ANTIFORENSIC SYNTHESIS OF MOTION VECTORS USING TEMPLATE ALGORITHMS

S. Milani, P. Bestagini, M. Tagliasacchi, and S. Tubaro

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano,

e-mail: {milani,bestagini}@elet.polimi.it,{marco.tagliasacchi,stefano.tubaro}@polimi.it

ABSTRACT

The identification of the video camera employed to acquire a video sequence is made possible by a large set of different footprints. Since video signals are always available in a compressed format, some of the most significant traces can be related to the coding tools of the implemented video codec (e.g., rate-distortion optimization, motion estimation strategy, etc.). As a matter of fact, an effective antiforensic attack, which aims at fooling the tools that identify the acquisition device, must appropriately alter these footprints. In the paper, we present an antiforensic strategy that targets a video camera detector which is based on the identification of the motion estimation strategy used by the video coder. The proposed approach synthesizes a set of motion vectors that approximate those that would have been generated by the algorithm to be mimicked. This method proves to be effective in attacking the detector while preserving the coding efficiency.

Index Terms— antiforensics, motion estimation, video codec identification, device detection, H.264/AVC.

1. INTRODUCTION

The authentication and validation of multimedia content often rely on the identification of the audio/video devices that acquired them [1]. This operation is usually performed looking for some distinctive features (called "footprints") that are left on the signal by the acquisition coding chain. Some of these clues refer to the specific acquisition hardware (e.g., PRNU patterns, Bayer masks, etc.); others are related to the specific processing operations that elaborate the signal from its acquisition to its final form (e.g., white balancing, interpolation, compression, etc.) [2].

These footprints are crucial since any additional modification of the acquired signal alters them in such a way that a correct validation is not possible any more. From these premises, an attacker could look for these traces and manipulate them in order to lead the tools of the forensic analyst to consider the analyzed content as authentically-generated by a specific device (different from the used one) [3]. Because of this possibility, multimedia forensic experts have recently focused on the investigation of antiforensic strategies that reveal the weak points of state-of-the-art solutions and suggest some possible remedies.

As for video signals, an important source of footprints is the video codec implementation[4] that is employed on the acquisition device. Because of the massive amount of data that characterize video signals, content is usually available in compressed format since its very acquisition [5]. Each camera vendor implements differently non-normative aspects of a video codec according to the hardware constraints, performance, intellectual property, final utilization of the product, and many other criteria.

Among these, motion estimation (ME) is one of the most significant non-normative tools of a video codec since it has a significant impact on the final quality of the reconstructed signal and on the required computational complexity. For this reason, camera vendors resort to fast motion estimation (FME) strategies that try to maximize the rate-distortion performance of the codec while minimizing the amount of operations. These algorithms are usually peculiar to a specific device and permit its identification [6]. An example is the approach in [7], which identifies the video codec by reprocessing the video signal with different FME algorithms and comparing the generated motion vector (MV) sets with the ones obtained from the bitstream. In this paper, we present an antiforensic strategy [8, 9, 10] that targets this forensic tool and show how it can be easily extended to other device detectors using MV statistics. The aim of the attacker is to create a coded video sequence that appears to be generated by a specific target device. Since an identifying features is the MV statistics, it is necessary to emulate the FME strategy adopted by the onboard video coder without the availability of its implementation (only a few template video sequences coded with the target FME are available). This counterfeit proves to be extremely useful in hiding traces of video forgeries [11] since the coder-related footprints of the video signal prove to be consistent as if no alteration was applied [12].

Given a target FME we want to emulate, the proposed antiforensic algorithm permits synthesizing, for the video sequence to be coded, a MV set close to the one that is generated by the FME to emulate. At the beginning, different template MV sets are generated by a pool of FME strategies available to the attacker. These MV estimates are then used to drive a Support Vector Regression (SVR) that generates the final MV values for the video sequence. The proposed estimator exploits the correlation between different algorithms and extends the analysis tools presented in [13]. Experimental results show that the proposed strategy permits fooling the detector quite effectively generating a video sequence whose rate-distortion performance is close to the one of the target FME.

In the following, Section 2 presents the detector adopted by the forensic analyst, while Section 3 describes the proposed antiforensic strategy. Section 4 reports the experimental performance obtained on a wide set of sequences, while Section 5 draws the final conclusions.

2. IDENTIFICATION OF THE FME VIA THE IDEMPOTENT PROPERTY

The considered FME strategy detector relies on the "idempotent" property of lossy coding. An operator is idempotent if reiterating

The project REWIND acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number:268478.

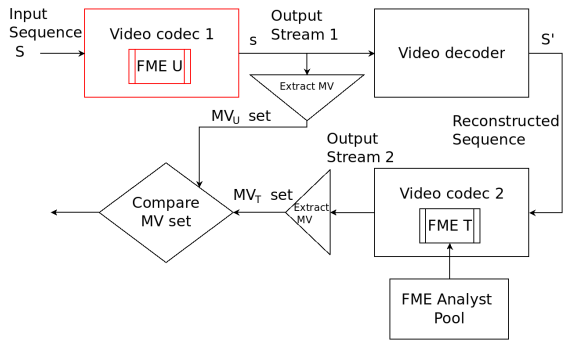

Fig. 1. FME detector using the idempotence property. The red block relates to the content creator, while black blocks relate to the forensic analyst.

its execution does not alter the output of the first iteration. In the literature, the idempotent property has been successfully exploited for the identification of the quantizer [14], the traces left by JPEG compression antiforensics [15], and the adopted video coding architecture [4]. The work in [7] addresses the problem of identifying the motion estimation algorithm by comparing the MV sets generated via different FME strategies and the one extracted from the compressed data.

A conceptual illustration of the adopted detection scheme is presented in Fig. 1. Let s denote the available data stream under analysis, which is the result of coding the video sequence S using the unknown FME strategy U. The analyst has available a set of $\mathcal{M}$ different FMEs. For each strategy $T \in \mathcal{M}$, he/she generates the output data stream s' recompressing the reconstructed sequence S' with video codec 2. In video codec 2, the adopted FME strategy is T. Then, the analyst tool extracts the motion vector sets MV_U and MV_T from s and s' and compares them via a difference function.

Motion vector sets can be related to a single or a group of frames from the sequence. In this paper, we adopt the same approach of [7], where the comparison is done frame-by-frame. We assume that sets MV_U and MV_T are made of elements $\mathbf{v}_n^U = (v_{n,x}^U, v_{n,y}^U)$ and $\mathbf{v}_i^T = (v_{n,x}^T, v_{n,y}^T)$, where n is the index of the motion compensation block in the analyzed frame ($n = 0, \ldots, N_B - 1$). Note that block partitioning is known to the analyst from the data stream s. Therefore, it is possible to force video codec 2 to use the same block partitioning adopted by codec 1, creating a one-to-one correspondence between MV pairs in s and MV pairs in s'. From this premise, we assume here that all the motion estimation blocks have the same size since extending the algorithm to blocks of variable size is quite straightforward and this simplification does not affect the generality of the algorithm.

In this work, the difference between MV_U and MV_T is characterized by the MSE computed between corresponding MV pairs, i.e.,

$$D(MV_U, MV_T) = \frac{1}{N_B} \sum_{n=1}^{N_B} \|\mathbf{v}_n^U - \mathbf{v}_n^T\|^2. \quad (1)$$

The detector will estimate that the unknown algorithm U correspond to the strategy T that minimize this difference, i.e.,

$$\hat{T} = \arg \min_{T \in \mathcal{M}} D(MV_U, MV_T). \quad (2)$$

Despite the minimum distortion is obtained whenever $T \equiv U$, the distortion measures $D(MV_U, MV_T)$ with $T \neq U$ provides significant information about the unknown algorithm. This fact has been used in the approach [13] where the measurements $D(MV_U, MV_T)$ computed from a set of selected algorithms T (called "eigenalgorithms") are employed as features that permit identifying the FME strategy even when this is not known or available to the analyst. The distances $D(MV_U, MV_T)$ characterize how similar the algorithms are and how they are related; following a principle similar to device localization via triangulation (used in wireless sensor networks and GPS systems), it is possible to identify the FME strategy in codec 1 [13].

In this work, we aim at investigating whether it is possible to reverse this process, thus synthesizing an MV set that is generated from sets MV_T, $T \in \mathcal{M}$, and proves to be close enough to MV_U.

Table 1. Algorithms used

Adopted algorithms			
T_1)	full s. (FS)	T_4)	new diamond s. (NDS) [18]
T_2)	spiral s. (SS)	T_5)	MVFAST [19]
T_3)	diamond s. (DS)	T_6)	UMHex [20]

3. SYNTHESIS OF MOTION VECTORS FROM TEMPLATES

In the previous section it was mentioned how distances $D(MV_U, MV_T)$ can be used in an FME detector to extend the limits of the "closed and known" set of detectable algorithms. In this section, it will be shown how the correlation between the MV set generated by an unavailable FME algorithm U and the MV sets generated by template strategies T can be used in an antiforensic attack against an idempotence-based FME detector. The aim of the attacker is to synthesize an MV set MV_Z for the coded video sequence s such that detector acknowledge it as if produced by algorithm U. This synthesis is performed by generating different MV sets MV_{T_k} using the template algorithms of the set $\mathcal{T} = \{T_k\}$, $k = 1, \ldots, N_T$. and calculating MV_Z from MV_{T_k} via a SVR regressor [16, 17]. The considered FME strategies are indexed in Table 1.

In this problem, we assume that the sizes of MV blocks are fixed to 8×8 pixels. The rate-distortion function can change for the different algorithms: its performance will be well modelled by the regression strategy we implemented.

We assume that a set of example sequences $s_i, i = 0, \ldots, N-1$, were coded using FME U and are available to the attacker. Moreover, the set $\mathcal{T}$ of template FME algorithms available to the attacker does not include U.

The approach can be divided into a training phase, where the correlation between U and the solution in $\mathcal{T}$ is learned, and a synthesis phase, where the coded sequence s is generated. In the end, the sequence is analyzed via the FME detector in [7] (described in Section 2).

3.1. Training phase

Given the sequence s_i, $i = 0, \ldots, N-1$, coded with the algorithm U to be synthesized, the analyst decodes and recompresses s_i into the sequences $s_{i,k}$ using the FME algorithm T_k. The coded bit stream is then parsed extracting the vectors $\mathbf{v}_n^{i,k} = (v_{n,x}^{i,k} v_{n,y}^{i,k})$, where the index n refers to the motion estimation block ($n = 0, \ldots, N-1$) and indexes x, y refer to the horizontal and vertical components, respectively.

Repeating the operation for all the algorithms in $\mathcal{T}$, it is possible to generate the arrays of features

$$\mathbf{f}_{n,x}^i = \left[v_{n,x}^{i,1} \ldots v_{n,x}^{i,N_T} \right], \ \mathbf{f}_{n,y}^i = \left[v_{n,y}^{i,1} \ldots v_{n,y}^{i,N_T} \right]. \quad (3)$$

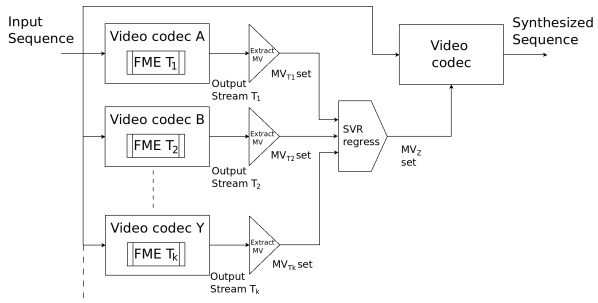

Fig. 2. Synthesis of coded sequence.

Parsing the bit stream s_i, it is possible to extract the vectors $\mathbf{v}_n^{i,U} = (v_{n,x}^{i,U} v_{n,y}^{i,U})$ generated by the algorithm U.

At this point, the attacker needs to implement the regression functions R_x and R_y that approximate $\mathbf{v}_n^{i,U}$, i.e.,

$$\hat{v}_{n,x}^{i,U} = R_x(\mathbf{f}_{n,x}^i) \rightarrow v_{n,x}^{i,U} \qquad \hat{v}_{n,y}^{i,U} = R_y(\mathbf{f}_{n,x}^i) \rightarrow v_{n,y}^{i,U} \tag{4}$$

In our implementation, this task has been accomplished using a SVR estimator [16]. Experimental results have proved that the accuracy of SVR estimation improves by splitting the regression into multiple context-related SVR estimator. To this purpose, the antiforensic attack distinguishes 6 different motion contexts characterized by the absolute values of motion vectors components found via the full search ME strategy (T_1), i.e.

$$\theta_c = \begin{cases} 0 & |v_{n,c}^{i,1}| \leq 6 \\ 1 & 6 < |v_{n,c}^{i,1}| \leq 16 \\ 2 & 16 < |v_{n,c}^{i,1}| \leq 32 \\ 3 & 32 < |v_{n,c}^{i,1}| \leq 64 \\ 4 & 64 < |v_{n,c}^{i,1}| \leq 128 \\ 5 & 128 < |v_{n,c}^{i,1}| \end{cases} \tag{5}$$

where index $c = x, y$ refers to the MV component.

The pairs $(v_{n,c}^{i,U}, \mathbf{f}_{n,c}^i)$, $(i = 0, \ldots, N-1, n = 0, \ldots, N_B - 1)$ are partitioned into 6 subsets according to the value of θ_c computed from the feature $v_{n,c}^{i,1}$. Within each subset, the attacker trains two SVR regressors $\hat{v}_{n,c}^U = S_{\theta,c}^U(\mathbf{f}_{n,c})$, $c = x, y$, assuming $v_{n,c}^{i,U}$ as target and $\mathbf{f}_{n,c}^i$ as input. The scheme of the whole synthesis strategy is reported in Fig. 2.

Note that in this process the attacker does not need to have much information about U. He/she only needs to have a sufficient number of sequences s_i coded with U that permit training the SVM regressor.

3.2. Synthesis phase

From this set of FME templates it is possible to generate a synthesized sequence s' where MVs are conforming to those generated by algorithm U. Given a sequence s whose MV field needs to be synthesized as if generated by algorithm U, N_T MV sets are generated recoding s with the algorithms in $\mathcal{T}$. For the n-th motion estimation block in s and MV component c, the antiforensic strategy computes the context θ_c and the array $\mathbf{f}_{n,c}^s$. These permit generating the synthesis motion vector component $\hat{v}_{n,c}^U$ processing $\mathbf{f}_{n,c}^s$ with $S_{\theta,c}^U$. The motion vector $\hat{\mathbf{v}}_n^U$ is then passed to a video codec that uses it to code the n-th block of sequence s (see Fig. 2).

This synthesis process proves to be quite effective in mimicking the unknown strategy U both considering the false positive rate obtained by the detector in [7] and by the rate-distortion performance of the synthesized algorithm. Both kinds of validation are reported in Section 4.

Table 2. Adopted sequences

Training	Test
`foreman, news, mobile, crew, soccer, bus, football, ice, paris, flower, highway, table, salesman, husky`	`container, coastguard, tempete, waterfall,`

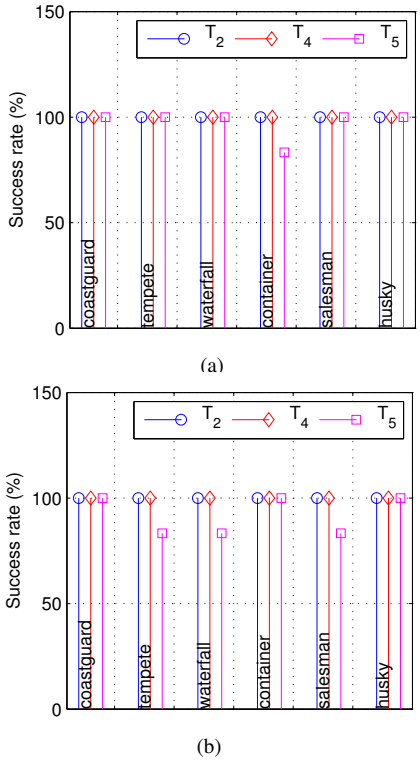

Fig. 3. False positive rate of idempotence detector for different sequences coded with different QP. a) QP=24 b) QP=32.

4. EXPERIMENTAL RESULTS

In order to test the effectiveness of the approach in synthesizing fake MV sets, we trained the SVR regressor using the set of algorithms $\mathcal{T}$ as templates and the sequences in Table 2 as templates. Video signals are in CIF resolution (352×288) and the adopted video codecs implement the standard H.264/AVC baseline profile. GOP structure is IPPP of length 15. In the tests we considered a single GOP of 15 frames in each sequence for the sake of complexity. However, the heterogeneity of processed video sequences make possible to generalize the results presented here.

Sequences in the column `Training` were used in the creation of regressors $S_{\theta,c}^U$ (s_i), while sequences in the column `Test` were synthetically generated (s). The performance of the antiforensic approach has been tested computing the false positive (FP) percentage for the ME detector in [7] processing a single frame from the sequence. This percentage corresponds to the success rate of the attacker since it counts the percentage of frames that were classified by the forensic analyst as generated by the algorithm U. Whenever the detected algorithm is different from the one that the attacker wants

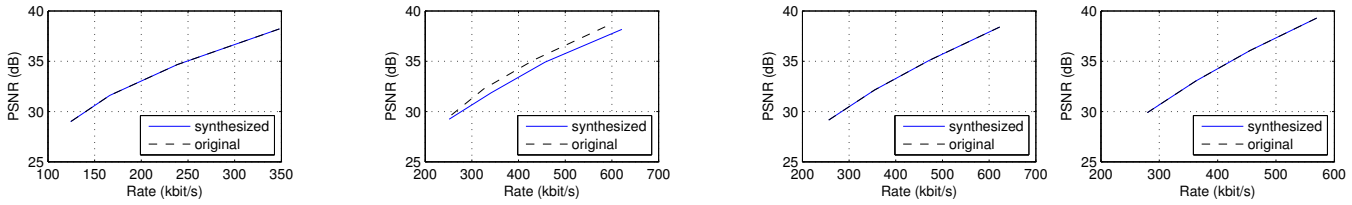

Fig. 4. Rate-Distortion performance on `coastguard` for synthesized algorithms T_2 (a), T_4 (b), T_5 (c) and on `container` for T_5 (d).

to synthesize, the proposed antiforensic attack fails.

In our tests, we initially assumed that the forensic analyst uses an algorithm set $\mathcal{T}_F$ to perform the idempotence-based FME detection, while the attacker uses the algorithm set $\mathcal{T}_A = \mathcal{T}_F \setminus \{U\}$ (where $U \in \mathcal{T}_F$ is the algorithm to be synthesized). Algorithms are referenced in Table 1.

Figure 3 reports the false positive rate obtained on different sequences in detecting algorithm T_2 (SS), T_4 (NDS [18]), and T_5 (MV-FAST [19]).

Note that the proposed antiforensic approach synthesizes algorithms T_2, and T_4 very well obtaining a success rate close to 100 % on all the sequences for $QP = 24$. As for algorithm T_5, the success rate decreases to 80 % for `container` since the amount of motion in the sequence is quite low and therefore, most of the algorithms generate MV sets that result highly correlated. As a matter of fact, in the detection phase the idempotence-based detector could estimate a distortion $D(MV_U, MV_{T_k}) < D(MV_U, MV_{T_5})$ for some frames (with $T_k \neq T_5$). This leads to a wrong algorithm detection. A similar effect can be noticed when the quantization noise increases. Figure 3 (b) reports the false positive percentages obtained with QP=32. In this case it is possible to see that success rates decrease for low motion sequences since MV estimation is made coarser by the compression noise. From these results, it is also possible to infer that the more the sequence presents a low amount of motion the more accurate the SVR regressor must be.

A second evaluation must consider the rate-distortion performance of the synthesized coder. Figure 4 reports the PSNR vs. rate plots for sequence `coastguard` comparing the data obtained from the original coder and those obtained for the synthesized strategy (points were obtained with QP=$20, 24, 28, 32$). It is possible to see that RD performance of the synthesized algorithm correspond to that of the original FME strategy to be mimicked. It is also possible to notice that strategy T_4 proves to be the most challenging to be synthesized (with respect to T_1 and T_5), i.e., it is quite difficult to fool the detector and keep the rate-distortion performance close to that of the original algorithm. Note that RD performance does not necessarily map directly to the success probability. To provide an evidence for this, we report the RD performance of algorithm T_5 for the sequence `container` (Fig. 4(d)). Note that the RD performance of the synthesized algorithm is quite close to that of the original solution despite success rate decreases to 80%.

The results that have been presented so far consider that $\mathcal{T}_A = \mathcal{T}_F \setminus \{T_U\}$, but it is possible to evaluate the performance of the approach whenever the amount of algorithms available to the analyst and the attacker changes significantly. Further analysis were devoted to investigate the effects of the number of template algorithms adopted by the attacker and the analyst on the final performance. To this purpose, Table 3 reports the average Bjontegaard ΔPSNR and ΔRate [21] (which measure the average difference between the interpolated rate-distortion curves), together with the average success probability, computed

Table 3. Success rate (%) with Bjontegaard ΔPSNR (dB) and ΔRate (%) for different analysis sets.

Algo $U = T_2$						
Sets	$\mathcal{T}_{A,1}$			$\mathcal{T}_{A,1}$		
	ΔPSNR	ΔRate	Succ.	ΔPSNR	ΔRate	Succ.
$\mathcal{T}_{F,1}$	0.00	+0.02	100.00	0.00	+0.01	100.00
$\mathcal{T}_{F,2}$	0.00	+0.02	100.00	0.00	+0.01	100.00
Algo $U = T_4$						
Sets	$\mathcal{T}_{A,1}$			$\mathcal{T}_{A,1}$		
	ΔPSNR	ΔRate	Succ.	ΔPSNR	ΔRate	Succ.
$\mathcal{T}_{F,1}$	−0.97	+8.70	100.00	−0.80	+7.48	100.00
$\mathcal{T}_{F,2}$	−0.97	+8.70	100.00	−0.80	+7.48	100.00
Algo $U = T_5$						
Sets	$\mathcal{T}_{A,1}$			$\mathcal{T}_{A,1}$		
	ΔPSNR	ΔRate	Succ.	ΔPSNR	ΔRate	Succ.
$\mathcal{T}_{F,1}$	0.00	0.00	100.00	0.00	0.00	100.00
$\mathcal{T}_{F,2}$	0.00	0.00	83.33	0.00	0.00	91.67

on the different test sequences of Table 2. Results were obtaining considering the possible sets $\mathcal{T}_{A,1} = \{T_1, T_2, T_4, T_5\}$ and $\mathcal{T}_{A,2} = \{T_1, T_2, T_3, T_4, T_5, T_6\}$ for the attacker and the sets $\mathcal{T}_{F,1} = \{T_2, T_4, T_5\}$, and $\mathcal{T}_{F,2} = \{T_2, T_3, T_4, T_5, T_6\}$ (note that T_1 (FS) is omitted since it is only used to create the context for regression). It is possible to notice that the cardinalities of $\mathcal{T}_F$ and $\mathcal{T}_A$ affect the false positive rate since the success rate for algorithm T_5 increases if $|\mathcal{T}_A| < |\mathcal{T}_F \setminus \{T_2\}|$. Moreover, the rate distortion performance of algorithm T_4 improves whenever more template solutions are available to the attacker. However, we must consider that the RD performance of the algorithm is not available to analyst (since it depends on the characteristics of originally-acquired sequence), and therefore, we believe that success rate plays a more important role for the attacker. In the end, it is possible to conclude that the conflict between attacker and forensic analyst is resolved in favor of which one employs more template algorithms.

5. CONCLUSION

The paper presented an antiforensic strategy targeting the detection of the FME strategy employed in the coding process of a video sequence under analysis. The proposed solution generates a motion vector sets using an SVR regressor that takes in input the motion vectors generated by a set of template algorithms. By exploiting the correlations between different motion estimation algorithms, the attacker synthesizes a coded video sequence that well approximate the one that would have been generated by the algorithm to be mimicked (both from the point of view of the detector and in terms of rate-distortion performance).

6. REFERENCES

[1] S. Milani, M. Fontani, P. Bestagini, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro, "An overview on video forensic," *APSIPA Transactions on Signal and Information Processing*, vol. 1, 2012, Available at http://journals.cambridge.org/abstract_S2048770312000029.

[2] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of cfa artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, pp. 1566–1577, 2012.

[3] M. Barni and B. Tondi, "The source identification game: An information-theoretic perspective," *IEEE Transactions on Information Forensics and Security*, vol. 8, pp. 450–463, 2013.

[4] P. Bestagini, A. Allam, S. Milani, M. Tagliasacchi, and S. Tubaro, "Video codec identification," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012.

[5] S. Milani, P. Bestagini, M. Tagliasacchi, and S. Tubaro, "Multiple compression detection for video sequences," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2012.

[6] P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro, "Video codec identification extending the idempotency property," in *European Workshop on Visual Information Processing (EUVIP)*, 2013.

[7] M. Sorell, "Video provenance by motion vector analysis: A feasibility study," in *International Symposium on Communications, Control, and Signal Processing (ISCCSP)*, 2012.

[8] M.C. Stamm and K.J.R. Liu, "Anti-forensics of digital image compression," *IEEE Transactions on Information Forensics and Security*, vol. 6, pp. 1050–1065, 2011.

[9] M.C. Stamm, W.S. Lin, and K.J.R. Liu, "Temporal forensics and anti-forensics for motion compensated video," *IEEE Transactions on Information Forensics and Security*, vol. 7, pp. 1315–1329, 2012.

[10] S. Milani, M. Tagliasacchi, and S. Tubaro, "Antiforensics attacks to Benford's law for the detection of double compressed images," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[11] P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro, "Local tampering detection in video sequences," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2013.

[12] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double mpeg compression," in *ACM Workshop on Multimedia and Security (MM&Sec)*, 2006.

[13] S. Milani, M. Tagliasacchi, and S. Tubaro, "Identification of the motion estimation strategy using eigenalgorithms," in *IEEE International Conference on Image Processing (ICIP)*, 2013.

[14] Z. Zhu and T. Lin, "Idempotent H.264 intraframe multi-generation coding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009.

[15] G. Valenzise, V. Nobile, M. Tagliasacchi, and S. Tubaro, "Countering JPEG anti-forensics," in *IEEE International Conference on Image Processing (ICIP)*, sept. 2011, pp. 1949 –1952.

[16] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[17] H. Drucker, C.J.C Burges, L. Kaufman, A.J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems (NIPS)*, 1996.

[18] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Transactions on Image Processing*, vol. 9, pp. 287–290, 2000.

[19] P. De Pascalis, L. Pezzoni, G.A. Mian, and D. Bagni, "Fast motion estimation with size-based predictors selection hexagonal search in H.264/AVC encoding," in *European Signal Processing Conference (EUSIPCO)*, 2004.

[20] T. Wiegand, "Version 3 of H.264/AVC," in *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6),* 12^{th} *Meeting*, Redmond, WA, USA, July 17 – 23, 2004.

[21] G. Bjontegaard, "Calculation of average psnr differences between rd-curves (vceg-m33)," in *presented at the* 13^{th} *ITU VCEG Meeting*, Austin, TX, USA, Apr. 2 – 4, 2001, VCEG-M33.