Sketch Attacks: A Note on Designing Video Encryption Method in H.264/AVC

Kazuki Minemura* and KokSheik Wong*
*Faculty of Comp. Sci. Info. Tech., University of Malaya, Kuala Lumpur, 50603, Malaysia.
Email: {kazuki.minemura@siswa.,koksheik@}um.edu.my

***Abstract*—In this work, we propose four sketch attacks on H.264/AVC encrypted-compressed video. First, we briefly describe the notion of sketch attack, then deploy the conventional sketch attacks, which are designed for still image, to sketch the frames of H.264/AVC compressed video. Next, we propose four sketch attacks to generate outline of the original frame by using partially decoded information of the H.264/AVC compressed video, including residue DC coefficients, residue AC coefficients, motion vectors, and macroblock bitstream size. Experiments are conducted to verify the performance of the proposed sketch attacks using the ICADR2013 and HEVC dash video data sets. The proposed sketch attacks are also compared with the four conventional sketch attacks. Results indicate that the proposed sketch attack can generate the outline of a frame regardless of its type, where the fidelity of the sketch image improves as resolution increases.**

I. Introduction

The continuous growth of ubiquitous network enables us to access vast number of multimedia contents in a faster and more convenient manner. Video based services such as video on demand, video sharing, video conferencing, advertisement and surveillance system have enriched and revolutionized our daily life. However, the raw video data (e.g., captured video through digital camera) is naturally huge in size and it requires large storage. To facilitate the aforementioned video services, video compression techniques are invented and deployed. The video coding expert group and the moving picture expert group (MPEG) started to research on video coding techniques for practical and commercial applications. Later, the international telecommunication union and MPEG formed the join video team and worked on an efficient video compression standard called H.264 advanced video coding (H.264/AVC) [1], also known as MPEG-4 part 10, which is currently the mostly adopted video compression standard in various digital devices, including smart phone, digital camera, and paid TV service.

As H.264/AVC dominates the market share, practical security tools are undoubtably needed to further enhance the viability of the H.264/AVC standard. A survey of H.264/AVC encryption methods [2] reported that the conventional encryption methods can be generally categorized into three classes, namely: (a) encryption before compression; (b) encryption during compression, and; (c) encryption on bitstream. As its name implies, class (a) encrypts the video prior to compression, but it compromises on the video compression performance. On the other hand, class (b) encrypts the components found in the H.264/AVC compression standards, e.g., intra-prediction type, macroblock (MB) with the same number of motion vectors, motion vector (MV), integer transform type, integer transformed coefficients, scanning order, CAVLC and CABAC. Finally, class (c) manipulates the encoded bitstream while ensuring that the resulting bitstream is still format compliant, i.e., the processes preserve the semantics of the H.264/AVC bitstream.

Most conventional encryption methods justified their robustness against cryptography attacks. For example, Said et al. showed that their proposed method [3], which is based on randomizing the sign of each transformed coefficient, is robust against low-complexity attack. Li et al. proposed [4] a perceptual encryption method for MPEG compressed video by selectively encrypting fixed-length codewords and reasoned that the proposed method is robust against known/chosen-plaintext attacks. Li et al. then put forward a general quantitative cryptanalysis [5] for permutation-only encryption [6]. While the conventional methods are secure from the perspective of cryptanalysis (i.e., from the perspective of cryptography), a format-compliant video encryption may not be completely secure from other forms of attack. In particular, instead of obtaining the encryption (decryption) key or recovering the high resolution video with actual frame rate, an adversary may be equally satisfied if additional information (e.g., low resolution of the plaintext video, coarse sketch/outline of the original video, low framerate version of the video) can be obtained directly from the encrypted video by deploying primitive signal processing techniques or collecting some elementary statistical features.

For this purpose, Li et al. [7] proposed a simple feature extraction method to sketch the outline of the original (i.e., plaintext) image directly from the encrypted JPEG image, which is severely distorted by perceptual encryption [4], partial encryption [8], transparent scrambling [9], or selective encryption [10]. Li et al.'s method relies on the number of nonzero-DCT coefficients, i.e., non-zero-count attack (NZCA), and the sketched image is represented as a binary image, viz., black and while. However, NZCA requires the adjustment of two threshold values to obtain the clear outline image. Minemura [11] et al. then proposed three sketching methods, namely, nonzero coefficient count (NCC), position of last nonzero coefficient (PLZ), and energy of AC coefficients (EAC), to generate outline of higher fidelity without the need to tune any threshold value. To distinguish these attacks from cryptanalysis, we name them as*sketch attack*.

978-616-361-823-8

Although promising results are obtained through the aforementioned sketch attacks, features extracted by these sketch attacks are limited to DCT coefficients in JPEG [12] and MPEG1/2 [13], [14]. Specifically, components of H.264/AVC standard [1], including quantization parameter, intra-prediction mode, and block sizes remain unexplored. In addition, the temporal axis of video is still not considered.

Therefore, in this work, four sketching attacks are proposed to generate the outline of H.264/AVC compressed video. Specifically, the proposed sketching methods rely on integer transformed coefficient, motion vector, and macroblock bitstream size, which are the main elements of H.264/AVC compressed video. The rest of this paper is organized as follows: Section II briefly reviews the state-of-the-art H.264/AVC standard. In Section III, four novel sketch attacks are proposed to sketch the outline of H.264/AVC compressed video, including (a) DC magnitude (DCM), (b) summation of AC coefficients (SAC), (c) motion vector magnitude (MVM), and (d) macroblock bitstream size (MBB). Section IV discusses the experimental results. Section V summarizes this work and presents our future directions for this research.

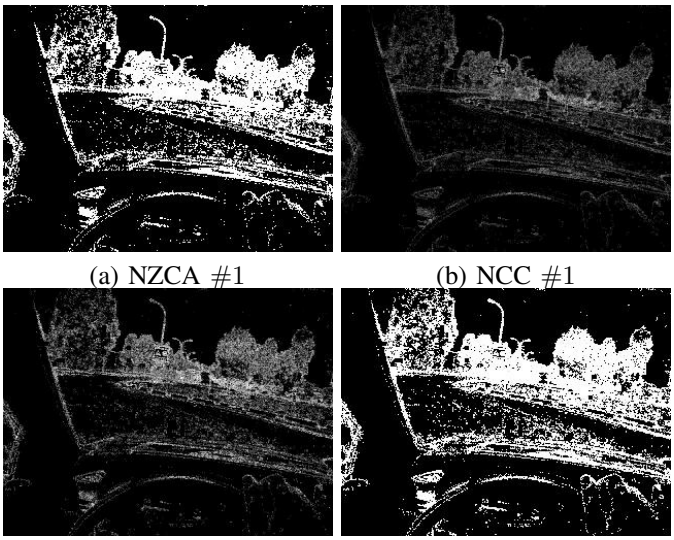
II. Preliminaries

In this section, we briefly review the state-of-the-art H.264/AVC compression standard [1]. H.264/AVC is designed to compress video effectively. First, each input frame is classified into to intra-frame (I-frame), where the frame is coded independently, or inter-frame (P- or B-frame), where the differences between the current frame and the reference frame(s) are coded, to minimize the video bitstream size. Each intra-frame is divided into non-overlapped blocks of 16×16 pixels called macroblock (MB). Each MB is further divided into non-overlapping square blocks of various sizes (i.e., one 16×16, four 8×8, sixteen 4×4 pixel blocks) and (intra-) predicted. Next, integer transform and quantization are applied to each MB. Then, each quantized MB is entropy coded using either context adaptive variable length coding (CAVLC) or context adaptive binary arithmetic coding (CABAC). Similarly, each inter-frame is also divided into MBs (i.e., blocks of 16×16 pixels), and each MB is further divided into blocks of various sizes (i.e., 16×16, 16×8, 8×16, 8×8, 8×4 or 4×4) depending on the quarter-pixel motion vector estimation results with multiple reference frames. Next, each MB is processed in the same manner as in the intra-frame.

III. Proposed sketch attacks

Without lost of generality, instead of considering the encrypted video, the sketch attacks are performed directly on a H.264/AVC compressed video (i.e., I-slice). Fig. 1(a-d) show the sketch frames by using the convention methods as follows: (a) NZCA with $t_1 = 2$ and $t_2 = 12$; (b) NCC; (c) PLZ, and; (d) EAC. All four conventional methods can generate an outline image for I-frame. On the other hand, Fig. 2 shows the images generated by applying (a) NZCA, (b) NCC, (c) PLZ and (d) EAC to a P-frame, respectively. It is apparent that the conventional methods fail to infer any additional information when an P- or B-frame is encountered. It is because these methods are designed to attack still images. As such, there is a need to specifically design sketch attacks for video by considering features that consistently appear throughout the entire video sequence. In addition to the temporal axis (which does not exist in still image), H.264/AVC adopts integer transform and various MB types. Besides, the features of transformed coefficient are different from the previous compression standards such as JPEG and MPEG1/2. Therefore, there are various unexplored opportunities in sketch attack for H.264/AVC compressed video.

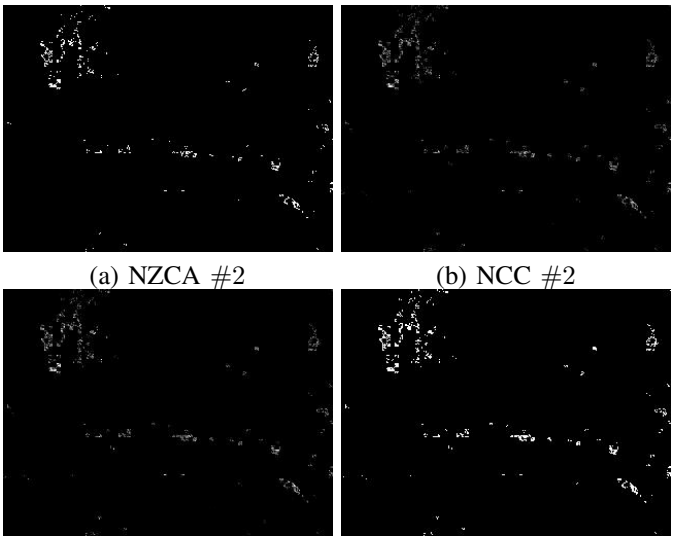
In this section, we propose four simple sketch attacks by further extending our previous work [11] to H.264/AVC compressed video to generate outline information. To facilitate



(a) NZCA #1 (b) NCC #1

(c) PLZ #1 (d) EAC #1

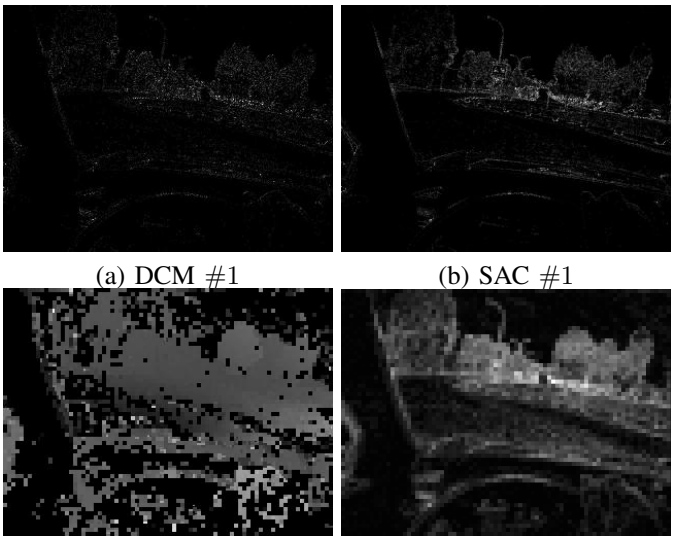
Fig. 1. Outline of the original I-frame sketched by using: (a) NZCA with $t_1 = 2$ and $t_2 = 12$; (b) NCC; (c) PLZ, and; (d) EAC



(a) NZCA #2 (b) NCC #2

(c) PLZ #2 (d) EAC #2

Fig. 2. Outline of the original P-frame, i.e., second frame (#2), sketched by using: (a) NZCA with $t_1 = 2$ and $t_2 = 12$; (b) NCC; (c) PLZ, and; (d) EAC



(a) DCM #1 (b) SAC #1

(c) MVM #2 (d) MBB #1

Fig. 3. Outline of the original frame sketched by applying the proposed sketch attacks: (a) DCM; (b) SAC; (c) MVM, and; (d) MBB

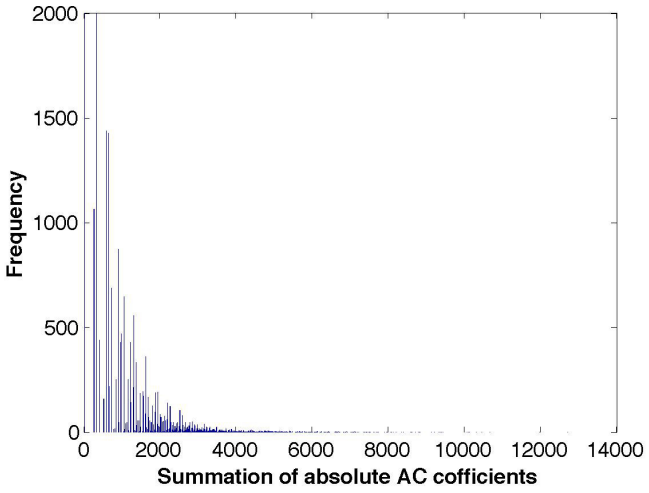


Fig. 4. Histogram of EAC#1

the presentation, we consider H.264/AVC compressed video in level 5.1 and baseline profile (BP), which utilizes 4×4 integer discrete cosine transform and 7 block sizes, namely, 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, and 4×4.

A. DC Magnitude (DCM)

The DC coefficients themselves collectively form the original frame with reduced resolution of $1/4$ for H.264/AVC. In the encoding process, instead of the raw values, the residue values (prediction error) are coded and they are readily available in the bitstream. However, the DC intra prediction process in H.264/AVC exploits the neighboring DCs' gradient and hence it functions in a similar manner as an edge detection. In other words, the residue DC values carry some form of edge information. Based on this observation, we propose to sketch the outline by representing the edges in a video frame as follows:

$$\phi_{DCM}(i,j) \leftarrow \text{round}\left(255 \times \frac{d(i,j)}{\max\{d(i,j)\}}\right), \qquad (1)$$

where d denotes the magnitude of the residue DC at the (i,j)-th 4×4 block. Fig. 3(a) shows the sketched image generated by Eq. (1). Although the sketched image is generally dark and of low contrast, we can observe the edges in the frame.

B. Summation of AC coefficients (SAC)

We improve the idea in [7], [11] by sketching the outline of the original image directly from AC coefficients in H.264/AVC compressed video. Minemura et al. [11] reported that EAC can generate a better outline of the original plaintext image without tuning any parameter. However, the EAC image is similar to NZCA image [7] when applied to a H.264/AVC compressed video (see Fig. 1(a) and (d)). Due to the high compression efficiency of H.264/AVC, the summation of magnitude of AC coefficients in a block is usually of small value. However, large value may exists, for example, $\max = 12704$ and $\mu = 313$ for the distribution shown in Fig. 4. Note that the difference between the maximum and average is of the order of two-digit. Therefore, most of the blocks having higher energy will be represented by the maximum value, i.e., 255 in the EAC image. To solve this problem, we modify the EAC attack by replacing the denominator (i.e., average) by the maximum value as follows:

$$\phi_{SAC}(i,j) \leftarrow \text{round}\left(255 \times \frac{s(i,j)}{\max\{s(i,j)\}}\right), \qquad (2)$$

where

$$s(i,j) \leftarrow \left(\sum_{u=1}^{4}\sum_{v=1}^{4}|g_{u,v}(i,j)|\right) - |g_{1,1}(i,j)|, \qquad (3)$$

where $|g_{u,v}(i,j)|$ denotes the magnitude of $g_{u,v}(i,j)$. Fig. 3(b) shows the sketched image generated by Eq. (2). Although, the resolution of Fig. 3(b) is $1/4$ of its original counterpart (due to 4×4 block transformation size), we can observe that the SAC image is of greater fidelity than EAC [11].

C. Motion Vector Magnitude (MVM)

We perform sketch attack based on the temporal information, which is an intrinsic property of a video. Since motion vector is often (but not always) related to the motion of an object, we exploit the magnitude of motion vector to sketch the outline image follows:

$$\phi_{MVM}(i,j) \leftarrow \text{round}\left(255 \times \frac{v(i,j)}{\max\{v(i,j)\}}\right), \qquad (4)$$

where $v(i,j)$ denotes the average motion vector magnitude in the (i,j)-th 16×16 macroblock. Fig. 3(c) shows the sketched image generated by Eq. (4). Note that the resolution of the sketched image is $1/16$ of its original counterpart.

D. Macroblock Bitstream Count (MBB)

In H.264/AVC, each frame is divided into various macroblocks (MB), where each MB is further coded by using either CAVLC or CABAC. Hence, the bitstream size of each MB (i.e., the raw number of bits including the header part) reflects its complexity. Specifically, a complex macroblock is expected to spend more bits for storage, and vice versa. Therefore, MB bitstream size can be utilized to sketch the outline of a video frame, and we propose to generate the outline as follows:

$$\phi_{MBB}(i,j) \leftarrow \text{round}\left(255 \times \frac{b(i,j)}{\max\{b(i,j)\}}\right), \qquad (5)$$

where $b(i,j)$ denotes the number of bits spent on coding the (i,j)-th MB. Fig. 3(d) shows the sketched image generated by Eq. (5). Similar to MVM, the resolution of the sketch image is $1/16$ of its original counterpart.

IV. Experimental results and discussions

To evaluation the performance of our proposed sketch attacks, all 15 video sequences from the ICDAR2013 video dataset [15] and 2 video sequences from Ultra High Definition HEVC dash dataset [16] are considered. The resolution of the ICDAR2013 videos range from 720×480 to 1280×960 pixels, while the resolutions of the HEVC dash videos are 1920×1080 and 3840×2160 pixels. The test videos are encoded into a H.264/AVC compressed video using the baseline profile with level 5.1 and then partially decoded for sketching the outline. Since the conventional sketch attacks are designed to attack encrypted-compressed still-image (i.e., JPEG and JPEG-XR), they are reimplemented to operate directly in the H.264/AVC compressed domain. In particular, the proposed and existing methods are implemented in H.264/AVC [1] using the reference software [17] for partial decoding and Matlab for generating the sketch image. In the following subsection, we discuss the results obtained by our proposed sketch attacks and put forward a guideline on designing an encryption technique for H.264/AVC compressed video.

A. Scenario I: Intra-Frame

First, we consider the scenario where only a single frame (i.e., I-frame) is considered for sketching purposes. It should be noted that fidelity of the sketched outline image is an important evaluation criteria of a sketch attack because any practical sketch attack is supposed to infer additional information about the input (i.e., encrypted-compressed) video. Fig. 3 shows the outputs generated by four proposed sketch attacks, i.e., DCM, SAC, MVM and MBB. Similarly, Fig. 1 shows the output generated by four conventional sketch attack techniques, namely, i.e., NZCA, NCC, LPZ and EAC, where $\#$ indicates the frame index. Note that two thresholds, i.e., r_1 and r_2, are required by NZCA and they are set to $r_1 = 2$ and $r_2 = 12$, respectively, throughout the experiments. As expected, DCM provides the outline image and we can observe the output in Fig. 3(a). Fig. 1(a-d) suggest that all conventional methods (i.e., NZCA, NCC, LPZ and EAC) are able to sketch the I-frame, i.e., inferring additional information. Although Minemura et al, [11] reported that EAC can represent the outline of the image more precisely than that of Li et al.'s method (i.e., NZCA [7]), for H.264/AVC compressed video, the output of EAC is almost identical to that of the binary NZCA image due to the larger range of values assumed by the coefficient energy.

On the other hand, the output of the proposed SAC, MVM, and MBB are shown in Fig 3(b-d), respectively. SAC outputs a sketch with higher fidelity when compared to the output of EAC, but the quality is inferior to the output of NCC and LPZ. On the other hand, MVM and MBB generate rough outline images due to the macroblock coding size limitation, which outputs a single number for every macroblock of size 16×16. However, MBB can sketch the frame outline more precisely than DCM, NZCA, EAC, SAC and MVM. To further investigate this promising approach, Fig. 5(a-c) show the sketched images generated by MBB using various input video sequences. It is obvious that the sketched image appear similar to the outline of the original input frame. In addition, the higher resolution of the video in question, the higher fidelity is achieved. This claim is confirmed by considering the same test videos but in two resolutions, namely, 1920×1080 and 3840×2160 pixels. The sketched images are shown in Fig. 5(e) and (d).

B. Scenario II: Inter-Frames

In this subsection, we demonstrate the effective of the proposed sketch attacks SAC and MBB in generating the outline over a series of frames, including Intra- and Inter-Frames. For comparison purposes, the conventional methods NCC and PLZ are also considered. Fig. 6 shows the original frame sequence and the corresponding sketched images for frame $\#1, \#2, \cdots, \#5$. Fig. 6(a-d) suggest that NCC, LPZ and SAC can successfully sketch the first frame (i.e, Intra-frame $\#1$) of the video sequence, but fail to provide additional information for the following (P-) frames. One reason is that the number of nonzero coefficients are significantly reduced in the case of an inter-frame. On the other hand, the proposed MBB technique is able to sketch the outline from both the I- and P-frames. This claim is supported by the output shown in Fig. 6(e), where a sketch of each frame can be generated. It is because MBB considers the relative features in each frame. That is, although the bitstream size of an inter-frame is considerably smaller than that of an intra-frame, MBB considers the ratio of $b(i,j)$ to $\max\{b(i,j)\}$, which is automatically adjusted regardless of the frame type. From a different perspective, MBB is utilizing the collective information, including MVs, transform coefficients and complexity of the MB, which affects the number of bits need to store the MB in question. Hence, MBB outperforms the conventional sketch attacks considered, as well as the proposed SAC and DCM.

C. Sketch Robustness

The robustness of the conventional and the proposed sketch attacks for H.264/AVC video encryption methods [2] are

TABLE I
THE ROBUSTNESS OF SKETCH ATTACKS FOR H.264/AVC VIDEO ENCRYPTION

	S	L	SDCT	MVD	SSO	Inter	Intra
NZCA [7]	✓	✓	×	✓	✓	×	✓
NCC [11]	✓	✓	×	✓	✓	×	✓
LPZ [11]	✓	✓	×	✓	✓	×	✓
EAC [11]	✓	×	×	✓	✓	×	✓
DCM	✓	×	×	✓	✓	×	✓
SAC	✓	×	×	✓	✓	×	✓
MVM	✓	✓	✓	×	✓	×	✓
MBB	✓	✓	✓	✓	✓	✓	✓

recorded in Table I for comparison purposes. Here, seven commonly deployed encryption operations in the compressed domain are considered. In particular, S refers to the process of randomizing the sign of nonzero transformed coefficients, L indicates the randomization of the level of transformed coefficients, SDCT signifies secret DCT transform, MVD means MV difference modification, SSO represents secret scannig order, Inter stands for shuffling inter blocks with same the number of MVs, and finally Intra denotes intra-prediction mode change. If row α and column β is marked with ✓ (×), it means sketch attack method α can (cannot) sketch the outline of the video when operation β is deployed.

It is observed that the sketch attacks proposed in [7] and [11] fail to sketch the outline when the DCT coefficients are modified, while the MV magnitude based method may also fail due to the modified MV information. On the other hand, our proposed methods, particularly, MBB, is viable in sketching the outline of the frame even when all seven commonly deployed encryption operations are applied. It is because our method only computes the total number of bits required to store each MB, in which case most perceptual / selective / format-compliant encryption algorithms aim at maintaining the bitstream size of the original video. Hence, these encryption operations do not affect the performance of the proposed sketch attack MBB.

Therefore, when designing an encryption technique for H.264/AVC compressed video, in addition to being robustness against the classical plaintext only (brutal force), known and chosen plaintext attacks, one should also ensure that it is robustness against sketch attack, particular the proposed MBB sketch attack, which is practical for both intra- and inter-frames.

V. CONCLUSIONS

Four novel sketch attacks are proposed to generate the outline image of the H.264/AVC encrypted-compressed video at reduced frame rate / resolution. The proposed method successfully sketched the outline image using the intra-frame of the compressed video. In addition, macroblock bitstream size based attack succeeded to sketch the outline for both intra and inter frames, while the conventional methods failed to provide additional information.

As future work, we want to improve the fidelity of the sketched image while achieving higher resolution than those achieved by the proposed methods. Implementation of these attacks on the latest video compression standard such as HEVC will be further pursued.

REFERENCES

[1] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 560–576, 2003.

[2] T. Stutz and A. Uhl, "A survey of h.264 avc/svc encryption," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 3, pp. 325–339, March 2012.

[3] A. Said, "Measuring the strength of partial encryption schemes," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 2, Sept 2005, pp. II–1126–9.

[4] S. Li, G. Chen, A. Cheung, B. Bhargava, and K.-T. Lo, "On the design of perceptual MPEG-video encryption algorithms," *IEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 2, pp. 214–223, 2007.

[5] S. Li, C. Li, G. Chen, N. G. Bourbakis, and K.-T. Lo, "A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Image Commun.*, vol. 23, no. 3, pp. 212–223, Mar. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.image.2008.01.003

[6] F. Dufaux and T. Ebrahimi, "A framework for the validation of privacy protection solutions in video surveillance," in *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, July 2010, pp. 66–71.

[7] W. Li and Y. Yuan, "A leak and its remedy in jpeg image encryption," *Int. J. Comput. Math.*, vol. 84, no. 9, pp. 1367–1378, Sep. 2007. [Online]. Available: http://dx.doi.org/10.1080/00207160701294376

[8] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *IEEE Trans. Signal Process.*, vol. 48, no. 8, pp. 2439–2451, 2000.

[9] M. Pazarci and V. Dipcin, "A MPEG2-transparent scrambling technique," *IEEE Trans. Consum. Electron.*, vol. 48, no. 2, pp. 345–355, 2002.

[10] A. Massoudi, F. Lefebvre, C. De Vleeschouwer, B. Macq, and J.-J. Quisquater, "Overview on selective encryption of image and video: challenges and perspectives," *EURASIP J. Inf. Secur.*, vol. 2008, pp. 5:1–5:18, Jan 2008. [Online]. Available: http://dx.doi.org/10.1155/2008/179290

[11] K. Minemura, Z. Moayed, K. Wong, X. Qi, and K. Tanaka, "Jpeg image scrambling without expansion in bitstream size," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2012, pp. 261–264.

[12] W. B.Pennebaker and J. L.Mitchell, *JPEG: still image data compression standard*. Van Nostrand Reinhold, 1992.

[13] *ISO/IEC Information technology – coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – part 2: video*, ISO/IEC Std. 11 172-2:1993, 1993.

[14] *ISO/IEC Information technology – generic coding of moving pictures and associated audio information: video*, ISO/IEC Std. 13 818-2:2000, 2000.

[15] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. Gomez i Bigorda, S. Robles Mestre, J. Mas, D. Fernandez Mota, J. Almazan Almazan, and L.-P. de las Heras, "Icdar 2013 robust reading competition," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, Aug 2013, pp. 1484–1493.

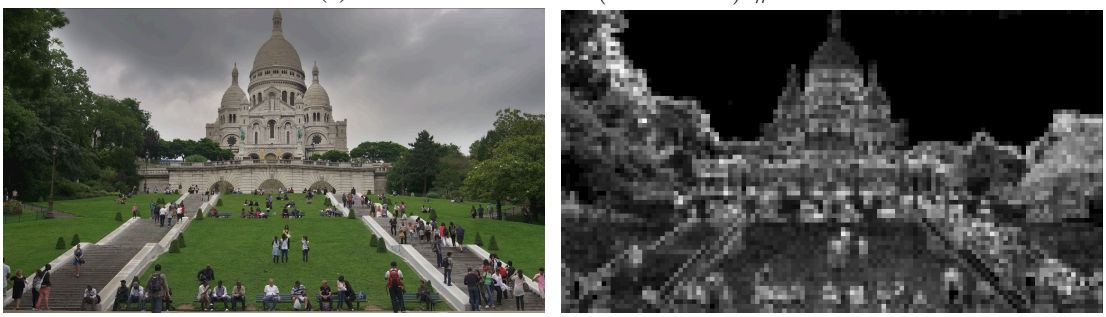
[16] J. L. Feuvre, J.-M. Thiesse, M. Parmentier, M. Raulet, and C. Daguet, "Ultra high definition hevc dash data set," in *MMSys*, 2014, pp. 7–12.

[17] *H.264/AVC reference software*, [Online]. Available: http://www.http://iphome.hhi.de/suehring/tml/.

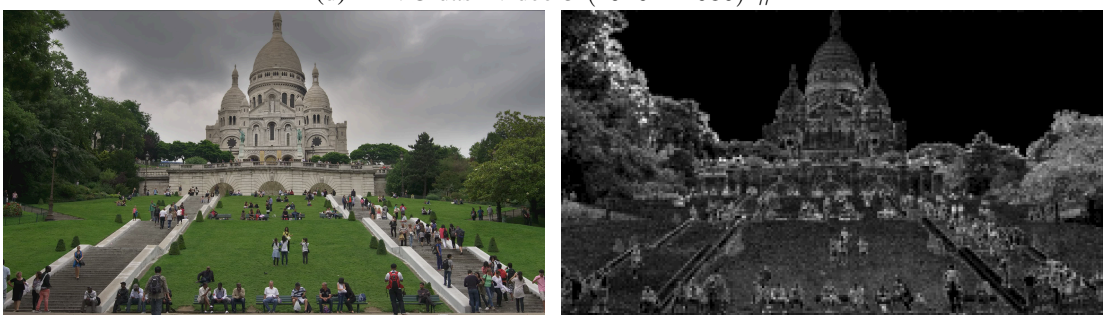
(a) ICDAR2013 video1 (720×480) #1

(b) ICDAR2013 video17 (1280×960) #1

(c) ICDAR2013 video32 (1280×720) #1

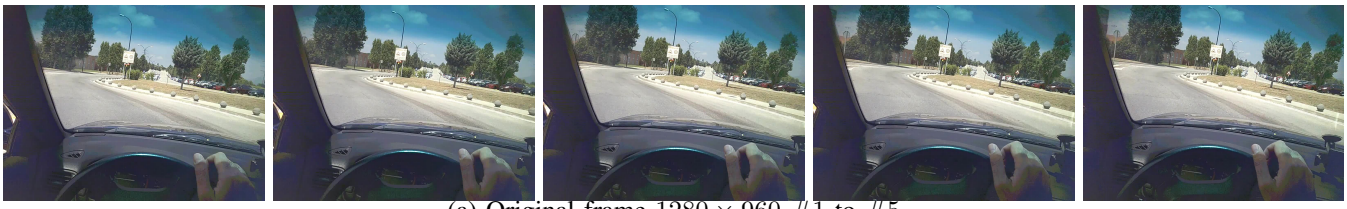


(d) HEVC dash video 5 (1920×1080) #1

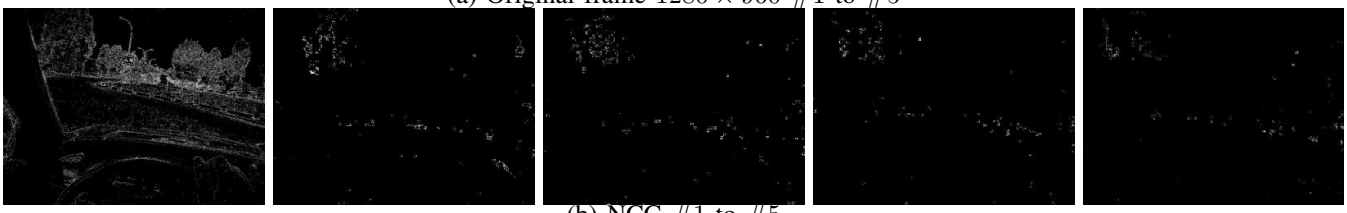


(e) HEVC dash video 13 (3840×2160) #1

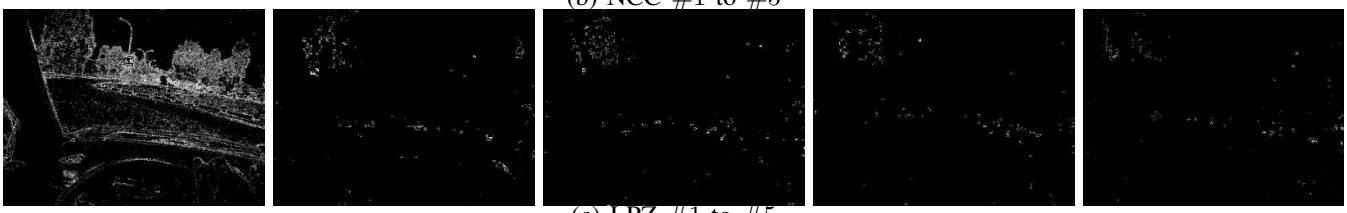
Fig. 5. Comparison of various input images



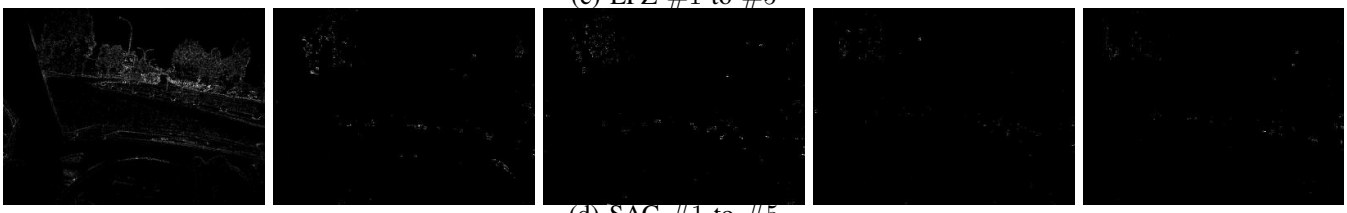
(a) Original frame 1280×960 #1 to #5



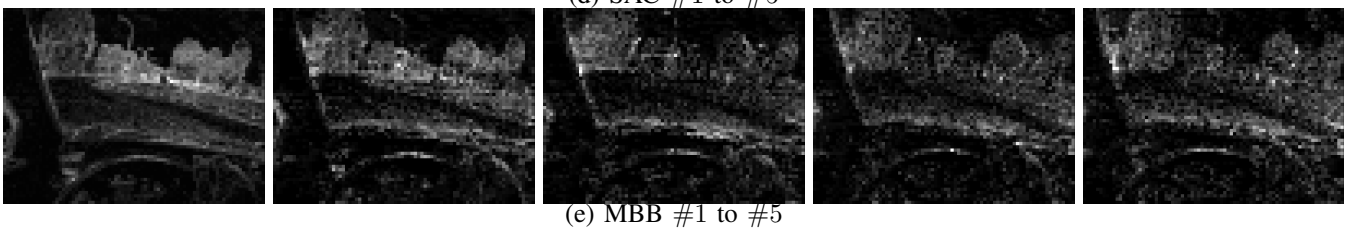
(b) NCC #1 to #5



(c) LPZ #1 to #5



(d) SAC #1 to #5



(e) MBB #1 to #5

Fig. 6. Original and sketched frame from (#1 to #5)