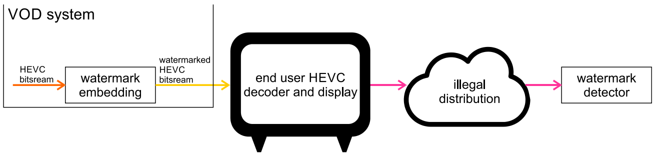

Fig. 2. The general idea and use case of the proposed watermarking system.

III. Description of the Proposed Method

A. Frame selection

The transform domain watermarking allows embedding watermark into any frame. However, there are several factors rendering the watermarking of all frames inefficient.

The first reason is computational complexity. As measured and described in [36], the I frames are the biggest in the bitstream i.e. in a 32-picture GOP (Group of Pictures) containing only one I frame, the I frame takes about 30% of a bitstream. The remaining 31 frames are 70% which makes them more suitable for processing. Since the B and P frames are smaller (and entropy encoding and decoding are significant factors in watermarking computational complexity) the processing of P and B frames only allows for reducing latency introduced by watermarking process.

The second factor is drift control. Any change in the I frame propagates to almost all frames in GOP since the I frame is the main source of motion compensated prediction. The same happens to P and B frames at the beginning of the GOP. The propagation of watermark thru motion-compensated prediction increases distortion and can cause visible artifacts. This leads to the idea of processing only the second half of any GOP in order to reduce drift. Moreover, the availability of non-watermarked pictures (1st half of the GOP) allows to model the influence of watermark attacks and helps in increasing the reliability of watermark detection.

The examples of picture selection in the proposed method are presented in Fig. 3.

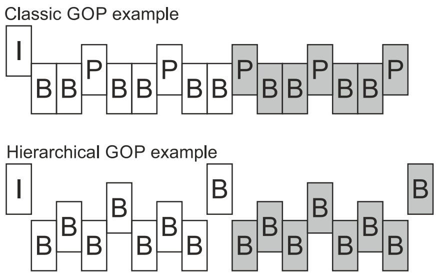

Fig. 3. Visualization of picture selection for classic and hierarchical GOP. Pictures marked as grey are processed during watermark embedding.

B. Distortion-optimized selection of TUs

In HEVC [8], the P and B frames can contain areas encoded using intra and inter prediction. As shown in [36], the B slices in HEVC bitstream contain a noticeable amount of bits representing intra-prediction modes. This leads to the conclusion that even in B frames, a significant part of PUs is encoded in intra mode. Unfortunately, the presence of intra-coded areas makes watermark embedding much more difficult. The intra-coded blocks use the edge of previously encoded blocks as a prediction source. In the case of embedding leading to alternate edge pixels, the intra prediction protracts the distorted values over the entire block. This leads to emerge of errors looking like streaks or dump patches. To make things worse, these distorted areas tend to flicker which makes them noticeable and annoying to viewers. Therefore, significant effort has to be taken to avoid the distortion introduced by intra-coded Prediction Units (PU).

The most efficient way of avoiding the above-mentioned errors is to keep intra-prediction sources unaltered. This leads to the development of a dedicated drift-optimized embedding algorithm. The proposed algorithm is designed to reduce computational complexity. At the first step, some rough check is made and more compute-heavy steps are taken only if necessary.

The watermarked picture is processed at the Coding Unit (CU) level. In the first step, all intra-encoded CUs are excluded from embedding. In the next step, inter-encoded CUs are examined. For every CU the neighboring blocks are identified. The intra prediction uses below-left, left, above-left, above and above-right neighboring unit edges as prediction signals. Therefore, for each watermarked CU, below, below-right, and right neighboring CUs have to be identified. If non of mentioned neighboring CUs is intra-encoded, the processed CU can be watermarked without any precautions.

In the opposite situation, the embedder checks if there is a risk of introduction of intra-related distortion. In the first step, the intra-prediction mode of all neighboring (below, below-right and right) intra-encoded CUs are checked. The information about the intra-prediction mode is then used to estimate if the currently processed block is a prediction source for neighboring units. For example: if a right neighbor of the currently processed block uses intra mode 34 (so-called right-top, see [37]) the currently processed block will not be a prediction source and the change of edge pixel values does not influence the right neighbor CU. If the investigation of all neighboring units intra mode leads to the conclusion that there is no risk of drift introduction, the processed CU can be also watermarked.

In the case where both above-mentioned paths are not available, the watermark embedding process is continued at the Transform Unit (TU) level. The inter-coded CU can have one or more TUs containing prediction errors. In the case of CU divided into more than one TU, the TUs are processed as follows: If TU does not belong to the right or bottom edge of CU, the watermark is embedded. If TU contacts the right or bottom edge, the previously described procedure is used to determine if the alternation of samples covered by TU can lead to intra-related drift. The decision of watermarking processed TU is made based on the calculated risk of drift introduction.

C. Sign hiding

In HEVC the new coding tool called Sign Data Hiding (SDH) was introduced [38]. The idea of SDH is to reduce the number of transmitted transform coefficient signs. In general, quantized transform coefficients are separated into sign and module and both sets of values are transmitted separately. Since the signs are unpredictable, SDH tries to hide one sign value in parity of transmitted coefficient amplitudes. This tool allows significantly increased compression efficiency. However, it reduces the flexibility and introduces some restrictions on transmitted coefficient amplitudes.

When SDH is enabled the sign of the last nonzero coefficient is embedded in the parity of the sum of the levels of the Coefficient Group (CG) [38]. Therefore, the parity of the sum of the levels of the CG has to be preserved during the watermarking process.

D. *Coefficient selection and watermark strength consideration*

As mentioned in Section II the proposed method reuses the idea of alternating the low-frequency AC components. In [5] the rationale of this approach was described. In HEVC [8] four transform sizes were defined: 4x4, 8x8, 16x16, and 32x32. All transforms used in inter-coded blocks are DCT based. In order to preserve watermark consistency, the same range of spatial frequencies has been chosen to be alternated. In the case of the 4x4 transform, this means the lowest AC coefficients with indexes (0,1) and (1,0). Similarly, for 8x8 transform the corresponding coefficients are (0,2), (0,3), (2,0) and (3,0). Coefficients for 16x16 and 32x32 transforms have been selected following the same approach.

During the watermark embedding process, the selected AC coefficient is modified to change the distribution of spatial frequencies in a decoded picture. The modification is done by multiplying the quantized transform coefficient by the watermark strength (*ws*) factor. The multiplication result has to be rounded (up or down) since quantized transform coefficients are represented as integer values. Moreover, the value of the modified coefficient needs to be adjusted in order to preserve sign information hidden in the sum of the levels of the Coefficient Group (CG).

The *ws* factor can be used to control the strength of the watermark. The higher the *ws* value, the stronger the watermark is, but at the cost of higher distortion. During the development of the described algorithm, we decided to evaluate different values of *ws* and check its influence on watermark robustness and image quality degradation.

E. *Watermark detection*

The watermark detection uses the method described in [4]. The watermark detector operates in the pixel domain and requires two signals the reference one (original, non-watermarked video) and investigated one. Detector processes every sequence extracting per frame data about the energy of spatial frequencies within the frame. Those statistics are rather compact (several values per frame) and can be stored instead of the original video.

The detector averages the spectral energy statistics for images from a watermarked group and images from non-watermarked groups surrounding the watermarked one. In the next step, detection concentrates on spatial energy related to coefficients modified during the watermarking process. The difference between the energy of watermarked and non-watermarked parts of GOP in the investigated sequence is calculated and normalized by the same difference in the reference sequence. The increase of energy corresponding to selected horizontal or vertical frequencies is interpreted as the appearance of 0 or 1 in watermarked signal.

IV. EXPERIMENTAL EVALUATION OF THE PROPOSED METHOD

The proposed method has been evaluated experimentally. To perform the evaluation, a set of test sequences was prepared. The watermarking process was performed by software designed and developed by the authors. The software includes HEVC entropy decoder/encoder (VLC for headers and CABAC for slice data) and watermarking module processing the entropy decoded data. The software does not perform image reconstruction (decoding) so image pixels were not used at any stage of watermarking process. The embedding software has been developed to evaluate the proposed method and cannot be considered optimized in any way, regardless of the throughput of ~130Mbit/s was achieved using a single core of consumer-grade CPU.

A. *Evaluation framework*

The proposed method has been evaluated using a set of 6 diversified video sequences. Each sequence has a FullHD (1080p) resolution and a length of 12-15 minutes. The entire sequence has been used during the evaluation. The used sequences can be described as: anim1 and anim2 - computer-animated adventure comedy films; epic1 – vintage, epic, historical drama; soap1 and soap2 – soap opera; war1 – modern war movie. Due to the low density of embedded watermark, we can not use typical test sequences recommended by the MPEG group since these sequences are too short (up to 12 seconds). The watermark strength (*ws*) multiplier values from set {1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8 3.0} have been used.

The experiment has been performed starting from an uncompressed sequence. In the first step, HEVC compressed bitstreams were prepared using the x265 [39] encoder. The x265 has been set to produce 8Mbit/s bitrate, operate using a "veryslow" preset, and use PSNR as a quality metric for rate-distortion optimization. The chosen x265 parameters guaranteed high compression efficiency and very good quality. The quality (measured as luma PSNR) and precise bitrate of each encoded sequence have been measured.

In the next step, all previously encoded sequences have been watermarked using all previously mentioned watermark strength values. Moreover, for each sequence and each *ws* value, a set of four random messages was used. As a result, we prepared a set of 240 watermarked bitstreams (corresponding to about 45 hours of video material).

Since the proposed method aims to create a robust watermark we decided to perform some attacks and measure the watermark's robustness. Experiments described in [4] and [5] show that one of the most difficult scenarios is when content is displayed on a TV and recorded by a camera (or camcorder). In such a case the decoded watermarked video is a subject of very strong distortion caused by: post-processing performed by TV (i.e. additional deblocking, contrast amplification, etc), geometrical distortion, dynamic range lost during capturing by a camera, camera side pre-processing, and lossy compression. All these sources of distortion lead to the creation of a scenario that is very demanding for robust watermarking. Unfortunately, the high amount of video material makes it very difficult to perform experiments using content recorded from TV by a camcorder. Therefore, we decided to simulate such an attack. The test recordings and the simulation framework based on subjective evaluation were prepared. The simulation framework aimed to mimic the distortion caused by the TV-to-camera path. The simulation includes: a strong deblocking filter (in 8x8 regions), slight image rotation, dynamic range clip to 0.1-0.9 of the original range, dynamic range expansion (contrast enhancement), HEVC encoding with ½ of original bitrate with the usage of "fast" preset.

All watermarked and attacked sequences have been processed by a watermark detector. The robustness of the watermark has been evaluated by measuring the percent of correctly recovered bits. Moreover, the watermarked sequences have been decoded and their quality was measured (PSNR) to evaluate the influence of the watermarking process on image quality. The entire evaluation framework has been summarized in Fig. 4.

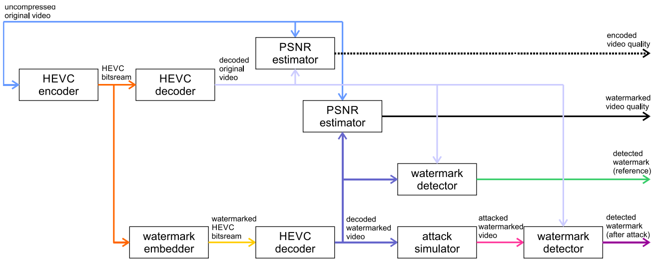

Fig. 4. Visualization of experimental evaluation workflow.

As a result watermark robustness was tested against two scenarios:

- Scenario1 – corresponding to screen-captured content (i.e. VODRip): HEVC reencoding with ½ bitrate and "fast" preset,
- Scenario2 – corresponding to camera recorded content (i.e. CamRip/Telecine) uses the previously mentioned attack simulation framework.

B. *Evaluation results*

The watermark robustness evaluation results were summarized in TABLE I. The results are averaged over all four messages. For the Scenario1 the *ws* values as low as 1.6 provide acceptable robustness. If camera recording is considered (Scenario2), the required *ws* has to be increased to the 2.2-2.4 range.

TABLE I. DETECTION RESULTS (AS A PERCENT OF CORRECTLY DETECTED BITS)

	Correctly detected bits for Scenario 1[%]									
	Watermark strength									
Sequence	***1.2***	***1.4***	***1.6***	***1.8***	***2.0***	***2.2***	***2.4***	***2.6***	***2.8***	***3.0***
anim1	53.7	77.5	92.7	93.8	92.6	97.9	98.0	98.0	98.0	98.1
anim2	52.5	72.9	89.9	90.5	89.3	97.6	97.7	98.5	98.3	97.7
epic1	56.4	81.0	98.7	98.5	98.2	99.8	99.8	99.8	99.8	99.8
soap1	52.8	70.0	95.0	96.0	94.5	99.9	99.9	99.9	99.9	99.9
soap2	54.4	78.5	95.4	96.1	94.9	99.9	99.9	99.9	99.9	99.9
war1	53.3	67.3	84.2	85.0	83.9	95.1	95.1	95.6	95.6	95.6
Average	53.9	74.5	92.6	93.3	92.2	98.4	98.4	98.6	98.6	98.5

	Correctly detected bits for Scenario 2[%]									
	Watermark strength									
Sequence	***1.2***	***1.4***	***1.6***	***1.8***	***2.0***	***2.2***	***2.4***	***2.6***	***2.8***	***3.0***
anim1	50.3	60.8	77.0	79.0	76.7	92.9	92.9	93.8	93.8	93.8
anim2	50.9	61.0	76.2	77.7	75.9	91.5	91.4	92.5	92.5	92.4
epic1	52.1	71.3	97.1	97.0	97.2	99.5	99.5	99.6	99.5	99.5
soap1	51.2	58.3	73.9	75.9	73.2	96.6	96.7	97.5	97.7	97.6
soap2	51.4	57.3	68.9	71.1	68.5	95.4	95.2	96.4	96.6	96.7
war1	50.7	57.6	71.0	73.3	70.6	90.2	90.3	91.1	91.3	91.3
Average	51.1	61.0	77.3	79.0	77.0	94.3	94.3	95.1	95.2	95.2

The TABLE II. and TABLE III. summarize the bitrate and quality changes caused by watermarking process.

Assuming that *ws* =1.6 and *ws*=1.8 are the most suitable for Scenario1, the watermarking-related bitrate increase is negligible (~0.3%) and can be neglected. The quality degradation is about 0.3dB which can be considered very low.

For Scenario2 the most suitable watermark strengths are *ws*=2.2 and *ws*=2.4. In this case, the watermarking process increases the resulting bitrate by ~1.8% and causes ~1.1dB quality loss. The bitrate increase is not significant and quality loss seems to be an acceptable tradeoff for camera recording robustness.

TABLE II. RELATIVE BITRATE INCREASE

	Bitrate increase [%]									
	Watermark strength									
Sequence	***1.2***	***1.4***	***1.6***	***1.8***	***2.0***	***2.2***	***2.4***	***2.6***	***2.8***	***3.0***
anim1	0.005	0.064	0.272	0.298	0.253	1.567	1.564	1.728	1.745	1.765
anim2	0.003	0.057	0.285	0.305	0.265	1.579	1.577	1.745	1.765	1.779
epic1	0.004	0.060	0.394	0.415	0.373	3.187	3.185	3.404	3.424	3.440
soap1	0.005	0.061	0.277	0.300	0.255	1.709	1.706	1.880	1.901	1.919
soap2	0.006	0.077	0.312	0.342	0.285	1.635	1.630	1.829	1.854	1.879
war1	0.004	0.042	0.201	0.217	0.188	1.064	1.062	1.184	1.196	1.210
Average	0.004	0.060	0.290	0.313	0.270	1.790	1.787	1.962	1.981	1.999

TABLE III. WATERMARKING INFLUENCE INTO IMAGE QUALITY

	Decoded image quality change (delta PSNR) [dB]									
	Watermark strength									
Sequence	***1.2***	***1.4***	***1.6***	***1.8***	***2.0***	***2.2***	***2.4***	***2.6***	***2.8***	***3.0***
anim1	-0.011	-0.088	-0.235	-0.319	-0.262	-1.008	-1.039	-1.330	-1.422	-1.524
anim2	-0.006	-0.069	-0.210	-0.277	-0.227	-0.909	-0.936	-1.225	-1.317	-1.406
epic1	-0.009	-0.091	-0.322	-0.404	-0.342	-1.630	-1.656	-1.999	-2.087	-2.168
soap1	-0.011	-0.088	-0.251	-0.331	-0.270	-1.083	-1.114	-1.418	-1.511	-1.595
soap2	-0.013	-0.111	-0.303	-0.413	-0.335	-1.174	-1.215	-1.590	-1.712	-1.842
war1	-0.009	-0.066	-0.182	-0.247	-0.201	-0.722	-0.748	-0.980	-1.047	-1.135
Average	-0.010	-0.085	-0.251	-0.332	-0.273	-1.087	-1.118	-1.424	-1.516	-1.612

C. *Objective quality of watermarked sequences*

The huge amount of test materials rendered the subjective evaluation very difficult. Requiring viewers to evaluate 45h of video materials is almost impossible. Therefore, we decided to perform a simplified subjective evaluation.

The subjective test participants were unaware of the type of artifacts introduced by watermarking process. Viewers were asked to watch one sequence of each type for different watermark strengths (1.6, 1.8, 2.2, 2.4) and decide if they see any noticeable distortion or annoying artifacts. No artifacts were noticed and most participants considered differently watermarked versions of one sequence as repetition of the same content.

V. CONCLUSIONS

In the paper, an original watermarking method is described for HEVC-compressed video. The main idea of the technique is to embed a message into nonzero transform coefficients in such a way that potential drift is limited. The complexity of the proposed solution is significantly lower than pixel domain watermarking (so it is a relatively fast approach), leading to huge energy savings by omitting full decoding and re-encoding. The experimental verification has been performed on six test sequences for different watermark strengths against two strong attacks modeling modifications caused by typical techniques used to create illegal copies of videos. The experimental verification demonstrates the strong robustness of the embedded watermarks with unnoticeable quality degradation.

ACKNOWLEDGEMENT

The research was supported by the Ministry of Education and Science of the Republic of Poland under the subvention for research SBAD.

REFERENCES

[1] D. Karwowski, T. Grajek, K. Klimaszewski, O. Stankiewicz, J. Stankowski, K. Wegner, "20 Years of Progress in Video Compression - from MPEG-1 to MPEG-H HEVC. General View on the Path of Video Coding Development," Book: Image Processing and Communications Challenges 8, Editors: Ryszard S. Choras, Advances in Intelligent Systems and Computing: No. 525, Springer International, pp. 3-15, ISBN: 978-3-319-47273-7, ISSN: 2194-5375, 2016.

[2] G. Doërr, J.-L. Dugelay: "A guide tour of video watermarking", Signal Processing: Image Communication vol. 18, no. 4, pp.263-282, Elsevier 2003.

[3] F. Hartung, B. Girod: "Watermarking of Uncompressed and Compressed Video", Signal Processing, vol. 66, no. 3, pp. 283-301, Elsevier, 1998.

[4] J. Stankowski, T. Grajek, M. Domański, "Fast watermarking of MPEG-4 AVC/H.264 encoded HDTV video bitstreams", 29th Picture Coding Symposium, PCS 2012, Kraków, Poland, 7-9 May 2012, pp. 265-268.

[5] J. Siast, J. Stankowski, T. Grajek, M. Domański, "Digital Watermarking with Local Strength Adjustment for AVC-Compressed HDTV Bitstreams", 30th Picture Coding Symposium, PCS 2013, San Jose, CA, USA, 8-11 December 2013, pp. 53-56.

[6] Y. Tew, K. Wong, R. C.-W. Phan, and K. N. Ngan, "Separable authentication in encrypted HEVC video," in Multimedia Tools and Applications, pp. 1–20, 2018.

[7] A. A. Elrowayati, M. F. L. Abdullah, A. A. Manaf, and A. S. Alfagi, "Tampering detection of double-compression with the same quantization parameter in HEVC video streams," in 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), pp. 174–179, 2017.

[8] ISO/IEC 23008-2, MPEG-H Part 2, High-Efficiency Video Coding, Edition 4, 2020, also: ITU-T Rec. H.265 Edition 8.0, 2021.

[9] ISO/IEC 14496-10, MPEG-4 Part 10, Generic coding of audio-visual objects, Advanced Video Coding. Edition 9, 2020, also: ITU-T Rec. H.264 Edition 14.0, 2021.

[10] T. Dutta and H. P. Gupta, "An efficient framework for compressed domain watermarking in P-frames of high-efficiency video coding (HEVC)–encoded video," ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 13, no. 1, p. 12, 2017.

[11] P.-C. Chang, K.-L. Chung, J.-J. Chen, C.-H. Lin, and T.-J. Lin, "A DCT/DST-based error propagation-free data hiding algorithm for HEVC intra-coded frames," Journal of Visual Communication and Image Representation, vol. 25, no. 2, pp. 239–253, 2014.

[12] S. Swati, K. Hayat, and Z. Shahid, "A watermarking scheme for high efficiency video coding (HEVC)," PloS one, vol. 9, no. 8, pp. 1–8, 2014.

[13] S. Gaj, A. Sur, and P. K. Bora, "A robust watermarking scheme against re-compression attack for H.265/HEVC," in Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), pp. 1–4, 2015.

[14] K. Ogawa and G. Ohtake, "Watermarking for HEVC/H.265 stream," in IEEE International Conference on Consumer Electronics (ICCE), pp. 102–103, 2015.

[15] Y. Tew, K. Wong, and R. C.-W. Phan, "HEVC video authentication using data embedding technique," in IEEE International Conference on Image Processing (ICIP), pp. 1265–1269, 2015.

[16] T. Dutta and H. P. Gupta, "A robust watermarking framework for high efficiency video coding (HEVC)–encoded video with blind extraction process," Journal of Visual Communication and Image Representation, vol. 38, pp. 29–44, 2016.

[17] A. A. Elrowayati, M. F. L. Abdullah, A. A. Manaf, and A. S. Alfagi, "Robust HEVC video watermarking scheme based on repetition-BCH syndrome code," International Journal of Software Engineering and Its Applications, vol. 10, no. 1, pp. 263–270, 2016.

[18] Y. Tew, K. Wong, R. C.-W. Phan, and K. N. Ngan, "Multi-layer authentication scheme for HEVC video based on embedded statistics," Journal of Visual Communication and Image Representation, vol. 40, pp. 502–515, 2016.

[19] S. Gaj, A. Kanetkar, A. Sur, and P. K. Bora, "Drift-compensated robust watermarking algorithm for H.265/HEVC video stream," ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 13, no. 1, pp. 1–24, 2017.

[20] G. Jiang, J. Wang, M. Yu, and F. Chen, "Zero-watermarking registration and detection method for HEVC video streaming against requantization transcoding," U.S. Patent 9,947,065.

[21] K. Joa, W. Lei, and Z. Lic, "A watermarking method by modifying QTCs for HEVC," Computing, Performance and Communication systems, pp. 8–16, 2018.

[22] G. Kaur, S. S. Kasana, and M. Sharma, "An efficient watermarking scheme for enhanced high efficiency video coding/H.265," Multimedia Tools and Applications, pp. 1–23, 2018.

[23] 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.

[24] A. A. Mohammed and N. A. Ali, "Robust video watermarking scheme using high efficiency video coding attack," Multimedia Tools and Applications, vol. 77, no. 2, pp. 2791–2806, 2018.

[25] A. Joshi, V. Jain, S. Ladda, and R. Kumar, "Real-time implementation of blind and robust watermarking for HEVC video coding," in 2018 IEEE International Symposium on Smart Electronic Systems (iSES), pp. 58–63, 2018.

[26] C. Wang, R. Shan, and X. Zhou, "Anti-HEVC recompression video watermarking algorithm based on the all phase biorthogonal transform and SVD," IETE Technical Review, vol. 35, no. sup1, pp. 42–58, 2018.

[27] X. Yu, C. Wang, and X. Zhou, "A hybrid transforms-based robust video zero-watermarking algorithm for resisting high efficiency video coding compression," IEEE Access, vol. 7, pp. 115 708–115 724, 2019.

[28] T. Shanableh, "Data embedding in HEVC video by modifying the partitioning of coding units," IET Image Processing, vol. 13, no. 11, pp. 1909–1913, 2019.

[29] G. Kaur, S. S. Kasana, and M. Sharma, "An efficient watermarking scheme for enhanced high efficiency video coding/H.265," Multimedia Tools and Applications, vol. 78, no. 9, pp. 12 537–12 559, 2019.

[30] G. Kaur, S. S. Kasana, and M. Sharma, "An efficient authentication scheme for high efficiency video coding/H.265," Multimedia Tools and Applications, vol. 78, no. 15, pp. 21 245–21 271, 2019.

[31] D. R. Galiano, A. Del Barrio, G. Botella, and D. Cuesta, "Efficient embedding and retrieval of information for high-resolution videos coded with HEVC," Computers & Electrical Engineering, vol. 81, p. 106541, 2020.

[32] M. N. Favorskaya and V. V. Buryachenko, "Authentication and copyright protection of videos under transmitting specifications," in Computer Vision in Advanced Control Systems-5, pp. 119–160, 2020.

[33] M. Z. Konyar, O. Akbulut, and S. Öztürk, "Matrix encoding-based high-capacity and high-fidelity reversible data hiding in HEVC," Signal, Image and Video Processing, pp. 1–9, 2020.

[34] S. Gaj, A. Sur, and P. K. Bora, "Prediction mode based H.265/HEVC video watermarking resisting re-compression attack," Multimedia Tools and Applications, pp. 1–31, 2020.

[35] 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.

[36] J. Stankowski, D. Karwowski, T. Grajek, K. Wegner, J. Siast, K. Klimaszewski, O. Stankiewicz, M. Domański, "Bitrate Distribution of Syntax Elements in the HEVC Encoded Video", International Conference on Signals and Electronic Systems, ICSES 2014, Poznań, Poland, 11-13 September 2014, ISBN: 978-1-4799-7009-4.

[37] J. Lainema, F. Bossen, W. Han, J. Min and K. Ugur, "Intra Coding of the HEVC Standard," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1792-1801, Dec. 2012, doi: 10.1109/TCSVT.2012.2221525.

[38] J. Sole *et al*., "Transform Coefficient Coding in HEVC," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1765-1777, Dec. 2012, doi: 10.1109/TCSVT.2012.2223055.

[39] x265 HEVC Encoder / H.265 Video Codec, https://www.x265.org/