

Fast Protection of H.264/AVC by Selective Encryption

Zafar Shahid, Marc Chaumont, William Puech

▶ **To cite this version:**

Zafar Shahid, Marc Chaumont, William Puech. Fast Protection of H.264/AVC by Selective Encryption. Singaporean-French Ipal Symposium (SinFra), Feb 2009, Fusionopolis, Singapore. pp.11-21, 10.1142/9789814277563_0002 . lirmm-00416013

HAL Id: lirmm-00416013

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00416013

Submitted on 11 Sep 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Author manuscript, published in "Singaporean-French IPAL Symposium, SinFra 2009, Fusionopolis, Singapore : "

FAST PROTECTION OF H.264/AVC BY SELECTIVE ENCRYPTION

Z. SHAHID, M. CHAUMONT and W. PUECH

LIRMM,UMR CNRS 5506, University of Montpellier II,
161, rue Ada, 34392 Montpellier CEDEX 05, France
E-mail: zafar.shahid@lirmm.fr, marc.chaumont@lirmm.fr, william.puech@lirmm.fr
www.lirmm.fr

This paper proposes a new method for the protection of copyrighted multimedia content. Here the problems of compression and selective encryption (SE) have been simultaneously addressed for the state of the art video codec H.264/AVC. SE is performed in the context adaptive variable length coding (CAVLC) module of video codec. For this purpose, CAVLC is converted to an encryption cipher using permutation of equal length codes from a specific variable length coding (VLC) table. In our scheme, entropy coding engine serves the purpose of encryption step without affecting the coding efficiency of H.264/AVC by keeping the bitrate unchanged and generating completely compliant bit stream. Our scheme requires a negligible computational power. Nine different benchmark video sequences containing different combinations of motion, texture and objects are used for experimental evaluation of the proposed algorithm.

Keywords: Selective encryption, H.264/MPEG-4 Part 10, Scrambling, CAVLC

1. Introduction

With the rapid growth of processing power and network bandwidth, many multimedia applications have emerged in the recent past. As digital data can easily be copied and modified, the concern about its protection and authentication have surfaced. Digital rights management (DRM) has emerged as an important research field to protect the copyrighted multimedia data. Encryption is used to restrict access of digital data to only authenticated users. Video data is huge in amount and multimedia applications have real time constraints, the concept of SE has evolved in which only a small part of the whole bit stream is encrypted.[1] In this work, we have transformed CAVLC module of H.264 into encryption process. Only the non-zeros coefficients (NZs) are encrypted which are encoded separately by CAVLC.

SE of H.264/AVC has been studied in Ref. 2 who has done partial

lirmm-00416013, version 1 - 11 Sep 2009

lirmm-00416013, version 1 - 11 Sep 2009

encryption of some fields as intra-prediction mode, residue data, inter-prediction mode and motion vectors. Ref. 3 has presented an idea of encryption for H.264/AVC. They do permutations of the pixels of macro-blocks (MBs) which are in ROI. The drawback of this scheme is that bitrate increases as the size of ROI increases. This is due to change in the statistics of ROI as it is no more a slow varying region which is the basic assumption for video signals. The use of general entropy coder as encryption step has been studied in the literature in Ref. 4. It encrypts NZs by using different Huffman tables for each input symbols. The tables, as well as the order in which they are used, are kept secret. This technique is vulnerable to known plaintext attack as explained in Ref. 5.

We organize our work as follows. In Section 2, overview of H.264/AVC and CAVLC is presented. It explains the working of CAVLC along with its limitations from encryption point of view. We explain the whole system architecture in Section 3. Section 4 contains its experimental evaluation and performance analysis including its computational efficiency, histogram distribution. In Section 5, we present the concluding remarks about the proposed scheme.

2. Preliminaries

2.1. *Overview of H.264/AVC*

H.264/AVC[6] is state of the art video coding standard of ITU-T and ISO/IEC. It has enhanced compression performance and an overview of this algorithm is shown in Fig. 1. In H.264/AVC, an input video frame is divided into blocks of 16x16, called MB and each of them is encoded separately. Each video frame can be encoded as *intra* or *inter*. In *intra* frame, the current MB is predicted spatially from MBs which have been previously encoded, decoded and reconstructed (MB at top and left). In *inter* mode, motion compensated prediction is done from the reference frames. The purpose of the reconstruction in the encoder is to ensure that both encoder and decoder use identical reference frames to create the prediction. If this is not the case, then the predictions in encoder and decoder will not be identical, leading to an increasing error or drift between the encoder and decoder. The difference between original and predicted frame is call residual. This residual is coded using transform coding followed by quantization and zigzag scan. In the last step, entropy coding comes into action. First, quantized transform coefficients are first run-length encoded. These runs and levels are then encoded using either CAVLC or context adaptive

lirmm-00416013, version 1 - 11 Sep 2009

binary arithmetic coding (CABAC). On the decoding side, compressed bit stream is decoded by entropy decoding module, followed by inverse-zigzag scan. These coefficients are then rescaled and inverse transformed to get the residual signal which is added to the predicted signal to get the original signal back.

H.264/AVC has some additional features as compared to previous video

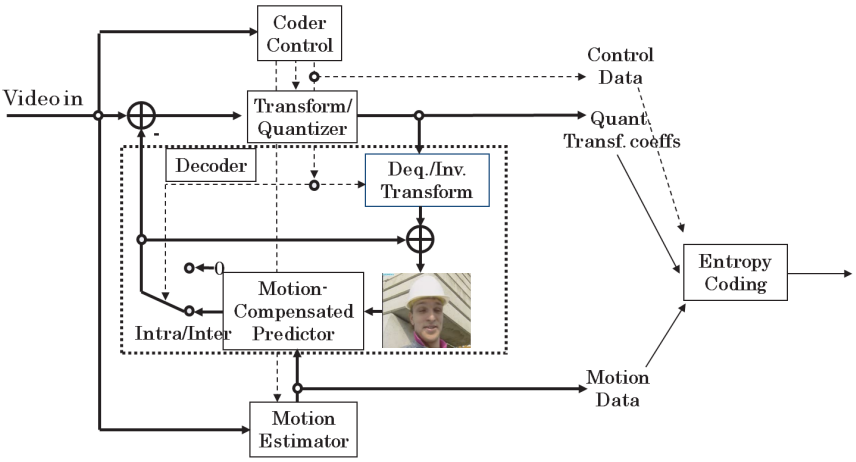


Fig. 1. Block diagram of state of the art H.264/AVC.

standards. In *baseline* profile of H.264/AVC, It has 4x4 transform in contrast to 8x8 transform of previous standards. In higher profiles, it offers transform coding for adaptive size. DCT transform has been replaced by Integer transform which does not need any multiplication operation and can be implemented by only additions and shifts. Thus it requires lesser number of computations. It uses a uniform scalar quantization. For *Inter* frame, H.264/AVC supports variable block size motion estimation, quarter pixel accuracy, multiple reference frames, improved skipped and direct motion inference. For *Intra* frame, it offers additional spatial prediction modes. All these additional features of H.264/AVC are aimed to outperform previous video coding standards.

2.2. *Code Adaptive Variable Length Coding*

H.264 provides two modes for entropy coding which are CAVLC[7] and CABAC.[8] In entropy coding, run length encoding is performed first as

it encodes levels and runs separately. In this paper, we are presenting an encryption scheme based on CAVLC. It scans the coefficients in reverse order (from high frequency NZs to low frequency NZs) as shown in Fig. 2. CAVLC is designed to better exploit the characteristics of NZs, it works in several steps as shown in Fig. 3. Encoding of total NZs and number of trailing ones (T1's) is done by a single syntax element named *coeff_token*. It is followed by coding of signs of T1's. Remaining NZs are then coded using seven VLC tables. Lastly, total number of zeros and then runs of zeros are coded.

To keep the bit stream compliant, which is a required feature for some direct operations (displaying, time seeking, cutting, etc.), we cannot encrypt *coeff_token*, total number of zeros and runs of zeros. The suitable syntax element which can be encrypted is the remaining NZs.

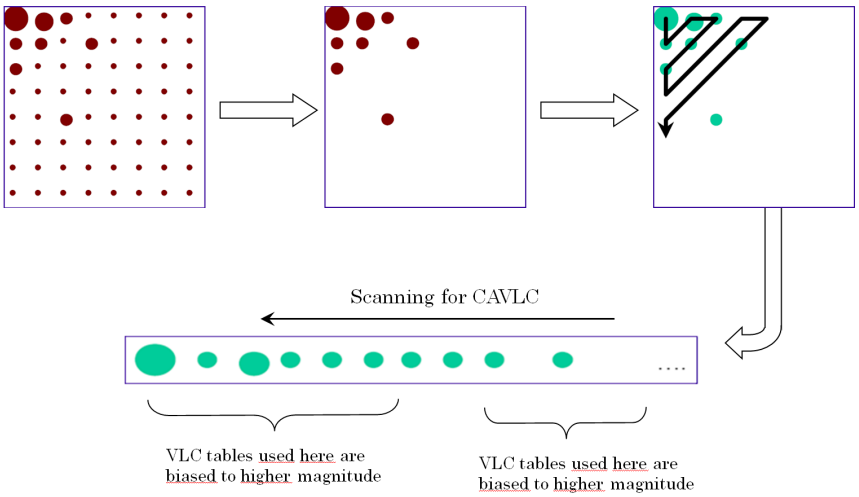


Fig. 2. Scanning order of NZs in CAVLC.

All the remaining NZs (sign and magnitude) are mapped to some code from a specific variable length coding (VLC) table. CAVLC used seven fixed VLC tables. Adaptive nature is introduced by changing the table based on the magnitude of the current NZ being coded. Threshold to increment the table number is given in Table 1. The tree representation of first four tables is shown in Fig. 4.

lirmm-00416013, version 1 - 11 Sep 2009

lirmm-00416013, version 1 - 11 Sep 2009

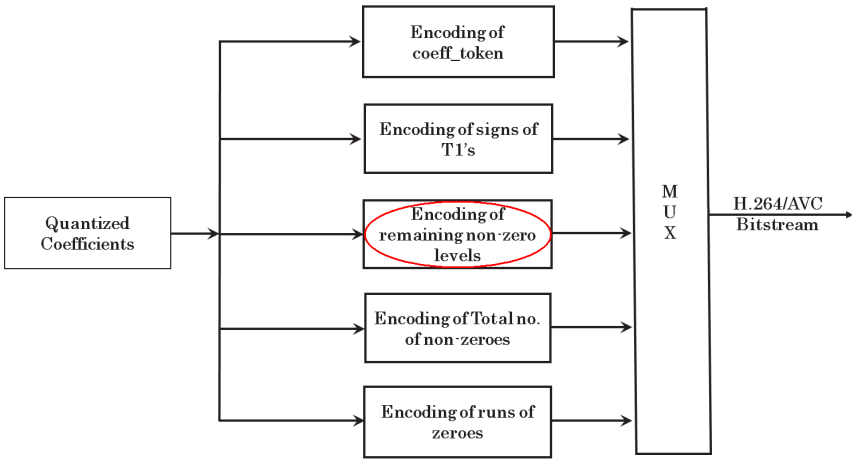


Fig. 3. Block diagram of CAVLC of H.264/AVC.

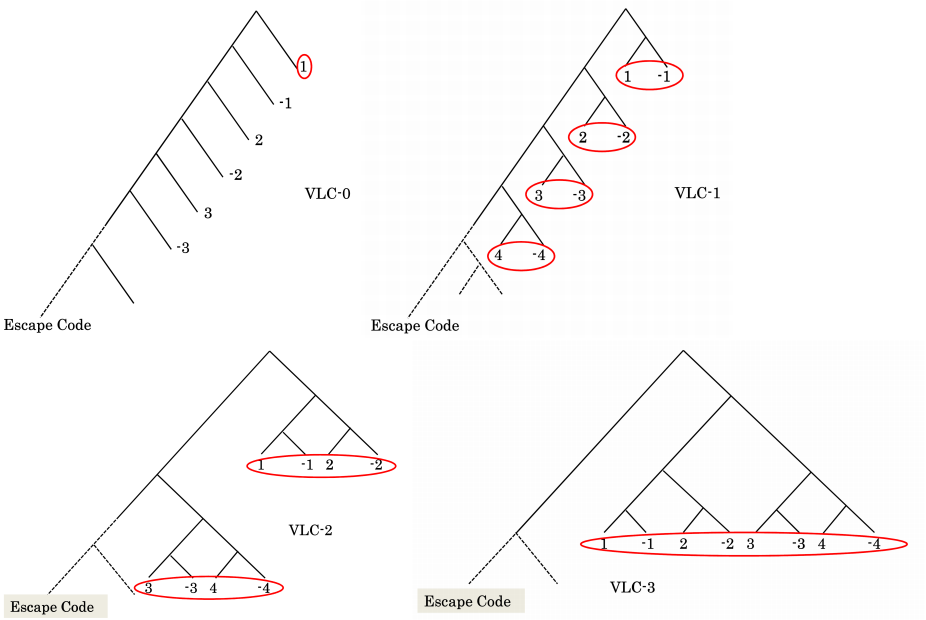


Fig. 4. Tree representation of first four VLC tables used in CAVLC.

lirmm-00416013, version 1 - 11 Sep 2009

3. System Architecture

To keep the size of the bit stream unchanged, we permute the NZ with only those NZs whose VLC code have the same length. We initialize pseudo-random number generator (PRNG) with a secret key. The permutation space is dependent on the VLC table. VLC codes, having same code length, constitute the permutation space for that table. Every table has a specific sized permutation space as shown in Table 1. The permutation space is 2^n where n is the number of VLC table being used. The block diagram of our scheme is shown in Fig. 5. In table VLC0, every NZ has different code length, consequently we cannot permute the NZs in this table.

Table 1. Permutation space and threshold of VLC tables used in CAVLC of H.264/AVC.

Table	Permutation Space	Threshold (to increment table)
VLC0	1 (2^0)	0
VLC1	2 (2^1)	3
VLC2	4 (2^2)	6
VLC3	8 (2^3)	12
VLC4	16 (2^4)	24
VLC5	32 (2^5)	48
VLC6	64 (2^6)	N/A

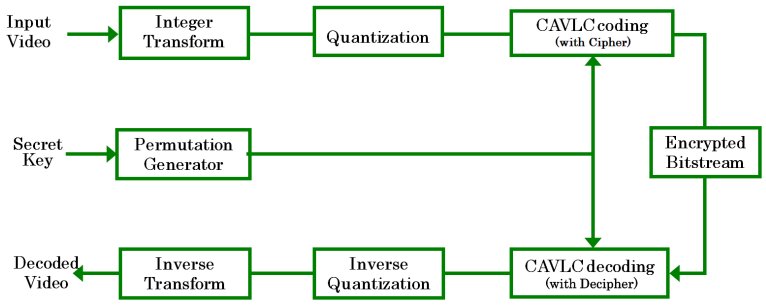


Fig. 5. Block diagram of encryption and decryption process in H.264/AVC.

lirmm-00416013, version 1 - 11 Sep 2009

3.1. *Encryption Process*

Let x be a quantized coefficient which is to be encoded by n^{th} table and the encrypted coefficients y can be given as:

$$\gamma = rand() \mod 2^n, \tag{1}$$

$$y = x + \gamma, \tag{2}$$

where 2^n is size of permutation space of table VLCn. The process can be done in converse in context adaptive variable length decoding (CAVLD) module of H.264 decoder.

3.2. *Histogram Modification*

Distribution of DCT transformed coefficients can be well modeled by Laplacian[9] and Cauchy[10] distributions as shown in Fig. 6. But during permutation process, PRNG treats all NZs having equal code length with uniform probability, thus producing the histogram of encrypted coefficients as stair case as shown in Fig. 6 which is not natural and can be used for cryptanalysis. In H.264, CAVLC uses multiple VLC tables, where the same NZ have different code length in different VLC tables as shown in Table 1. So the histogram is not pure staircase but this effect is obvious in some parts of the graph. To keep the distribution of the encrypted coefficients same as the original ones, we have modified the PRNG framework to make it capable to generate the NZs with their original probabilities.

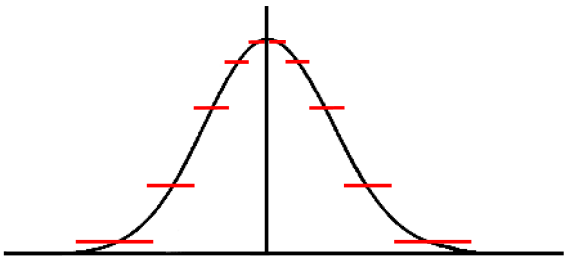
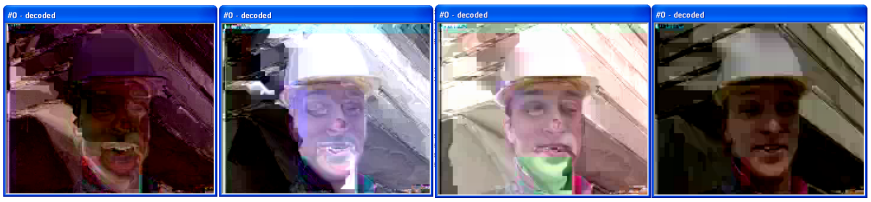


Fig. 6. Coefficient histogram before after permutation with standard PRNG.

lirmm-00416013, version 1 - 11 Sep 2009

4. Experimental Results

For the experimental results, nine benchmark video sequences have been used for the analysis in QCIF format. Each of them represents different combinations of motion (fast/slow, pan/zoom/rotation), color (bright/dull), contrast (high/low) and objects (vehicle, buildings, people). The video sequences 'bus', 'city' and 'foreman' contain camera motion while 'football' and 'soccer' contain camera panning and zooming along with object motion and texture in background. The video sequences 'harbour' and 'ice' contain high luminance images with smooth motion. 'Mobile' sequence contains a complex still background and foreground motion. To demonstrate the efficiency of our proposed scheme, we have compressed 100 frames as INTRA of each sequence at 30 fps.



(a) (b) (c) (d)

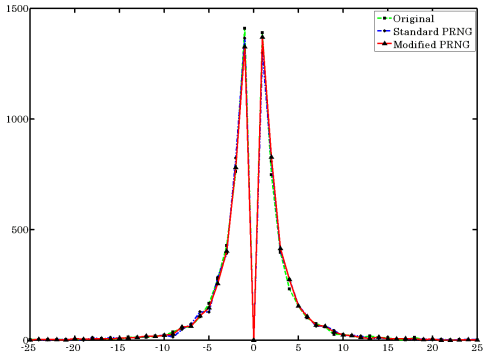
Fig. 7. Decoding of encrypted *foreman* image with: (a) QP = 18, (b) QP = 24, (c) QP = 30 and (d) QP = 36.

Table 2. Comparison of PSNR without encryption and with selective encryption (SE) for *foreman* sequence at different QP values.

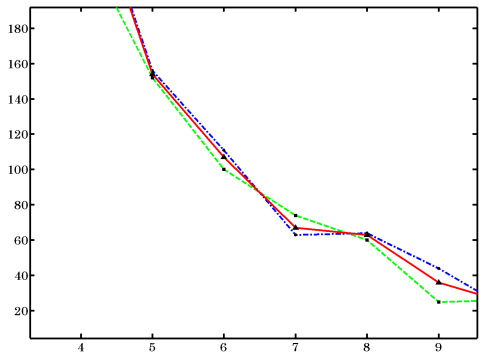
Sequence foreman	QP	PSNR (Y) (dB) Without encryption	PSNR (Y) (dB) With SE	PSNR (U) (dB) Without encryption	PSNR (U) (dB) With SE	PSNR (V) (dB) Without encryption	PSNR (V) (dB) With SE
	18	44.45	8.60	45.62	25.60	47.42	24.92
	24	39.41	9.26	41.70	28.01	43.86	25.78
	30	34.93	9.40	39.38	30.01	41.00	26.09
	36	30.78	10.31	37.33	34.76	38.10	30.22

Fig. 7 shows the encrypted video frames at different *quantization parameter* (QP) values of *foreman* video sequence. Their PSNR values are given in Table 2 and they are compared with the PSNR obtained for the same video frames without encryption. One can note that with increase in QP, the quality of the encrypted video increases. It is because of the fact that with higher QP, there are lesser NZs. Their magnitude is comparatively

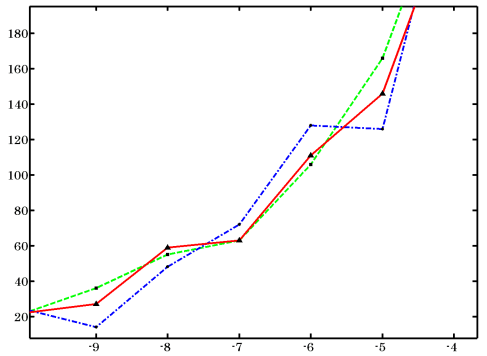
lirmm-00416013, version 1 - 11 Sep 2009



(a) Histogram of Transformed Coefficients.



(b) Magnified version of positive x-axis.



(c) Magnified version of negative x-axis.

Fig. 8. Comparison of histograms of original transformed coefficients, encrypted transformed coefficients using PRNG with equal probabilities and encrypted transformed coefficients using PRNG with their natural probabilities.

Table 3. Comparison of PSNR without encryption and with SE of benchmark video sequences at QP 18.

Sequence	PSNR (Y) (dB)		PSNR (U) (dB)		PSNR (V) (dB)	
	Without encryption	With SE	Without encryption	With SE	Without encryption	With SE
bus	44.28	7.81	45.22	25.52	46.51	25.62
city	44.30	9.93	45.84	27.24	46.76	28.36
crew	44.82	9.20	45.81	24.50	45.67	22.14
football	44.61	9.62	45.70	18.76	45.98	25.13
foreman	44.45	8.60	45.62	25.60	47.42	24.92
harbour	44.11	8.18	45.60	22.50	46.63	26.64
ice	46.58	9.81	48.70	23.45	49.18	18.96
mobile	44.46	8.11	44.14	14.82	44.05	12.17
soccer	44.32	8.78	46.59	24.10	47.82	25.93

smaller. Threshold to increment table is met less frequently. Consequently, more NZs uses VLC0 table and they are not encrypted. However, we can note that, whatever the QP value, PSNR of the SE video is lower than without encryption, even with a high QP value.

Table 3 compares the PSNR of all benchmark video sequences at QP value '18' without encryption and with SE. Our method is computationally very efficient. Although it depends on the contents of video and the quantization value, yet in proportion to overall computation which a video codec consumes, it is negligible.

Comparison of coefficient histograms before and after encryption is shown in Fig. 8. It can be seen that encrypted coefficients which uses PRNG with adaptive probabilities gives much better results and removes staircase problem. This graph contains only those non-zero levels which are encrypted. It does not contain those levels which are either coded as T1's or which are coded with table VLC0.

5. Conclusion

In this paper, a novel framework for SE of H.264/AVC based on CAVLC has been presented. Owing to unique probability for each NZ in the permutation space which is close to its natural probability, the histogram of encrypted NZs resembles the histogram of actual NZs. Real-time constraints have been handled successfully by having the same bitrate and by having compliant bit stream.

The experiments have shown that we can achieve the desired level of encryption in each frame, while maintaining the full H.264 video compression compliance, under a minimal set of computational requirements. The

lirmm-00416013, version 1 - 11 Sep 2009

proposed system can be extended to protect only ROI in the video[11] for videosurveillance and can be applied to medical image transmission.[12]

Acknowledgment

This work is in part supported by the VOODDO (2008-2011) project of the french ANR (Agence Nationale pour la Recherche).

References

1. Uhl, A. and Pommer, A., [*Image and Video Encryption: From Digital Rights Management to Secured Personal Communication*], Springer (2005).
2. Lian, S., Liu, Z., Ren, Z., and Wang, Z., "Selective Video Encryption Based on Advanced Video Coding," *Lecture Notes in Computer Science, Springer-Verlag* **3768**, 281–290 (2005).
3. Carrillo, P., Kalva, H., and Magliveras, S., "Compression Independent Object Encryption for Ensuring Privacy in Video Surveillance," in [*ICME*], (jun 2008).
4. Wu, C. and Kuo, C. J., "Design of Integrated Multimedia Compression and Encryption Systems," *IEEE Transactions on Multimedia* **7**(5), 828–839 (2005).
5. Jakimoski, G. and Subbalakshmi, K. P., "Cryptanalysis of Some Multimedia Encryption Schemes," *IEEE Transactions on Multimedia* **10**, 330–338 (April 2008).
6. Wiegand, T., Sullivan, G. J., Bjontegaard, G., and Luthra, A., "Overview of the h.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology* **13**, 560–576 (July 2003).
7. G.Bjontegaard and Lillevold, K., "Context-Adaptive VLC Coding of Coefficients," in [*JVT Document JVT-C028*], (may 2002).
8. Marpe, D., Schwarz, H., and Wiegand, T., "Context-based adaptive binary arithmetic coding in the h.264/avc video compression standard," *IEEE Transactions on Circuits and Systems for Video Technology* **13**, 620–636 (July 2003).
9. Smoot, S. and Rowe, L., "Study of DCT Coefficient Distributions," in [*Proc. of the SPIE Symposium on Electronic Imaging, volume 2657*], 403–411 (1996).
10. Altunbasak, Y. and Kamaci, N., "An Analysis of the DCT Coefficient Distribution with the H.264 Video Coder," in [*Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004 (ICASSP'04)*], **3**, 177–80 (may 2004).
11. Rodrigues, J.-M., Puech, W., and Bors, A., "Selective Encryption of Human Skin in JPEG Images," in [*Proc. IEEE Int. Conf. on Image Processing, Atlanta, USA*], 1981–1984 (Oct. 2006).
12. Puech, W. and Rodrigues, J., "A New Crypto-Watermarking Method for Medical Images Safe Transfer," in [*Proc.* 12^{th} *European Signal Processing Conference (EUSIPCO'04), Vienna, Austria*], 1481–1484 (2004).

lirmm-00416013, version 1 - 11 Sep 2009