IDENTIFICATION OF THE MOTION ESTIMATION STRATEGY USING EIGENALGORITHMS

S. Milani, M. Tagliasacchi, S. Tubaro

Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milan, Italy
e-mail: milani@elet.polimi.it,marco.tagliasacchi@polimi.it
stefano.tubaro@polimi.it

ABSTRACT

The identification of the device, or device model, that was used to acquire a video sequence is a very challenging task, since it has to rely on subtle traces left by the processing steps applied to the raw acquired data. Previous works have tried to address this problem leveraging the traces left by the imaging sensor. However, in the case of video, lossy coding is often quite aggressive, thus making these methods impractical. In this work, we reverse the analysis strategy and exploit the traces left by lossy coding as telltale for the adopted acquisition device. Specifically, we aim at detecting the implementation of the video codec by identifying the adopted motion estimation algorithm. Indeed, motion estimation is not defined in video coding standards and, as such, it represents one of the non-normative tools that can be customized in the design of the encoder. The key tenet consists in studying the correlation between the motion vectors obtained from the decoded bitstream, and those computed using a set of known and diverse motion estimation algorithms, called eigenalgorithms. In our work, we generalize a method recently appeared in the literature, which assumes that the motion estimation algorithm used is necessarily one of those available during the analysis. Experimental results show that the approach is able to successfully identify the motion estimation algorithm in most cases.

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

1. INTRODUCTION

The identification of the source, e.g., the device (or device model) used to acquire visual content, is one of the main research challenges in the area of multimedia forensics. Indeed, device identification might be useful for the authentication and validation of visual content, or for tracing back its origins. To address this problem, the methods proposed in the literature analyze the distinctive traces (footprints) left by the processing steps applied during the acquisition-processing-coding chain that characterizes consumer devices like digital cameras, camcorders and smart-phones [1].

In the past, several different footprints related to the acquisition phase have been thoroughly investigated. In most cases, though, visual content is distributed in a compressed format. On the one hand, lossy coding conceals the traces left by the sensor, so that acquisition-based footprints might not be easily identified. On the other hand, it introduces other characteristic footprints, which can be exploited to identify the device. This is a particularly relevant in the case of video, since sequences might be encoded with a wide variety of available coding standards (e.g., MPEG-2, H.264/AVC, etc.), and each encoder follows a specific implementation of those tools that are not explicitly defined by the standard. These non-normative tools include, e.g., the algorithms to perform motion estimation, coding mode decision, rate control, etc. It is often the case that different vendors adopt proprietary implementations, which might also differ depending on the specific device model.

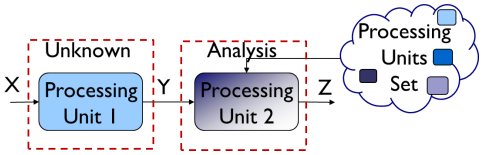


Fig. 1. Processing and analysis chain using idempotence property.

In this work, we focus on the identification of the adopted motion estimation (ME) algorithm as distinctive feature of a specific codec implementation. The high computational complexity of ME and its impact in terms of coding efficiency have led to the design of several different instances of fast motion estimation (FME) algorithms, each tuned to the specific hardware (video-processor) embedded in the acquisition device. Indeed, the identification of the ME algorithm adopted at the encoder, based on the available compressed video sequence, might be used by the forensic analyst to identify the acquisition device or vendor.

The proposed solution relies on the "idempotent" property of lossy coding. An operator is idempotent if reiterating its execution does not alter the output of the first iteration. In practical multimedia forensics cases, the output of the n-th iterations is highly correlated with the output of the first one. A conceptual illustration of the proposed identification scheme based on the idempotent property is presented in Fig. 1. Let Y denote the available data under analysis, which is the result of processing X with the unknown *Processing Unit 1*. The analyst has available a set of processing units. For each of them, he generates the output data Z by means of the *Processing Unit 2* and measures the correlation between Y and Z. The processing unit that led to the highest measure of correlation is identified as the one used to output the observed data Y.

In the literature, the idempotent property has been successfully exploited for the identification of the quantizer [2], the traces left by JPEG compression antiforensics [3], and the adopted video coding architecture [4]. Similarly to this paper, the work in [5] addresses the problem of identifying the motion estimation algorithm. However,

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.

in all aforementioned works, a successful detection is possible only if the analyst perfectly knows all the possible instances of processing units (e.g., quantization matrices, video coding standards, motion estimation algorithms) that can be adopted to process the signal, together with their operating parameters. This condition is normally indicated as "closed and known set". Unfortunately, the forensic analyst operates under more challenging and adverse conditions, since he might not know, or have access to, the possible instances of the processing units. For the problem of determining the motion estimation algorithm, we show in this paper that it is possible to design an identification strategy which is able to discriminate different algorithms, even when their implementations are not known to the analyst.

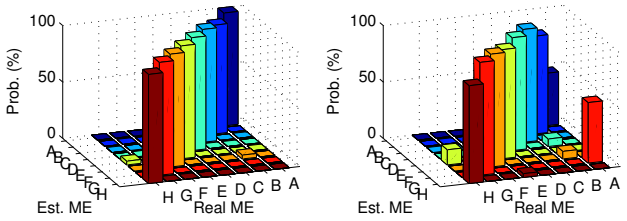
In the following, Section 2 illustrates an overview of the proposed method. Section 3 summarizes the method in [5], which works under the the closed and known set condition. The proposed method is described in Section 4. Section 5 show the experimental results and Section 6 discusses the future developments.

2. OVERVIEW OF THE PROPOSED METHOD

The proposed method receives as input the bitstream of a video sequence. The bitstream is decoded to obtain the motion vector (MVs) estimated by the encoder (i.e., *Processing Unit 1*, in Fig. 1) and the reconstructed video sequence in the pixel domain. Then, the forensic analyst processes the decoded sequence by means of different motion estimation algorithms (i.e., Processing Units Sets in Fig. 1), thus producing different sets of MVs. The observation tenet is that the motion estimation algorithm satisfies (at least approximately) the idempotent property. That is, if the motion estimation algorithm used by the forensic analyst matches the one used by the original encoder, the MVs will be very similar to those available from the bitstream. Otherwise, the similarity is weaker. In a way, the motion vectors are (approximately) preserved when the same algorithm is adopted. The forensic analyst has access to a finite set of motion estimation algorithms. Among those, he selects a subset of algorithms, called *eigenalgorithm* in analogy with eigenvectors or eigenfunctions, that are meant to provide a discriminative representation of the wide variety of existing motion estimation algorithms, including those that are not available to him. As such, each analyzed sequence is mapped to a point in the coordinate system defined by the eigenalgorithms. Sequences processed by the same encoder will be mapped to points that are clustered together, thus providing a sort of "signature" of the employed motion estimation algorithm.

In some cases, the available bitstream might be the result of an encoder adopting a motion estimation algorithm which does not belong neither to the set of eigenalgorithms, nor to the algorithms known to the analyst, e.g., because the proprietary implementation of the algorithm is not made available by the device vendor. In this adverse case, it is still possible to measure the degree of similarity of the motion vectors extracted from the bitstream, with those obtained by means of the eigenalgorithms, thus relaxing the limiting assumption of "closed and known set".

Note that despite the name, eigenalgorithms do not form an *orthogonal* set. Indeed, all motion estimation algorithms seek to achieve the same goal, i.e., minimizing the energy of prediction residuals. Hence, they might share some common elements. As a consequence, if a sequence was encoded with one of the available eigenalgorithms, the available motion vectors might be similar (although to a lesser extent) to those computed by means of other eigenalgorithms.



(a) (b)

Fig. 2. Ambiguity matrices for different ME strategies and QP values. a) QP=20 b) QP=28.

3. IDENTIFYING KNOWN MOTION ESTIMATION ALGORITHMS

In this section, we consider the simpler case in which the forensic analyst has access to all the possible motion estimation algorithms adopted by the encoders used to generate the available sequences. In general, the set of motion vectors estimated by an encoder depends on: i) the characteristics of motion in the sequence, and ii) the adopted motion estimation algorithm. In this paper, we consider motion estimation by block matching, which is widely adopted in all video coding standards. Given a block of pixels, a motion estimation algorithm searches for a similar block in the previously coded frames, to be used as predictor. Depending on the adopted motion estimation algorithm, search might be limited to different sets of candidate blocks. For example, many motion estimation algorithms perform an initial search considering a set of possible candidates lying close to the current block location (or close to the locations pointed by MVs of neighboring blocks). In addition, the search pattern can be modified during the estimation process. Other algorithms build a statistical model for all the possible candidates and choose the most probable ones.

Let X denote an uncompressed video sequence and $\mathcal{M}$ a finite set of m possible motion estimation algorithms. The sequence X is compressed with a video encoder that adopts the motion estimation algorithm $\bar{M}$. Let $\hat{X}$ denote the corresponding video sequence reconstructed at the decoder and $\bar{\mathcal{V}} = \{\bar{\mathbf{v}}(i,t), i = 1,\dots,N_B, t = 1,\dots,N_F\}$ the set of motion vectors extracted from the bitstream. Note that the size N_B of blocks used by motion estimation is available, since this can be determined by decoding the bitstream. In order to identify the motion estimation algorithm, the forensic analyst processes the available sequence $\hat{X}$ with each of the motion estimation algorithms in $\mathcal{M}$, thus producing m different sets of motion vectors. The identification of the motion estimation algorithm can be performed, according to [5], by looking for the one that produces the motion vectors to be the most similar to $\bar{\mathcal{V}}$. That is,

$$\hat{M} = \arg\min_{M\in\mathcal{M}} \frac{1}{N_F}\sum_{t=1}^{N_F} D(\mathcal{V}_t, \bar{\mathcal{V}}_t) \tag{1}$$

where $\bar{M}$ is assumed to be included in $\mathcal{M}$ and $D(\mathcal{V}_t, \bar{\mathcal{V}}_t)$ is the distance measure between the sets of motion vectors $\mathcal{V}_t$ and $\bar{\mathcal{V}}_t$ corresponding to the t-th frame. The distance is computed as

$$D(\mathcal{V}_t, \bar{\mathcal{V}}_t) = \frac{1}{N_B}\sum_{i=1}^{N_B} \|\mathbf{v}(i,t) - \bar{\mathbf{v}}(i,t)\|^2. \tag{2}$$

In general, the identification accuracy is affected by different factors: i) the characteristics of the video content, i.e., the complexity

of motion (indeed, for static sequences, most algorithm will estimate motion vectors which are very similar to each other and close to zero); ii) the intrinsic similarities that might exist among the algorithms in the set $\mathcal{M}$; iii) the amount of distortion introduced by encoding the original sequence X to $\hat{X}$. Figure 2 illustrates the identification results in the form of a confusion matrix when the motion estimation algorithm is sought using (1), and the set of possible algorithms includes $m = 8$ elements, as indicated in Table 1. The sequences available to the forensic analyst were encoded using H.264/AVC at two different quality levels corresponding to, respectively, $QP = 20$ and $QP = 32$. It is possible to notice that, as the quality decreases (i.e., the QP value increases), the probability of wrong detection (slightly) increases for some algorithms. This is due to the fact that the encoder performed motion estimation on the original uncompressed video sequence X. Conversely, the forensic analyst performs motion estimation on the decoded sequence $\hat{X}$. The coding artifacts introduced at higher values of QP are such that applying the same motion estimation algorithm $\bar{M}$ might lead to different motion vectors.

However, in many practical situations, the forensic analyst might not have access to an implementation of the motion estimation algorithm $\bar{M}$ used to encode the available sequence, that is $\bar{M} \notin \mathcal{M}$. In the next section, we show that it is still possible to discriminate the adopted motion estimation algorithm, representing each sequence in a vector space defined by a known set of eigenalgorithms.

4. IDENTIFYING KNOWN AND UNKNOWN MOTION ESTIMATION ALGORITHMS

In our work, we consider the general case in which a video sequence is encoded using a motion estimation algorithm $\bar{M} \in \mathcal{U}$, where $\mathcal{M} \subset \mathcal{U}$. Hence, $\bar{M}$ might, or might not, belong to the set of algorithms $\mathcal{M}$ available to the forensic analyst.

We approach the problem in two steps. First, we assume that the subset $\mathcal{N} \subset \mathcal{M}$ of eigenalgorithms is given ($|\mathcal{N}| = n < m = |\mathcal{M}|$), and we show how it is possible to learn a classifier that, given a sequence, determines whether it was encoded with one of the known algorithms in $\mathcal{M}$ or by another algorithm in $\mathcal{U} \setminus \mathcal{M}$. In the former case, we also show how we can determine which was the adopted motion estimation algorithm. Then, we focus on how to select the subset of eigenalgorithms $\mathcal{N}$.

Given a set of eigenalgorithms $\mathcal{N} = \{N^{(1)}, N^{(2)}, \ldots, N^{(n)}\}$, the forensic analyst prepares a set of sequences, each encoded multiple times with one of the motion estimation algorithms in $\mathcal{N}$. Let $\hat{X}$ the sequence decoded from one of the generated bitstream and $\bar{M} \in \mathcal{M}$ the corresponding motion estimation algorithm used at the encoder. The sequence $\hat{X}$ is processed to compute the motion vectors for all algorithms in $\mathcal{M}$. Hence, each frame t is mapped to a vector in a n-dimensional space, which is defined as follows

$$\mathbf{d}_t = \left[D(\mathcal{V}_t^{(1)}, \bar{\mathcal{V}}_t), \ldots, D(\mathcal{V}_t^{(n)}, \bar{\mathcal{V}}_t)\right]^T \in \mathbb{R}_+^n \qquad (3)$$

In case $\bar{M} = M^{(j)} \in \mathcal{N}$, the j-th component of the vector $\mathbf{d}_t$ is likely to be smaller than the others. Instead, in case $\bar{M} \in \mathcal{M} \setminus \mathcal{N}$, we observed that sequences encoded with the same motion estimation algorithm tend to be clustered together in this n-dimensional space defined by the eigenalgorithms. This is illustrated in Figure 3, which shows how a set of sequences encoded with one of the motion estimation algorithm in $\mathcal{M} = \{A, B, C, D, E, F\}$ is mapped to the n-dimensional space defined by the $n = 3$ eigenalgorithms $\mathcal{N} = \{A, B, E\}$. For each algorithm in $\mathcal{M}$, we define a region

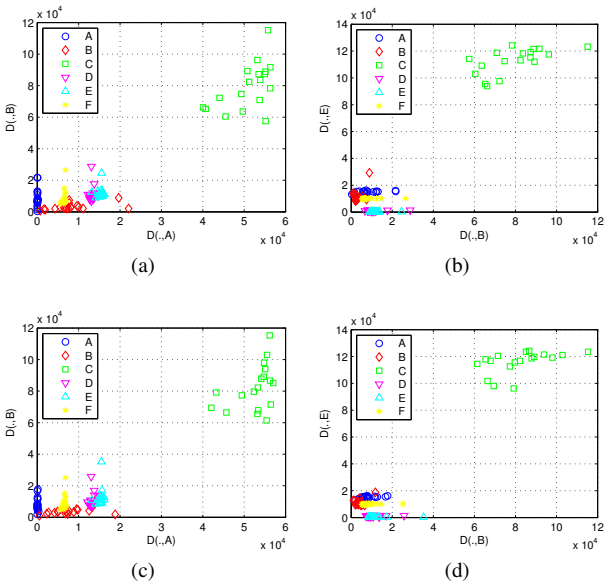


Fig. 3. Values of $\mathbf{d}_t$ for different algorithms where $\mathcal{M}_e = \{A, B, E\}$ with QP=20 (subfigures a,b) and QP=28 (subfigures c,d). Points $\mathbf{D}(X)$ are plotted in two separate graphs $D(\cdot, A)$-vs.-$D(\cdot, B)$ (a,c) and $D(\cdot, B)$ vs. $D(\cdot, E)$ (b,d). The algorithms referred by labels are reported in Table 1.

$R^{(i)} \in \mathbb{R}_+^n$, $i = 1, \ldots, m$, that can be represented by means of its centroid

$$\mathbf{c}^{(i)} = E[\mathbf{d}_t | M^{(i)}], i = 1, \ldots, m, \qquad (4)$$

where $E[\cdot|M^{(i)}]$ represents the average of vectors $\mathbf{d}_t$ computed from sequences encoded using $M^{(i)}$. For every vector $\mathbf{d}_t$ we also compute the reliability score $\sigma(\mathbf{d}_t)$ defined as the standard deviation of quantities $\|\mathbf{d}_t - \mathbf{c}^{(i)}\|_2$ with respect to all the centroids $\mathbf{c}^{(i)}$, i.e.,

$$\sigma(\mathbf{d}_t) = dev\left[\|\mathbf{d}_t - \mathbf{c}^{(i)}\|\|_2\right]_{i=1,\ldots,m}, \qquad (5)$$

which indicates the average affinity of $\mathbf{d}_t$ to the identified regions. The purpose of this reliability metric is to find out whenever the point $\mathbf{d}_t$ is not sufficiently close to one of the centroids to permit a reliable classification. As a consequence, it is possible to associate to each region $R^{(i)}$ a confidence level $r^{(i)} = E[\sigma(\mathbf{d}_t)|M^{(i)}]$.

4.1. Classification procedure

Given a set of n eigenalgorithms $\mathcal{N}$ and a set of regions $R^{(i)}$, $i = 1, \ldots, m$, a frame of an input sequence $\hat{X}_t$, for which the motion estimation algorithm is to be determined, is mapped to a vector $\mathbf{d}_t$ as shown in (3) and its corresponding reliability $\sigma(\mathbf{d}_t)$. If there is a region $R^{(k)}$ such that $k = \arg\min_i \|\mathbf{d}_t - \mathbf{c}^{(i)}\|_2$ (mapping to the closest centroid) and $\sigma(\mathbf{d}_t) > \epsilon r^{(k)}$ (reliability check), the classifier estimates that the algorithm $M^{(k)}$ was adopted to code the current frame. Otherwise, the frame is deemed to be generated by means of a motion estimation algorithm in $\mathcal{U} \setminus \mathcal{M}$. Note that, in this case, if a large enough number of frames is observed, it is possible to use the corresponding vectors as seeds for computing the $(m+1)$-th region. This process can be further iterated as more sequences (and motion estimation algorithms) are observed.

Table 1. Sequences and algorithms used

$\mathcal{M}$		$\mathcal{U} \setminus \mathcal{M}$	
A)	full s. (FS)	G)	new diamond s. (NDS) [6]
B)	spiral s. (SS)	H)	4SS [7]
C)	diamond s. (DS)		
D)	circular s. (CS) [8]		
E)	MVFAST [9]		
F)	UMHex [10]		

$\mathcal{S}_T$			$\mathcal{S}_e$
coastguard	crew	football	bus
soccer	tempete	harbour	city
flower	mobile	ice	table

4.2. Selecting the eigenalgorithms

We are left with the problem of determining which is the most appropriate set of n eigenalgorithms $\mathcal{N}$. Intuitively, we want the eigenalgorithms to be representative and diverse, so that they will enable to discriminate the algorithms in $\mathcal{M}$, as well as to determine when another algorithm in $\mathcal{U} \setminus \mathcal{M}$ is used. We proceed by enumerating all the possible subsets of n algorithms in $\mathcal{M}$. For each candidate subset of eigenalgorithm, we compute a measure of the discriminative power of this subset. To this end, we consider a set of sequences encoded with algorithms in $\mathcal{M}$. Hence, having fixed $\mathcal{N}$, we learn the m regions as described above. Then, we consider a different set of sequences, this time encoded with either a known algorithm in $\mathcal{M}$, or an unknown algorithm in $\mathcal{U} \setminus \mathcal{M}$. Using these sequences we compute the classification error rate, i.e., the fraction of times that the correct motion estimation algorithm is identified. In case of algorithms in $\mathcal{U} \setminus \mathcal{M}$, a correct classification occurs when the the vector is not assigned to either one of the m regions.

5. EXPERIMENTAL RESULTS

In our experiments, all sequences were encoded using H.264/AVC with baseline profile. We modified the encoder so as to enable the selection of different motion estimation algorithms, which were split in two sets, respectively, $\mathcal{M}$ (to determine the eigenalgorithms $\mathcal{N}$ and the regions corresponding to each algorithm in $\mathcal{M}$) and $\mathcal{U} \setminus \mathcal{M}$, to validate the proposed method. Similarly, a set $\mathcal{S}_T$ of sequences (CIF format) were used in the training phase, while the set $\mathcal{S}_e$ was used for testing. Both algorithms and sequences are reported in Table 1. Sequences were coded with GOP structure IPPP of 15 frames and fixed QP values. The classification is performed based on the set of motion vectors extracted from each frame. Better results are expected by fusing the results obtained from several frames of the same sequence.

We selected a set of $n = 3$ eigenalgorithms. To be fair, we avoided to use the algorithms in $\mathcal{U} \setminus \mathcal{M}$ for training. To this end, due to the limited number of available algorithm, we adopted a sort of leave-one-out approach. That is, during training, we let $\mathcal{U}' = \{A, B, C, D, E, F\}$ and we built $\mathcal{M}'$ with one of the subsets with 5 elements from $\mathcal{U}'$, so as to have one algorithm in $\mathcal{U}' - \mathcal{M}'$, reiterating the process for each candidate subset $\mathcal{M}'$. Following this procedure, we identified the set $\mathcal{N} = \{A, B, E\}$.

Table 2 reports the average fraction of correct detections (CD) and false detections (FD) with different quantization parameters when using the proposed method, or the one described in Section 3 (i.e., [5]). Note that, when testing [5], we assume that all motion estimation algorithms in $\mathcal{U}$ are available to the analyst. We observe that our method achieves nearly the same results, but without explicit knowledge of the algorithms in $\mathcal{U} \setminus \mathcal{M}$ during training. For example, the percentage of correctly detected algorithms by the proposed method is as high as 96% for $QP = 20$, i.e., it is only marginally worse than the accuracy of [5], which is equal to 96.6%. It can also be observed that the accuracy decreases gracefully at higher values of QP, while remaining always above 87%.

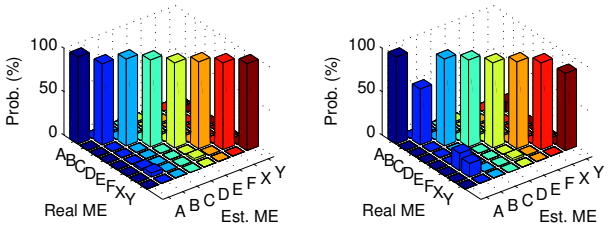


Fig. 4. Ambiguity matrices for different ME strategies and QP values. a) QP=20 b) QP=28.

Table 2. Performance of the proposed method vs. [5] .

		Eigenalgo.		**Idempot. [5]**	
Setting	QP	CD (%)	FD (%)	CD (%)	FD (%)
$\mathcal{M}$	20	98.89	1.05	99.14	0.81
	32	88.33	11.50	87.68	8.01
$\mathcal{U} \setminus \mathcal{M}$	20	96.08	3.97	96.92	2.81
	24	95.64	4.92	95.64	3.54
	28	91.90	8.19	91.60	5.58
	32	88.56	10.70	87.23	7.20

Figure 4 illustrates the performance of the proposed method more in detail, by means of confusion matrices for $QP = 20$ and $QP = 28$. We observe that the algorithms in $\mathcal{M}$ are correctly identified most of the times. In some cases, algorithm A and F are classified as algorithm B. Note that G and H were not known to the analyst. However, the proposed method correctly identifies them as not belonging to the set $\mathcal{M}$ and, at the same time, discriminates between them. We indicated X and Y, instead of G and H on the *Est. ME* axis since labelled data was not available at training time.

6. CONCLUSIONS

The paper presents a method to identify the motion estimation algorithm used by a video encoder. This is achieved by comparing the motion vectors extracted from the bitstream, with those that can be computed based on a small set of representative motion estimation algorithms, called eigenalgorithms. The method can be adopted to identify algorithms for which the software implementation is not available, e.g., because it is proprietary. The method might be used a preliminary analysis tool whenever there is a suspect of a possible patent violation. Future works will be devoted to extend the proposed approach to other processing units and investigate how an adversary might fool the method.

7. 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, no. 1, 2012, Available at http://journals.cambridge.org/abstract_S2048770312000029.

[2] Ziyuan Zhu and Tao Lin, "Idempotent h.264 intraframe multi-generation coding," in *Proc. of ICASSP 2009*, april 2009, pp. 1033 –1036.

[3] G. Valenzise, V. Nobile, M. Tagliasacchi, and S. Tubaro, "Countering jpeg anti-forensics," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, sept. 2011, pp. 1949 –1952.

[4] P. Bestagini, A. Allam, S. Milani, M. Tagliasacchi, and S. Tubaro, "Video codec identification," in *Proc. of ICASSP 2012*, Kyoto, Japan, Mar. 25 – 30, 2012.

[5] M. Sorell, "Video provenance by motion vector analysis: A feasibility study," in *Proc. of ISCCSP 2012*, Rome, Italy, May 2 – 4, 2012, pp. 35–42.

[6] Shan Zhu and Kai-Kuang Ma, "A New Diamond Search Algorithm for Fast Block-MatchingMotion Estimation," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287–290, Feb. 2000.

[7] Lai-Man Po and Wing-Chung Ma, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313–317, June 1996.

[8] Alexis M. Tourapis, Oscar C. Au, and Ming L. Liou, "Fast motion estimation using circular zonal search," in *Proc. of SPIE VCIP99*, 1999, pp. 1496–1504.

[9] Paolo De Pascalis, Luca Pezzoni, Gian Antonio Mian, and Daniele Bagni, "Fast motion estimation with size-based predictors selection hexagonal search in h.264/avc encoding," in *Proc. of EUSIPCO 2004*, Sept. 6 – 10, 2004.

[10] 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.