Scrambling for Privacy Protection in Video Surveillance Systems

Frédéric Dufaux, *Member, IEEE*, and Touradj Ebrahimi, *Member, IEEE*

***Abstract*—In this paper, we address the problem of privacy protection in video surveillance. We introduce two efficient approaches to conceal regions of interest (ROIs) based on transform-domain or codestream-domain scrambling. In the first technique, the sign of selected transform coefficients is pseudorandomly flipped during encoding. In the second method, some bits of the codestream are pseudorandomly inverted. We address more specifically the cases of MPEG-4 as it is today the prevailing standard in video surveillance equipment. Simulations show that both techniques successfully hide private data in ROIs while the scene remains comprehensible. Additionally, the amount of noise introduced by the scrambling process can be adjusted. Finally, the impact on coding efficiency performance is small, and the required computational complexity is negligible.**

***Index Terms*—Privacy, selective encryption, surveillance, video processing.**

I. Introduction

VIDEO surveillance systems are omnipresent nowadays, with large systems in use in strategic places such as public transportation, airports, city centers, or residential areas. The prevailing sense of insecurity at the beginning of this century, with terrorist threats and high criminality, renders the intensive use of video surveillance tolerable despite its Orwellian big brother nature. However, people have a legitimate fear of this invasion of their personal privacy, with this objection slowing down a wider acceptance of video surveillance systems.

In this paper, we address the issue of privacy protection in video surveillance, with a goal to be able to conciliate the needs of video surveillance with the objection of privacy invasion. This issue has been previously addressed in [1]–[10].

In [1], the scene is represented using an object-based representation. Depending on the end-user access control authorizations, the system subsequently renders a modified version of the video where some objects are masked out. Hence, privacy-sensitive data is not transmitted. In [2], privacy filters, expressed using a privacy grammar, are introduced. These filters are applied on incoming video sensor data, preventing access to privacy-sensitive information.

In [3], it is postulated that face recognition techniques pose the threat to automatically identify people in a video surveillance scene, hence increasing the invasion of privacy. This issue is addressed by introducing an algorithm to de-identify faces such that many facial characteristics are preserved but the face cannot be reliably recognized.

Manuscript received November 14, 2007; revised March 8, 2008. First published July 9, 2008; current version published August 29, 2008. This paper was recommended by Guest Editor M.-T. Sun.

The authors are with Emitall Surveillance SA, CH-1820 Montreux, Switzerland (e-mail: frederic.dufaux@emitall.com; touradj.ebrahimi@emitall.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2008.928225

In [4], encryption is used to conceal faces. The process is invertible only for authorized users in possession of the secret encryption keys, hence preserving the privacy of people under surveillance. The problem of privacy for JPEG 2000 video [11] is tackled in [5] and [6]. More specifically, video analysis is applied to identify regions of interest (ROIs) corresponding for instance to people or faces. In [5], a wavelet-domain or codestream-domain conditional access control technique is proposed to subsequently scramble code-blocks corresponding to these ROIs. Alternatively, in [6], these same code-blocks are downshifted to the lowest quality layer of the codestream. By restricting the transmission bandwidth, the ROIs are decoded to a lower quality.

In [7], a region-based transform-domain scrambling technique is proposed to preserve privacy. ROIs are first estimated and the corresponding transform ac coefficients are scrambled by pseudorandomly inverting their signs, concealing any privacy-sensitive data. The approach is compatible with discrete wavelet transform (DWT) or discrete cosine transform (DCT), and the cases of Motion JPEG 2000 [11] and MPEG-4 [12] are considered.

The technique in [8] removes from the video information corresponding to authorized personnel. This privacy information is then hidden in the video using a compressed-domain watermarking technique based on a perceptual model and can only be recovered with a secrete key.

Finally, an MPEG-7 camera is proposed in [9] and [10] which features an embedded processor to perform video analysis. The camera outputs an MPEG-7 compliant data stream made of relevant descriptors which allow for video monitoring and surveillance without transmitting actual video data.

In this paper, we propose an extension of our earlier work [7]. We consider the case of MPEG-4 which is currently the most widespread standard in video surveillance. We first review the transform-domain scrambling approach initially introduced in [7]. We also show how it can be extended to scramble dc coefficients whenever a stronger scrambling is needed. Then, we introduce a new codestream-domain scrambling. Indeed, in the case of IP cameras, the incoming video stream is already compressed and it is advantageous to apply the scrambling process directly on the codestream in order to save computational complexity. This approach consists in parsing the codestream and in pseudorandomly inverting some of the bits corresponding to the ROI ac coefficients. In both the transform-domain and codestream domain approaches, the scrambling process depends on a secret encryption key which can be in possession of law-enforcement authorities who are consequently the only ones able to unlock and view the whole scene in clear. Finally, we perform a much more extensive and thorough performance study than in [7], considering not only the privacy protection capability and coding efficiency, but also a study of the security against both brute-force and error concealment attacks, and results in a real video surveillance setting.

Our proposed scrambling technique provides with a number of benefits when compared to competing approaches. First, the same scrambled codestream is transmitted to all terminals independently from their access rights. Unauthorized clients, who do not possess the secret key, can only view a distorted version of the content where privacy-sensitive data is concealed. Conversely, authorized clients, e.g., law-enforcement authorities, can unscramble the codestream and recover the truthful scene. Moreover, the scrambling process is flexible: the amount of noise injected can be adapted from slightly fuzzy to very noisy, and the scrambling can be confined to finely delineated ROIs. Finally, the impact in terms of coding efficiency is small, and it demands a low computational complexity.

This paper is structured as follows. We present the proposed transform-domain scrambling technique in Section II, whereas the alternative codestream-domain variation is introduced in Section III. To evaluate performance, experimental results are presented and discussed in Section IV. Finally, we draw some conclusions in Section V.

II. Transform-Domain Scrambling

Here, we describe the proposed transform-domain scrambling technique for privacy protection.

Hereafter, we consider more explicitly MPEG-4 [12], as it is the prevailing standard in current video surveillance equipments. However, the approach is straightforwardly extensible to all transform-coding techniques based on DCT such as Motion JPEG or AVC/H.264.

MPEG-4 is based on a motion-compensated (MC) block-based DCT [12]. More specifically, frames are coded as intra-frame, predictive-frame, or bidirectional-frame. In all cases, each frame is divided into 16×16 macroblocks (MBs). In turn, each MB is composed of four 8×8 luminance blocks and two 8×8 chrominance blocks. Each of these 8×8 blocks is DCT transformed, resulting in 64 DCT coefficients: one dc and 63 ac coefficients.

The scrambling can effectively be applied on the quantized DCT coefficients and outside of the MC loop, as illustrated in Fig. 1(a). This approach also guarantees that the scrambled video stream has a fully standard compliant syntax. Note that, in the encoder, unscrambled data are used in the MC prediction loop.

At the decoder side, authorized users perform unscrambling of the coefficients prior to the MC loop, as depicted in Fig. 1(b), for a fully reversible scrambling process as the same unscrambled data are used in the MC prediction loop as in the encoder. Conversely, unauthorized users are still able to correctly decode the video stream, except for the scrambled coefficients. However, in this case, scrambled data are used in the MC prediction loop, hence introducing a drift.

One of the challenges is to be able to correctly decode both the ROI and the background for unauthorized decoders due to the drift in the MC prediction loop. Note that this issue has typically not been addressed in previous conditional access control techniques, as it only happens when applying scrambling on given regions.

The scrambling process is based on a pseudorandom number generator (PRNG) initialized by a seed value. Multiple seeds

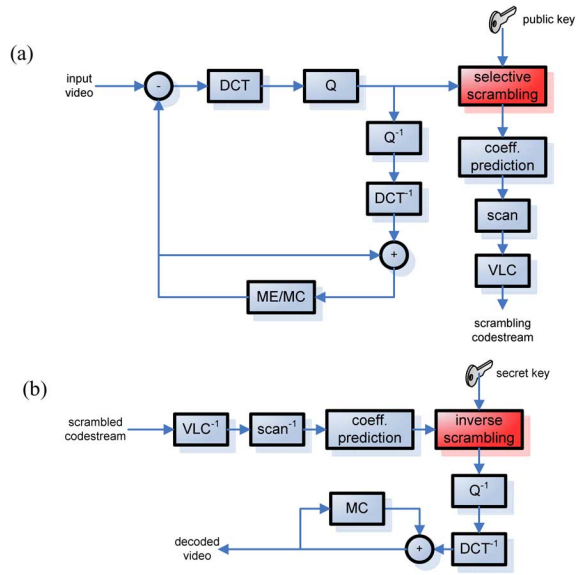

Fig. 1. Transform-domain scrambling in MPEG-4: (a) encoder/scrambler and (b) decoder/unscrambler.

can be used to strengthen the security. The seed values are then encrypted, preferably using asymmetric encryption, and inserted in the codestream, e.g., in private data. In our implementation, we define one seed value for a group of pictures, hence resulting in a negligible overhead. Authorized users, in possession of the secret encryption key, can recover the seed values and hence reproduce the same pseudorandom sequence to descramble the coefficients.

In order to unscramble the codestream, authorized decoders need to know the shape of the ROI. The latter has therefore to be transmitted as private data in the MPEG-4 codestream. In this paper, we consider a ROI defined on a MB basis as a good tradeoff between shape granularity and overhead. Transmitting the resulting binary mask, without compression, therefore requires 1 b per MB, corresponding to 396 b per frame for CIF format.

A. AC Coefficients

As previously stated, the scrambling process should not have a negative impact on coding efficiency. In general, dc coefficients are strongly correlated. A natural choice is therefore to apply scrambling to the ac coefficients. Furthermore, whereas the amplitude of ac coefficients is correlated, their signs are not. Finally, in MPEG-4, the length of codewords for ac coefficients remains unchanged if the sign of the coefficient is flipped.

Consequently, we propose to scramble the quantized ac coefficients of the blocks corresponding to the ROI by pseudorandomly flipping their sign

$$q\text{ACceff} = \begin{cases} -q\text{ACcoeff}, & \text{if } random\_bit = 1 \\ +q\text{ACcoeff}, & \text{otherwise.} \end{cases} \quad (1)$$

The amount of scrambling can be adjusted by restricting the scrambling to fewer ac coefficients. Straightforwardly, as the scrambling is merely flipping signs of selected coefficients, the technique requires negligible computational complexity.

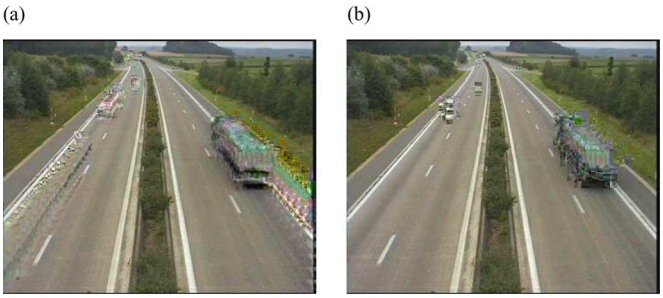

Fig. 2. Video scrambling: (a) normal MB-type decision resulting in drift and (b) modified MB-type decision removing drift.

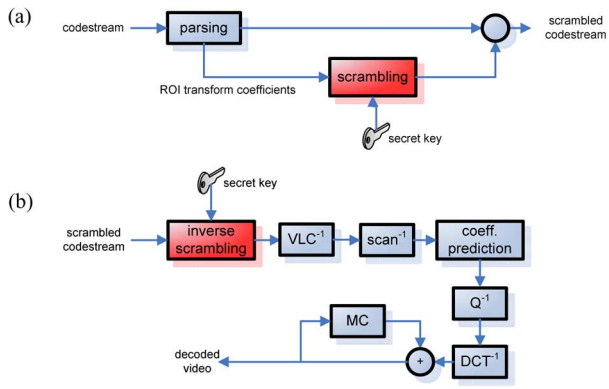

Fig. 3. Codestream-domain scrambling in MPEG-4: (a) transcoder/scrambler and (b) decoder/unscrambler.

B. DC Coefficients

Scrambling all ac coefficients usually provides with a sufficient level of concealment. In case a stronger scrambling is needed, it is also possible to additionally scramble dc coefficients.

In the case of Intra MB, the scrambling is done by pseudo-randomly altering the quantized dc coefficients as shown in (2) at the bottom of the page, where $\mathrm{DCscale}$ is the scaling factor for dc coefficients as defined in MPEG-4, which depends on the quantizer step size and the MB type. While this may decrease coding efficiency, the penalty is limited by the fact that relatively few MBs are intra-coded, and only a subset of those are actually scrambled. Nevertheless, even though few MBs are intra-coded, they carry a lot of information in a compressed video stream, and hence their scrambling has a significant impact on the amount of distortion introduced in the scrambled sequence.

In the case of inter MB, the scrambling is similar to the one for ac coefficients, namely the sign is pseudorandomly inverted as described in Section II-A.

C. MB-Type Decisions

From Fig. 1(b), it can be observed that an unauthorized decoder will use a different MC loop when compared with an authorized decoder. Consequently, this may lead to a drift and results in artifacts in the scrambled sequence, as depicted in Fig. 2(a). In order to remove this unwanted effect, the MB-type decision can be modified during encoding. More precisely, unscrambled MBs in the current frame, colocated with a scrambled MB in the reference frame, are always intra coded. This modification prevents the drift in MC loop and consequently removes the artifacts in the scrambled sequence, as shown in Fig. 2(b).

III. Codestream-Domain Scrambling

Nowadays, the trend in video surveillance systems is to utilize IP cameras. Namely, the incoming stream is already compressed when outputted by an IP camera. In this scenario, it is therefore beneficial to apply the scrambling directly in the codestream-domain as this saves computational complexity. The challenge of this approach is to produce a scrambled stream which still conforms to the standard syntax. One has therefore to be careful when pseudorandomly flipping bits of the codestream.

Hereafter, we introduce a new codestream-domain scrambling fulfilling this requirement. It avoids fully decoding and reencoding the video, but it still requires parsing the codestream in order to identify which bits correspond to the target syntax elements. The proposed scheme is illustrated in Fig. 3 for both the transcoder/scrambler and the decoder. Finally, note that while this codestream-domain scrambling approach shares some similarities with the transform-domain scrambling, it is very different from a system point of view.

A. AC Coefficients

In MPEG-4, DCT coefficients are coded using run-length. The first codewords of the variable length code (VLC) table to encode the intra chrominance ac coefficients and Inter luminance and chrominance $\mathrm{DC}+\mathrm{AC}$ coefficients is given in Table I. It can be observed that the codewords for a given coefficient value and its opposite are identical except for the last bit "s" which takes values "0" or "1" for a positive or negative value, respectively. The VLC table to encode the intra luminance ac coefficients has a similar structure and exhibits also this property. In other words, in the codestream-domain, the sign of an ac coefficient can be flipped by merely modifying the last bit "s" of the corresponding codeword.

Hence, ac coefficients scrambling can be effectively performed by pseudorandomly flipping the last bit "s" of the ac codewords, following the same philosophy and rational as in Section II-A. It is also possible to control the intensity of the scrambling by restricting the number of ac coefficients selected for scrambling.

$$qDCcoeff = \begin{cases} \left(qDCcoeff + \frac{2048}{2DCscale} - 2\right) \bmod \left(\frac{2048}{DCscale} - 2\right) + 1, & \text{if } random\_bit = 1 \\ qDCcoeff, & \text{otherwise} \end{cases} \quad (2)$$

TABLE I
VLC INTER LUMINANCE AND CHROMINANCE DC + AC COEFFICIENTS AND INTRA CHROMINANCE AC COEFFICIENTS

Index	Last	Run	Level	Bits	VLC code
0	0	0	1	3	10s
1	0	0	2	5	1111 s
2	0	0	3	7	0101 01s
3	0	0	4	8	0010 111s
4	0	0	5	9	0001 1111 s
5	0	0	6	10	0001 0010 1s
6	0	0	7	10	0001 0010 0s
7	0	0	8	11	0000 1000 01s
8	0	0	9	11	0000 1000 00s
9	0	0	10	12	0000 0000 111s
10	0	0	11	12	0000 0000 110s
11	0	0	12	12	0000 0100 000s
12	0	1	1	4	110s
13	0	1	2	7	0101 00s
14	0	1	3	9	0001 1110 s
15	0	1	4	11	0000 0011 11s
16	0	1	5	12	0000 0100 001s
…	…	…	…	…	…

Last: "0" there are more non-zero coefficients in this block, "1" this is the last non-zero coefficient in this block,
Run: the number of successive zeros preceding the coded coefficient,
Level: non-zero value of the coded coefficient, "s" denotes the sign of level, "0" for positive and "1" for negative.

Fig. 4. Transform-domain scrambling for *Hall Monitor*: (a) 63 ac coefficients and (b) dc + 63 ac coefficients.

Finally, note that, by design, this approach does not affect coding efficiency as the scrambling does not alter the number of bits in the codestream.

B. DC Coefficients

In MPEG-4, dc coefficients are differentially encoded. More precisely, each dc coefficient is predicted from the dc of the block above or the block on the left. The difference between the prediction and the actual coefficient is computed and this last value is encoded. For this reason, it is not possible to simply modify the differential DC value in the codestream as in the ac case.

As previously stated, it is usually not needed to scramble dc coefficients, as the scrambling of all ac coefficients result in a sufficiently strong protection. Nevertheless, if dc coefficient scrambling is preferred, then the same approach as in Section II-B can also be used. This requires parsing the codestream and to partially decode and reencode the unscrambled MBs which are using scrambled MB for prediction.

C. MB-Type Decisions

The same drifting phenomenon has discussed in Section II-C occurs also with the codestream-domain scrambling, and the same modification of the MB-type decision has to be used to resolve it. In the codestream-domain approach, it implies to parse the codestream and to partially decode and reencode some MBs. As shown in Section IV-B, this event occurs rarely in typical video surveillance sequences.

IV. SIMULATION RESULTS

Here, we evaluate the performance of the proposed region-based transform-domain and codestream-domain scrambling techniques in terms of privacy protection, coding efficiency, security, and complexity. Two video test sequences in CIF format are used: *Hall Monitor* and *Road*. Each sequence has a ground truth segmentation mask defining ROI. Experiments have been performed with the MPEG-4 MoMuSys Verification Model [13].

A. Privacy Protection

We first consider the capability of the scrambling technique to hide information in ROI of the video. Figs. 4 and 5 show results for the transform-domain scrambling.

It can be observed that the scrambling results in a strong distortion which is clearly sufficient to conceal ROI data so that people and objects can no longer be identified but the scene remains interpretable. Moreover, the additional scrambling of the dc coefficients brings an even stronger obscuration.

Results with the codestream-domain scrambling approach are similarly shown in Figs. 6 and 7. The same observations as for the transform-domain scrambling can be made.

B. Coding Efficiency

Another important criterion to evaluate the performance of the proposed scrambling techniques is coding efficiency. Indeed, it is important that coding performance is not adversely impacted. For this purpose, we compare the two cases when no scrambling is applied (i.e., corresponding to the original MPEG-4 VM codec) and when scrambling and unscrambling is performed (i.e., for an authorized client). The rate-distortion performances obtained with the transform-domain and codestream-domain scrambling, for the mode scrambling 63 ac coefficients, are given in Figs. 8 and 9, respectively.

It can be observed that the proposed scrambling has a negligible impact on coding efficiency. For *Hall Monitor*, the rate

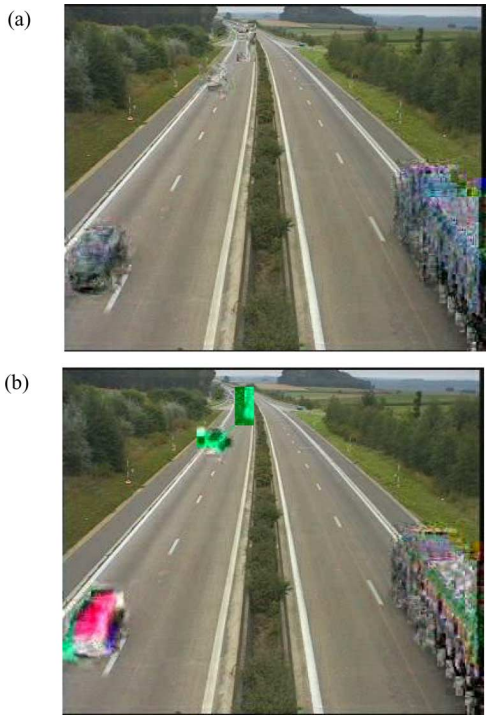

Fig. 5. Transform-domain scrambling for *Road*: (a) 63 ac coefficients and (b) dc + 63 ac coefficients.

Fig. 6. Codestream-domain scrambling for *Hall Monitor*: (a) 63 ac coefficients and (b) dc + 63 ac coefficients.

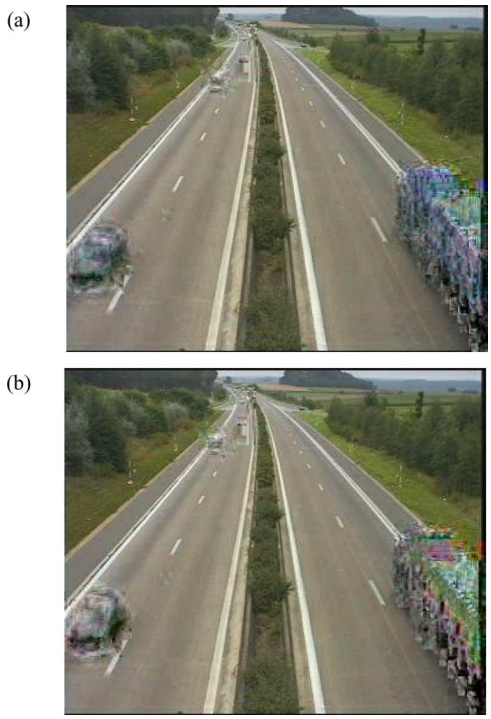

Fig. 7. Codestream-domain scrambling for *Road*: (a) 63 ac coefficients and (b) dc + 63 ac coefficients.

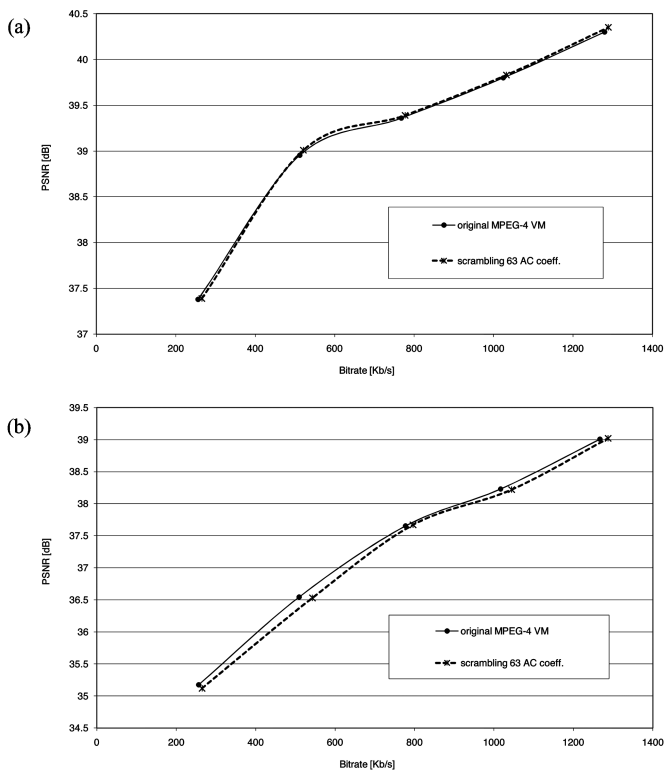

Fig. 8. Rate distortion coding efficiency comparison without and with transform-domain scrambling: (a) *Hall Monitor* and (b) *Road*.

increase is between 1%–4%, with most of the penalty due to the overhead to transmit the segmentation mask. While the coding efficiency loss is slightly larger for *Road*, it remains minimal with a rate increase between 1%–6%. In the latter case, the worse performance is due to the modification of the MB-type decision as described in Section II-C, which is forcing more MB to be intra coded for this sequence. This is confirmed by Table II, which shows the average number of MBs per frame

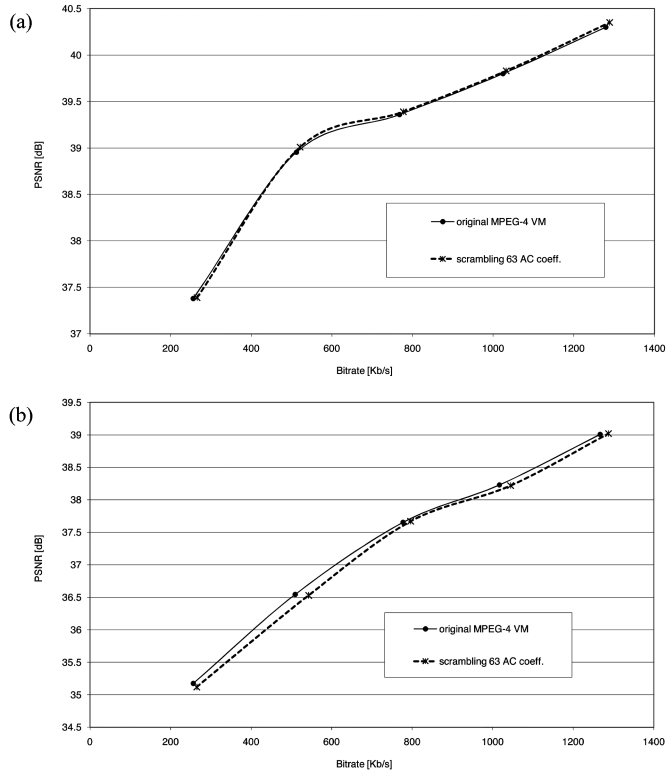

Fig. 9. Rate distortion coding efficiency comparison without and with codestream-domain scrambling: (a) *Hall Monitor* and (b) *Road*.

TABLE II
AVERAGE NUMBER OF MBS FORCED TO INTRA IN ORDER TO PREVENT DRIFT

Sequence	Average number of MB forced to Intra [per frame]
Hall Monitor	1.007
Road	3.067

forced to be intra coded in order to prevent drift. This event occurs three times per frame for *Road* on average, against once per frame for *Hall Monitor*. Nevertheless, it remains infrequent.

The transform-domain scrambling itself has a minimal impact on coding efficiency, as the length of codewords for ac coefficients remains identical whenever the coefficient sign is flipped. The codestream-domain scrambling itself does not modify the coding performance, as it only inverts some bits of the codestream.

C. Security

1) Brute-Force Attack: We now consider the security of the proposed scrambling technique against a brute-force attack. We address more specifically the mode scrambling all 63 ac coefficients. Note also that this analysis is identical for the transform-domain and codestream-domain approaches. Assuming that the attacker knows the ROI, we consider an exhaustive search of all combinations reversing the signs of all nonzero ac coefficients in the ROI. Table III reports the statistics for ROI data for the two test sequences, including the average number of nonzero ac coefficients in the ROI at 1 Mb/s and 256 kb/s.

TABLE III
STATISTICS OF ROI DATA

Sequence	Avg nb of MB in ROI [per frame]	Avg nb of non-zero AC coeff in ROI at 1Mb/s [per frame]	Avg nb of non-zero AC coeff in ROI at 256 kb/s [per frame]
Hall Monitor	29.9	1680	726
Road	44.6	2114	536

(a) (b)

Fig. 10. Error concealment attack: (a) *Hall Monitor* and (b) *Road.*

Even for the most vulnerable case, the sequence *Road* at 256 kb/s, an attacker has already to try reversing the signs of 536 coefficients in order to unscramble one frame, representing 2^{536} combinations. The security is even stronger at 1 Mb/s. Even though Table III represents a small sampling of typical video surveillance sequences, it gives a strong argument to confidently affirm that the proposed scrambling method provides with a good level of security against a brute-force attack.

2) Error Concealment Attack: Instead of the brute-force attack, an attacker may try an error concealment attack, which aims at concealing scrambled/encrypted data. Here, we assess the security of our proposed scrambling method against an attack where ROI ac coefficients are simply set to 0 at the decoder. In other words, this attack consists of extrapolating the scrambled data by motion compensation of the previous frame using the motion vectors which are available to the attacker. Note that we assume also that the ROI is known.

Fig. 10 shows the results of such an attack on the transform-domain scrambling when 63 ac coefficients are scrambled. We observe that this attack is inefficient. Furthermore, more sophisticated error concealment is unlikely to produce better results, as by definition the scrambled foreground objects have different characteristics when compared to the background and therefore cannot be extrapolated from the latter.

D. Scrambling in a Real Video Surveillance Setting

While in the previous sections we have shown results using some well-known video test sequences and associated ground-truth segmentation masks, we now substantiate the effectiveness of the proposed scrambling approach in a real video surveillance environment setting.

For this purpose, we show results on a video sequence captured in real surveillance situations with an analog camera. The video is then processed using a hardware device featuring a

(a) (b)

Fig. 11. Privacy protection scrambling in a real video surveillance setting: (a) original sequence and (b) with transform-domain scrambling.

real-time DSP implementation of background/foreground segmentation, MPEG-4 encoding and scrambling. The scrambling is using the transform-domain approach where 63 ac coefficients are pseudorandomly scrambled. Results are shown in Fig. 11. It can be observed that both pedestrians and moving cars are successfully scrambled.

V. Conclusion

In this paper, we have described a technique to address the issue of privacy in video surveillance. Regions of interest, assumed to correspond to privacy-sensitive data, are scrambled. In one approach, scrambling is applied in the transform-domain by pseudorandomly flipping the sign of transform ac coefficients or by pseudorandomly modifying dc coefficients. In a second approach, scrambling is performed directly on the compressed codestream by pseudorandomly inversing some bits corresponding to ac coefficients.

Simulation results show that the proposed scrambling techniques are successful at concealing privacy-sensitive information while leaving the scene comprehensible. The protection depends on a secret encryption key and the process is fully reversible for authorized users, e.g., law-enforcement authorities, in possession of the latter. After scrambling, the resulting codestream is still standard compliant. Simulations results show that this is obtained with a negligible impact on coding efficiency and a small computational complexity. Moreover, the scrambling process is flexible and the amount of distortion introduced can be adjusted from mere fuzzy to very noisy. Finally, the method is shown to be secured against brute-force or error concealment attacks.

Acknowledgment

The authors would like to express their gratitude to Y. Abdeljaoued for providing the results in Fig. 11. The authors would also like to thank the reviewers for their valuable comments, which have helped to improve this manuscript.

References

[1] A. W. Senior, S. Pankanti, A. Hampapur, L. Brown, Y.-L. Tian, and A. Ekin, Blinkering Surveillance: Enabling Video Privacy Through Computer Vision 2003, IBM Tech. Rep. RC22886.

[2] D. A. Fidaleo, H.-A. Nguyen, and M. Trivedi, "The networked sensor tapestry (NeST): A privacy enhanced software architecture for interactive analysis of data in video-sensor networks," in *Proc. ACM 2nd Int. Workshop Video Surveillance Sensor Networks*, New York, 2004, pp. 46–53.

[3] E. Newton, L. Sweeney, and B. Malin, Preserving Privacy by De-Identifying Facial Images Carnegie Mellon Univ., 2003, Tech. Rep. CMU-CS-03-119.

[4] T. E. Boult, "PICO: Privacy through Invertible Cryptographic Obscuration," in *Proc. IEEE/NSF Workshop Comput. Vis. Interactive Intell. Environ.*, Nov. 2005, pp. 27–38.

[5] F. Dufaux and T. Ebrahimi, "Video Surveillance using JPEG 2000," in *SPIE Proc. Applications of Digital Image Processing XXVII*, Denver, CO, Aug. 2004, pp. 268–275.

[6] I. P. Martinez, X. Desurmont, J. Meessen, and J.-F. Delaigle, "Robust human face hiding ensuring privacy," in *Proc. Int. Workshop Image Anal. Multimedia Interactive Services (WIAMIS)*, Montreux, Switzerland, Apr. 2005.

[7] F. Dufaux and T. Ebrahimi, "Scrambling for video surveillance with privacy," in *Proc. IEEE Workshop on Privacy Research In Vision*, New York, Jun. 2006, p. 160.

[8] W. Zhang, S. S. Cheung, and M. Chen, "Hiding privacy information in video surveillance system," in *Proc. IEEE Int. Conf. Image Process.*, Genova, Italy, Sep. 2005, pp. II-868–II-871.

[9] F. Dufaux and T. Ebrahimi, "Recent advances in MPEG-7 cameras," in *SPIE Proc. Applications of Digital Image Processing XXIX*, San Diego, CA, Aug. 2006, vol. 6312, pp. 631211.1–631211.8.

[10] [Online]. Available: http://www.eptascape.com/products/eptacam.htm

[11] D. Taubman and M. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Norwell, MA: Kluwer, 2002.

[12] T. Ebrahimi and F. Pereira, *The MPEG-4 Book*. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[13] *ISO/IEC 14496-7/DAM1 Optimized Reference Software for Coding of Audio-Visual Objects*, , Mar. 2003, ISO/IEC JTC1/SC29/WG11 WG11N5550.