Out-of-The-Loop Information Hiding for HEVC Video

Luong Pham Van, Johan De Praeter, Glenn Van Wallendael, Jan De Cock, and Rik Van de Walle

In: 2015 International Conference on Image Processing (ICIP), 3610 - 3614, 2015.

To refer to or to cite this work, please use the citation to the published version:

Pham Van, L., De Praeter, J., Van Wallendael, G., De Cock, J., and Van de Walle, R. (2015). Out-of-The-Loop Information Hiding for HEVC Video. *2015 International Conference on Image Processing (ICIP)* 3610 - 3614.

OUT-OF-THE-LOOP INFORMATION HIDING FOR HEVC VIDEO

Luong Pham Van, Johan De Praeter, Glenn Van Wallendael, Jan De Cock, and Rik Van de Walle

Ghent University - iMinds - Multimedia Lab, Ghent, Belgium

ABSTRACT

Communication using internet and digital media is more and more popular. Therefore, the security and privacy of data transmission are highly demanded. One effective technique providing this requirement is information hiding. This technique allows to conceal secret information into a video file, an audio, or a picture. In this paper, we propose a low complexity out-of-the-loop information hiding algorithm for a video pre-encoded with the high efficiency video coding standard. Only selected components such as the motion vector difference and transform coefficients of the video are extracted and modified, bypassing the need of fully decoding and re-encoding the video. In order to reduce the propagation error caused by hiding information, the dependency between video frames is taken into account when distributing the information over the frame. Several embedding strategies are investigated. The experimental results show that the information should be hidden in smaller blocks to reduce quality loss. Using a smart distribution of information across the frames can keep the quality loss under 1 dB PSNR for an information payload of 15 kbps. When such a strategy is used, embedding information in the transform coefficients only slightly outperforms the modification of motion vector differences.

Index Terms— Data Hiding, High Efficiency Video Coding, Motion Vector Difference, DCT coefficients.

1. INTRODUCTION

With the current technology, people can easily and flexibly communicate through internet and digital media. Therefore, transmission of digital data needs to be more secure, especial for banking and military information. One effective solution is information hiding which conceals the secret message into video. In general, the information is embedded to the video without changing its perceptual quality. Therefore, only the sender and receiver can realize the existence of the information in video.

During the last decade, many information hiding algorithms have been proposed for existing video coding standards (e.g. MPEG, H.264/AVC). These techniques usually map the input information to a component of the video such as the discrete cosine transform (DCT) coefficients [1, 2], motion vectors [3], and intra prediction mode [4, 5]. However, to the best of our knowledge, there has been little focus on watermarking for the recently finalized High Efficiency Video Coding (HEVC) standard [6]. In [7], the information is hidden in the coding block size during the encoding loop of the HEVC encoder. Although this technique prevents propagation errors, the complexity is high due to the decoding and re-encoding step needed to embed the information into a pre-encoded video bit stream. Additionally, a specialized encoder is needed for the embedding. If unique information needs to be inserted into multiple copies of the video, such an approach would require the computationally expensive encoding step to be executed multiple times. As such, this approach does not scale well. Since embedding the information during the encoding loop becomes infeasible, the information should be inserted directly into the bit stream, outside the encoding loop.

In this paper, we propose a low-complexity out-of-the-loop technique for information hiding by inserting the data into a pre-encoded HEVC video stream without fully decoding and encoding. Instead, only a low-complexity entropy decoding and encoding is required to access and modify selected bit stream components (DCT coefficients, motion vector differences). Additionally, the propagation error of inserting information is analyzed. To decrease this error, the information is distributed over the different frames based on the inter-prediction dependencies between frames.

The rest of the paper is organized as follows. Section 2 briefly reviews the coding modes in HEVC. In Section 3, the proposed information hiding technique for pre-encoded video streams is elaborated on. The experimental results and analysis are presented in Section 4. Finally, conclusions are addressed in Section 5.

2. HEVC CODING STRUCTURE

The HEVC standard supports large coding block sizes with a flexible partitioning scheme. The biggest block (typically 64x64 pixels), known as a coding tree unit (CTU), is recursively split into smaller coding units (CUs) [8]. This CU partitioning process is performed for CUs from depth 0 (CTU) to depth 3 (8x8 pixel CU). Each CU is further split into the prediction units (PU) for inter- and intra-prediction, and trans-

---

The activities described in this paper were funded by Ghent University, iMinds, the Agency for Innovation by Science & Technology (IWT), the Fund for Scientific Research (FWO Flanders), and the European Union, and were carried out using the Stevin Supercomputer Infrastructure at Ghent University.

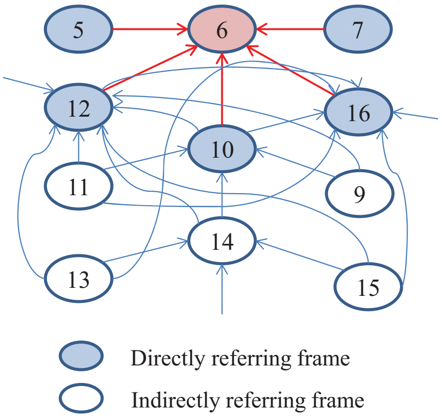

Fig. 1. A part of the reference map of inter-coded frames using the random access configuration.

form unit (TU) for residual coding. For each out of eight possible PU partitioning modes, best block matching is performed to find the best matching block of the current PU in reference frames. This process results in a prediction error (residual) and a motion vector of the PU. The difference between this motion vector and the motion vector of a neighbouring encoded PU is encoded. The prediction error of the CU is transformed to the DCT domain by using a squared Residual Quad-Tree (RQT). The RQT is evaluated from depth 0 (32x32 pixels) to depth 3 (4x4 pixels). These transform coefficients are then quantized and entropy encoded.

3. PROPOSED TECHNIQUES

In the proposed techniques, the information is hidden in the compressed domain, so without a full decoder-encoder loop. To achieve this, the syntax elements of the video stream are modified to include the input information. To modify the syntax elements, only a low-complex entropy decoding and encoding needs to be performed contrary to full decoding and encoding. The main challenge is thus to determine the optimal type and amount of bit stream elements to modify. Since a bit stream contains many motion vectors and DCT coefficients, this paper investigates the performance of hiding information in these venues.

Since modifying the syntax elements of the bit stream causes a mismatch in coding information between encoder and decoder, errors are propagated throughout the video. Therefore, the following section proposes a selection strategy to select the distribution of information across the blocks and frames in the bit stream. Thereafter, the techniques to modify the motion vectors and DCT coefficients are described.

3.1. Selection of blocks to hide information

The selection of blocks which contain the hidden information depends on the amount of the information that needs to be hidden. This selection criterion works on two levels: the frame level and the intra-period level.

At the frame level, the visual quality loss of an individual frame should be minimized when adding information. Due to the characteristics of the human visual system, the quality loss caused by the blocking artefacts resulting from hiding the input information is more visible in smooth areas while it is hard to detect in complex areas. These smooth areas are often encoded by using big blocks. In contrast, complex areas are coded using small blocks. Therefore, to minimize the visual quality loss, information should be hidden in small blocks within a frame. The bigger block sizes are thus only considered when the amount of added information is high.

At the intra-period level, the error propagation between frames should be minimized. This error propagation is caused by adding information to frames that are used by other frames as reference pictures for inter-prediction. E.g., the errors introduced in frame A by adding information may propagate to another frame B when frame B relies on frame A for inter-prediction. As such, frame B is a referring to frame A, whereas frame A is referred by frame B.

The influence on error propagation of a frame is measured by the number of other frames directly or indirectly referring to this frame. These dependencies are determined by the coding structure of the video. For instance, HEVC supports hierarchical prediction, which means that frames can be classified according to different levels of dependencies. In this prediction structure, intra frames are independently encoded and are referred by inter frames. An inter frame between two successive intra frame can be referred by one or more other frames. Therefore, any errors in this frame can propagate to referring frames. Finally, some frames are not referred by any other frame. Consequently, the errors in these frame do not affect the other frames.

In Fig. 1, a part of the referring map of frames between two successive intra frames encoded using the random access configuration [9] is drawn. The intra-period is 32 and the size of the group of pictures (GOP) is 8. As seen in this figure, frame 6 has five directly referring frames. It also has several indirectly referring (by the way of frame 10, 12, and 16) frames. On the other hand, the odd-numbered frames have no referring frames.

Before adding information to a video stream, the error propagation influence of each frame is evaluated. Using this influence, a frame is classified into a high, medium or low influence layer. The hidden information is allocated differently for each layer: more information is added to frames in low influence layers whereas less information is added to frames in high influence layers. Within the same layer, the embedded information is equally distributed over the frames.

3.2. Information embedding

The information embedding process uses the odd-even criterion [10]. The modified value of the syntax where the information hidden is odd if the input bit is 1. Otherwise, this value is even. If x is the original value of the syntax element (e.g.

Table 1. Frame classification in dependency layers.

Layer	Frame(number of referring frames)
L0	8(16), 16(16), 24(14)
L1	4(11), 6(9), 12(11), 20(11), 28(9)
L2	The others

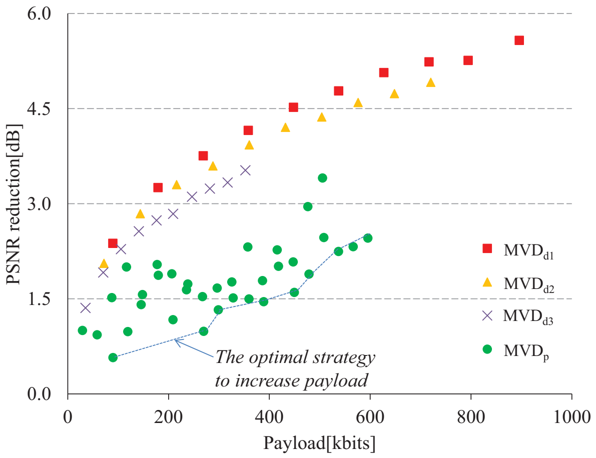

Fig. 2. Visual quality lost and information payload when modifying motion vector differences of videos.

motion vectors, or non-zero DCT coefficients), and *w* is the input bit, then the modified value *x'* is obtained as:

$$x' = sgn(x) * \lfloor |x|/2 \rfloor * 2 + w \qquad \text{with} \quad x \neq 0$$

Additionally, if *x'* equals 0, no information is hidden to ensure that all information can be detected in the decoder.

4. EXPERIMENTAL RESULTS

The experiments evaluate the performance of the proposed information embedding techniques in terms of information capacity and visual quality loss. Moreover, a comparison between hiding information in transformed coefficients and motion vector difference is made.

A total of 23 sequences with a playtime of 10 seconds have been used to test the information hiding algorithms [9]. Of these, 21 sequences have input resolutions varying from 416x240 up to 1920x1080 pixels while the two sequences Traffic and PeopleOnStreet have a resolution of 3840x2048 pixels. These sequences are first encoded using the HEVC reference software HM 16 [11]. Evaluation is based on the random access configuration (RA). The intra period is set to 32 such that stream switching or error recovery can be provided. The quantization parameter is selected from the following set $\{22, 27, 32, 37\}$. Thereafter, the proposed solution embeds random information in the motion vector difference or the transform coefficients. The modified bit stream is then reconstructed using a normal decoder. The PSNR of the reconstructed video is obtained using the original video as the reference. The PSNR reduction is calculated by subtracting the PSNR of an unmodified stream with the PSNR of a modified stream which includes the embedded information.

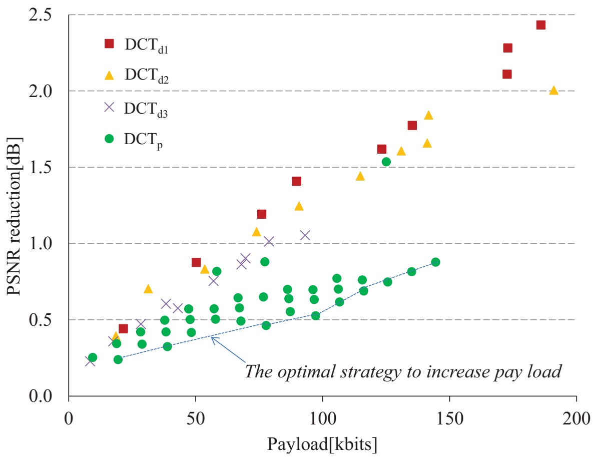

Fig. 3. Visual quality lost and information payload when modifying DCT coefficients of videos.

4.1. Payload and PSNR reduction analysis

This experiment measures PSNR reduction when the amount of input information increases. Three different embedding strategies have been evaluated: modification of motion vector differences in CUs at depth 3 (MVD_{d3}), or CUs at depth 3 and depth 2 (MVD_{d2}),or CUs at both depth 3, 2, and 1 ($MVDd1$). Similarly, DCT coefficients are modified for only TUs with size 4x4 (DCT_{d3}) or TUs with size 4x4 and 8x8 (DCT_{d2}), or TUs with sizes 4x4, 8x8, and 16x16 (DCT_{d1}). The amount of added information is increased from 10% to 100% of all available motion vectors or DCT coefficients in steps of 10% in every inter-coded frame.

On the other hand, the intra-period level distribution strategy explained in Section 3 is carried out (MVD_p and DCT_p). Using this scheme, the inter-coded frames are classified into three layers based on the number of referring frames as shown in Table 1. When the distance in picture order count between two frames is larger than 10, the influence between them is considered as 0, since blocks are much more likely to refer to closer frames. No input information is not hidden in layer L0. In layer L1, only blocks at depth 3 and depth 2 are used for embedding information. Blocks at depth 3, depth 2, and depth 1 are used in layer L2. Thirty-six combinations are tested by varying the amount of added information in both layer L1 and L2 from 0% to 100% of all available motion vectors or DCT coefficient in steps of 20%. The experimental results are shown in Fig. 2 and Fig. 3. Three important conclusions are drawn from these figures.

Firstly, adding information to small blocks performs better than adding it to larger blocks. With the same amount of embedded information, adding information to a block at depth

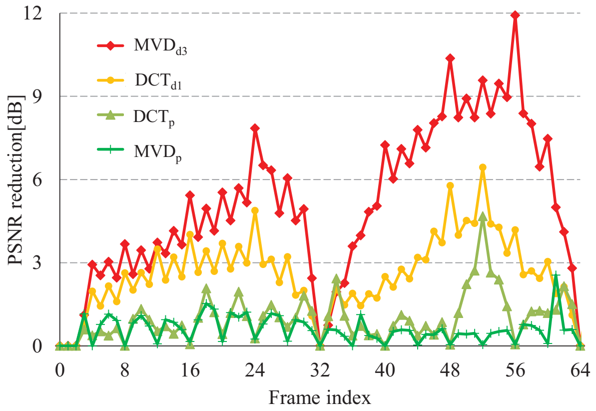

Fig. 4. Errors propagate strongly if dependencies between frames are not taken into account (MVD_{d3} and DCT_{d3}).

3 results in a smaller PSNR reduction compared to adding information to blocks at depth 2 and depth 1. For instance, when motion vectors are modified for video (Fig. 2) and only CU depth 3 is considered, a PSNR reduction of 3.5 dB is obtained. However, if the CUs at both depth 1 and 2 are modified to embed the same amount of information, the PSNR reduction is higher (4.5 dB).

Secondly, when taking frame dependencies into account, layer L2 should be filled first, since errors from this layer will propagate less to other frames. When the amount of added information becomes too much to contain in only L2, L1 also starts filling up, which results in a more drastic PSNR reduction. The lines in Fig. 2 and Fig. 3 show that this strategy can achieve a high payload with the smallest quality drop.

Finally, using a smart frame distribution strategy to minimize error propagation performs better than adding information equally to each frame. By using this strategy, the PSNR reduction can be kept below 3 dB and 1 dB for MVD_p and DCT_p, respectively.

4.2. Frame-by-frame analysis

In order to evaluate the propagation of quality loss and to compare embedding data into motion information and DCT coefficients, the PSNR reduction of the 64 first frames of PartyScene is analysed after embedding 20 kbits in total (2 kbps) using different methods. The video is encoded using QP 27. The result is depicted in Fig. 4.

It can be seen in Fig. 4 that embedding the same amount of information in motion information (MVD_{d3}) results in higher quality losses than modifying DCT coefficients (DCT_{d3}). When a motion vector difference of a PU is modified, the predicted block changes. Therefore, all pixels in this PU are affected, resulting in a high visual quality loss. In contrast, when the last non-zero DCT coefficient is modified, only the frequency corresponding to this coefficient is affected. In addition, the last non-zero coefficient is usually at a high frequency such that the quality impact is minimal.

(a) MVD_{d3} (23.10 dB) (b) MVD_p (33.04 dB)

(c) DCT_{d1} (31.28 dB) (d) DCT_p (32.23 dB)

Fig. 5. The visual quality of frame 106 of PartyScene (QP 27) after inserting information.

By exploiting the dependencies between frames to distribute the information across several layers, MVD_p and DCT_p result in very low PSNR losses. The variance of PSNR losses is also small, which results in a smoother visual quality. Although DCT_p performs slightly better than MVD_d, the capacity of MVD_p is higher. Therefore, when a lot of data must be added, MVD_p can be used.

The visual quality of frame 106 of PartyScene (QP 27) after embedding 20 kbits is shown in Fig. 5. When MVD_{d3} is used, artefacts can clearly be seen (e.g. on the wall in the upper right corner of the picture). On the other hand, the qualities of the other techniques are similar and better.

5. CONCLUSIONS

In this paper, we proposed a low complexity technique to embed information into encoded HEVC video streams without fully decoding and encoding the video. The experimental results show that quality loss can be minimized by adding information first to the smallest blocks of each frame and by taking frame dependencies into account when distributing information across frames. When a smart distribution of information is applied across frames, modifying DCT coefficients only slightly outperforms adding information to motion vector difference.

6. REFERENCES

[1] A. Mansouri, A.M. Aznaveh, F. Torkamani-Azar, and F. Kurugollu, “A Low Complexity Video Watermarking

in H.264 Compressed Domain," *IEEE Transactions on Information Forensics and Security*, vol. 5, pp. 649–657, Dec. 2010.

[2] A. Boho, G. Van Wallendael, A. Dooms, J. De Cock, G. Braeckman, P. Schelkens, B. Preneel, and R. Van De Walle, "End-To-End Security for Video Distribution: The Combination of Encryption, Watermarking, and Video Adaptation," *IEEE Signal Processing Magazine*, vol. 30, pp. 97–107, Mar. 2013.

[3] H.A. Aly, "Data Hiding in Motion Vectors of Compressed Video Based on Their Associated Prediction Error," *IEEE Transactions on Information Forensics and Security*, vol. 6, pp. 14–18, Mar. 2011.

[4] C.-H. Liu and O. T.-C. Chen, "Data Hiding in Inter and Intra Prediction Modes of H.264/AVC," *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 3025–3028, May 2008.

[5] Y. Hu, C. Zhang, and Y. Su, "Information Hiding Based on Intra Prediction Modes for H.264/AVC," *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1231–1234, Jul. 2007.

[6] G.J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[7] Y. Tew and K. Wong, "Information Hiding in HEVC Standard Using Adaptive Coding Block Size Decision," *IEEE International Conference on Image Processing (ICIP)*, pp. 5502–5506, Oct. 2014.

[8] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block Partitioning Structure in the HEVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 2, pp. 1697–1706, Dec. 2012.

[9] F. Bossen, "Common Test Conditions and Software Reference Configurations," *JCTVC-I1100*, Geneva, CH, May 2012.

[10] C.-K. Chan and L. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, vol. 3, pp. 469–474, Mar. 2004.

[11] Rosewarne C. McCann, K. and, B. Bross, M. Naccari, K. Sharman, and G. J. Sullivan, "High Efficiency Video Coding (HEVC) Encoder Description v 16 (HM16)," *document JCTVC-R1002, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 Joint Collaborative Team on Video Coding (JCT-VC)*, Jul. 2014.