Countering Anti-Forensics by Means of Data Fusion

Marco Fontani[a], Alessandro Bonchi[b], Alessandro Piva[b] and Mauro Barni[c]

[a]CNIT, University of Siena, 53100 Siena, Italy;
[b]University of Florence, 50139 Firenze, Italy;
[c]University of Siena, 53100 Siena, Italy;

ABSTRACT

In the last years many image forensic (IF) algorithms have been proposed to reveal traces of processing or tampering. On the other hand, Anti-Forensic (AF) tools have also been developed to help the forger in removing editing footprints. Inspired by the fact that it is much harder to commit a perfect crime when the forensic analyst uses a multi-clue investigation strategy, we analyse the possibility offered by the adoption of a data fusion framework in a Counter-Anti-Forensic (CAF) scenario. We do so by adopting a theoretical framework, based on Dempster-Shafer Theory of Evidence, to synergically merge information provided by IF tools and CAF tools, whose goal is to reveal traces introduced by anti-forensic algorithms. The proposed system accounts for the non-trivial relationships between IF and CAF techniques; for example, in some cases the outputs from the former are expected to contradict the output from the latter. We evaluate the proposed method within a representative forensic task, that is splicing detection in JPEG images, with the forger trying to conceal traces using two different counter-forensic methods. Results show that decision fusion strongly limits the effectiveness of AF methods.

Keywords: Counter Forensics, Image Forensics, Data Fusion, Multi-clue decision

1. INTRODUCTION

In the last years Image Forensics (IF) has emerged as a way to investigate the processing history of digital images in a blind fashion.[1] Usually, IF tools work by searching a specific footprint that was left by processing operations as a side effect. Therefore, each IF tool usually behaves well in detecting whether a *certain* processing was applied. In real applications, however, there are many possible ways to tamper with an image, and the analyst does not know which forensic traces should be searched. A thorough analysis must rely on the use of many different tools, and merge their outputs as intelligently as possible. This path has been recently investigated in the IF field, and several methods are now available for data fusion.[2–4] As a new challenge to IF, anti-forensic (AF) methods are emerging, whose goal is to remove the footprints left during processing, making forensic analysis harder.[5] On the other hand, AF tools may leave their own footprints, and counter-anti-forensic (CAF) tools are being designed to detect them as well.

In such a scenario, the forensic analyst needs to simultaneously tackle with both the variety of detectable footprints and the presence of an adversary equipped with AF tools. If CAF tools are available to the analyst, a more robust analysis can be carried out, provided that outputs from IF tools and their CAF versions are interpreted properly, thus going back to information fusion. To the best of our knowledge, so far information fusion and CAF have been investigated only separately.

In this paper we analyse the possibility offered by the adoption of a data fusion framework in a CAF scenario. An intuitive solution could be to use all the available tools in an additive fashion ("OR" fusion rule), that means classifying the image as tampered when either an IF or a CAF tool detects the footprint it is looking for. This approach, however, does not take into account the following facts: i) IF tools may be searching for mutually exclusive traces,[2] so some combinations of tool outputs could be excluded; ii) for a given footprint, IF and

Further author information: (Send correspondence to Marco Fontani)
Marco Fontani: E-mail: marco.fontani@gmail.com
Alessandro Piva: E-mail: alessandro.piva@unifi.it
Mauro Barni: Email: barni@dii.unisi.it

CAF algorithms are expected to be in contradiction, so if both kinds of tools detect their footprint this should at least raise some doubts about the correctness of the outputs; and iii) detecting some kinds of anti-forensic processing does not necessarily imply that the image is a fake (e.g., full-frame linear filtering may be seen as an AF technique, but usually it has to be considered a common, innocent operation). To properly account for these facts and dig into the full potentialities offered by data fusion for CAF, we adopt a recently introduced[2] theoretical framework, based on Dempster-Shafer Theory of Evidence. Such framework allows to fuse the scalar output coming from the tools, explicitly modeling the logical relationships (e.g., compatibility) between traces searched by different tools. The system we propose synergistically merges information coming from IF and CAF tools, leading to a significant reliability improvement with respect to the use of single tools. In the scope of this paper, we focus on the task of image splicing detection: given an image and a suspect region, the analyst wants to understand whether the region has been pasted from another picture or not. This choice is motivated by the wide availability of IF tools targeting this task.

The rest of the paper is organized as follows: we briefly introduce Dempster-Shafer Theory in Section 2, then we discuss about integration of CAF and IF tools in Section 3. Two case studies are defined and experimentally investigated in Section 4, and finally Section 5 concludes the paper.

2. BASICS OF DEMPSTER-SHAFER THEORY

Dempster-Shafer Theory of Evidence (DST)[6,7] is a widely employed mathematical framework allowing to make reasoning and inference in contexts where uncertainty and lack of information are strong. Indeed, one of the most attractive features of DST is the capability of modeling uncertainty and doubt in an explicit and very simple way, especially compared to classical Bayesian probability theory. In the following, we introduce the fundamental definitions and tools of this theory, for a deeper description we refer to the original book.[7] To begin with, let us call $\Theta = \{\theta_1, \theta_2, \dots \theta_n\}$ the set of possible conclusions that we can reach about some elements of interest, where elements of Θ are mutually exclusive and exhaustive. Properly choosing the set Θ is very important in DST: it must represent the desired granularity of information that we want to (or we can) reach, so that choosing $\Theta = \{\text{car}, \text{truck}, \text{bus}, \text{scooter}, \text{motorbike}\}$ may be less appropriate than choosing $\Theta = \{\text{four-wheeled}, \text{two-wheeled}\}$ for some tasks.

Instead of assigning probabilities to the element of Θ, DST focuses the attention on the *power set* of Θ, which is called "frame of discernment". This means that we can assign a degree of support, called *belief*, to each possible subset of Θ, which obviously includes single elements of Θ as a sub-case. Such a degree of support takes values within 0 and 1, where smaller values indicate a weaker support on the set. Before making an example, we formalize the concept of support.

DEFINITION 2.1. *Let Θ be a frame. A function $m^\Theta : 2^\Theta \rightarrow [0,1]$ is called a Basic Belief Assignment (BBA) over the frame Θ if:*

$$m^\Theta(\emptyset) = 0; \qquad \sum_{A \in 2^\Theta} m^\Theta(A) = 1 \tag{1}$$

where the summation is taken over every possible subset A of Θ.

As a classical example, imagine that a physician sees a patient and is asked to assign a support on the frame defined by the set $\mathcal{D} = \{\text{cancer}, \text{pneumonia}, \text{cold}\}$. He may return the following BBA:

$$m^\Theta(X) = \begin{cases} 0.7 & \text{for } X = \{\text{cold}\} \\ 0.2 & \text{for } X = \{\text{pneumonia}\} \\ 0.1 & \text{for } X = \{\text{cancer} \cup \text{pneumonia}\} \end{cases}. \tag{2}$$

Such an assignment tells that the physician is mostly convinced that the patient is affected by a simple cold (first line), but he also saw some symptoms that are compatible with pneumonia (second line) and, moreover, some symptoms that are compatible both with pneumonia and cancer (third line). Thanks to the fact that support is assigned to subsets of Θ, we can easily model the difference between symptoms that are only compatible with pneumonia and those that do not allow to discriminate between pneumonia and cancer. It must be stressed that the mass assigned to the set $\{\text{cancer} \cup \text{pneumonia}\}$ does not "propagate" to the subset $\{\text{pneumonia}\}$ due

to the lack of information allowing to discriminate between cancer and pneumonia: the mass assigned to a set represents the support committed to exactly that set. On the other hand, we may want to evaluate, based on the available knowledge, the total support for the set $\{\text{cancer} \cup \text{pneumonia}\}$. Intuitively, this should also include the mass of every subset, that means adding up the masses on the second and third line of (2). This is the concept of *belief* for a set, that is defined as follows:

DEFINITION 2.2. *Given the BBA in 2.1, the Belief function* $Bel : 2^{\Theta} \to [0, 1]$ *is defined as follows:*

$$Bel(A) = \sum_{B|B \subseteq A} m(B). \tag{3}$$

$Bel(A)$ summarizes all our reasons to believe in A with the available knowledge. In our example, therefore, we have $Bel(\{\text{cancer} \cup \text{pneumonia}\}) = 0.3$.

Here we do not discuss the interesting properties of these functions nor their relationships, that are well treated in the original work.[7] We only notice that specifying one between $m(\cdot)$ and $Bel(\cdot)$ is enough to allow deriving the other.

2.1 Combining sources of evidence

One of the most attracting features of DST is the rule for combining several sources of evidence. Given two BBAs defined over the same frame of discernment, and obtained by independent sources of evidence, Dempster's rule allows to merge them into a single belief assignment.

DEFINITION 2.3. *Let* Bel_1 *and* Bel_2 *be belief functions over the same frame* Θ *with BBAs* m_1 *and* m_2*. Let us also assume that* K*, defined below, is positive. Then for all non-empty* $X \subseteq \Theta$ *the function* m_{12} *defined as:*

$$m_{12}(X) = \frac{1}{1-K} \cdot \sum_{\substack{A,B \subseteq \Theta: \\ A \cap B = X}} m_1(A) m_2(B) \tag{4}$$

where $K = \sum_{A,B:A \cap B = \emptyset} m_1(A) m_2(B)$*, is a BBA function defined over* Θ *and is called the orthogonal sum of* Bel_1 *and* Bel_2*, denoted by* $Bel_1 \oplus Bel_2$*.*

Dempster's rule is commutative and associative, but it is not idempotent, meaning that observing twice the same evidence will change the belief assignments. This fact motivates the assumption of independence of the sources of evidence: we must ensure we are not counting twice the same fact. Unfortunately, there is not a clear and formal definition of independence in DST, also due to the heterogeneity of possible sources of evidence. We may think that using Dempster's rule is safe when information is gathered by different means, or listening to experts who did not influence each other. The meaning of the denominator of (4) also deserves attention: K is obtained by pooling the incompatible evidence provided by the two sources (the summation is taken over sets whose intersection is empty), so we can consider it as a measure of the conflict between the two BBAs. The effect of dividing the pooled evidence by $1 - K$ is to remove the conflicting evidence from the fused belief assignment, and redistribute that part of evidence on the non-conflicting assignments.

As a final remark about Dempster's rule, we focus on the other assumption in Definition 2.3, that requires sources of evidence being defined over the same frame of discernment. Intuitively this means that only sources telling about the same facts can be merged, and this is a rather indisputable requirement. However, DST allows to "rewrite" a BBA on a different frame of discernment by either extending or marginalizing the set over which the frame is defined (the formal definition of these operations can be found in the work by Shafer[7]). Therefore, when BBAs defined over different frames have to be combined, they can be extended to the same reference frame and then properly merged together.

3. MULTI-CLUE ANALYSIS FOR COUNTERING ANTI-FORENSICS

In this section we discuss the possibility of modelling the relationships between IF and CAF tools using DST. In order to do that, we first introduce a recently proposed model for decision fusion in image forensics that allows to merge the outputs from several IF tools.[2,8] Then, we turn to discuss about possible ways to integrate information from CAF tools within such a framework.

3.1 Adopted Framework

A DST-based framework for combining the evidence stemming from several tools has been proposed.[2] The framework performs fusion at the so-called "score-level", meaning that it combines the scalar output produced by each IF tool, without considering its internal features. Being based on DST, the framework can be used to merge together tools that are based on different analysis algorithms, due to the independence assumption behind Dempster's combination rule. One of the most important features of the framework is that it allows to explicitly write (when they are available) the compatibility relationships between different image forensic traces: this part is crucial to our goal of integrating IF and CAF tools, and it will be described later in more detail. The basic idea underlying the framework is to model each IF tool as a source of evidence about the presence or absence of a specific forensic trace within the analysed image or region. This is done by defining a different set for each trace containing two elements: "the trace is present" and "the trace is not present", and by mapping the tool output to a BBA assignment over the frame of discernment generated by that set. For example, if we have a tool A searching for a trace called α, the frame of discernment will be the power set of $\Theta_\alpha = \{t\alpha, n\alpha\}$, where $t\alpha$ means that the trace is present and $n\alpha$ means the opposite. Information provided by the tool is then modeled with the following belief assignment:

$$m_A^{\Theta_\alpha}(X) = \begin{cases} A_T & \text{for } X = \{(t\alpha)\} \\ A_N & \text{for } X = \{(n\alpha)\} \\ A_{TN} & \text{for } X = \{(t\alpha) \cup (n\alpha)\} \end{cases}, \tag{5}$$

where A_T, A_N and A_{TN} are functions mapping the scalar output of the tool to a mass value. In the original version of the framework, these functions were statically defined by the user;[2] recently it has been shown that they can be learned automatically so to account also for auxiliary information, like characteristics of the analysed image of region, that may affect the behavior of the tool.[8] In this work we choose the latter approach, which is preferable because it adapts the interpretation of tool output by the light of the general characteristics of the content; details about the chosen characteristics are given in Section 4, because they depend on the specific IF tools that are considered.

Equation (5) shows how to model the information stemming from one tool. When the information gathered by another tool B must also be considered by the system two cases are possible: tool B may either be searching for the same trace α or for a different trace, say β. In the former case, a BBA $m_B^{\Theta_\alpha}$ like the one in (5) is obtained, while in the latter case a BBA $m_B^{\Theta_\beta}$ enters the system, which has the same meaning described above but concerning the trace β. These two cases must be addressed differently when we turn to the next step of the framework, which consists in merging the information obtained by various tools. As long as the same trace is concerned, Dempster's rule (4) can be used directly to summarize the information provided by different tools about that trace. On the other hand, if two tools looking for different traces α and β are to be fused, their BBA must be first extended on a common frame, that is the power set of $\Theta_\alpha \times \Theta_\beta$, and then merged together. Details about this procedure are provided in the original work,[2] and will not be exposed here.

3.1.1 Information about relationships between forensic traces

It is important to notice that, up to this point, information has been combined together without introducing knowledge about traces relationships. Since image forensic footprints are usually well defined and based on a specific phenomenon, the forensic analyst should be able, in most cases, to determine which combination of traces can be present at the same time and which can not. Going back to the previous example, it may be that traces α and β cannot be present at the same time, due to the way they are defined. The framework allows the analyst to include such information by writing a BBA like the following one:

$$m_{comp}(X) = \begin{cases} 1 & \text{for } X = \{(t\alpha, n\beta) \cup (n\alpha, t\beta) \cup (n\alpha, n\beta)\} \\ 0 & \text{for } X = \{(t\alpha, t\beta)\} \end{cases}, \tag{6}$$

where we see that a null mass is given to the event $\{(t\alpha, t\beta)\}$, meaning that that event is deemed to be impossible. This BBA can be fused with the one obtained by fusing tools searching for traces α and β, and Dempster's rule will, by definition, remove and redistribute masses supporting the not-plausible event.

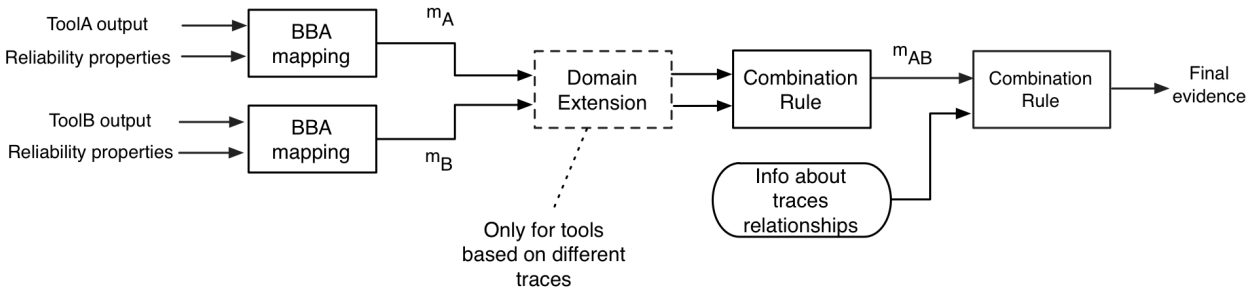


Figure 1.

After introducing compatibility relationships, the framework comes to an end: the produced BBA resumes the knowledge brought by tools and by forensic analyst's expertise. Once this point is reached, the analyst may want to evaluate the belief of a certain set, e.g. the set containing all the events where at least one trace of forgery has been detected.[2] Evaluating the belief from a given BBA is trivial using equation (3). Summarizing the system works according to the following flow (see Figure 1):

1. the scalar output from each tool is converted to a BBA, possibly taking into account parameters that may influence the reliability of the tool;[8]
2. contribution from tools are merged together in one of two possible ways:
   - the BBAs assigned by tools looking for different traces are adapted to a common domain and fused;
   - the BBAs resulting from tools searching for the same forensic trace are directly fused using Dempster's rule;
3. compatibility relationships between different traces (modeled through a BBA) are introduced using Dempster's rule.

3.2 Integrating IF and CAF methods

We now investigate how CAF tools can be integrated in the framework described in the previous section, so that the fore sic analysis remains reliable even in the presence of an adversary employing AF tools. The main idea behind CAF is to search for the traces that are left by AF tools; indeed, most existing AF tools only attempt to remove some kind of forensic trace (e.g., comb-shaped histograms in the pixel or DCT domain), but they do not care about making the statistics of the produced signal close to that of an untouched content. Only very recently the first steps have been taken towards universal counter-forensic techniques, but they are limited to the specific scenario where analyst's tools are based on first-order statistics.[9] As a result, we can say that the application of common AF tools may introduce new footprints that are possibly harder to detect, but still detectable.

In such a scenario, the forensic analyst should consider the event that the analysed content has been manipulated and then "cleaned" by the forger. This means that, whenever they are available, CAF tools should also be part of the set of tools employed during the analysis. If we go back to the fusion framework described in Section 3.1, CAF tools can be modeled as standard IF tools: they simply search for a specific trace and provide some information about its presence or absence. However, the complementary nature of these tools compared to IF tools raises some questions about how they should be integrated. We can distinguish between two possible approaches:

- a *cascade scheme*, where outputs from IF tools are first merged using the system in Fig. 1, and CAF tools are considered later, possibly introducing knowledge about relationship between CAF and IF traces;
- a *mixed scheme*, where IF and CAF tools are treated at the same level.

The cascade scheme is more convenient from a complexity point of view: after fusing information about IF traces, the analyst may marginalize such information to reduce the cardinality of sets. For example, the analyst may summarize the information about all traces of a specific family (e.g., JPEG-related traces) into a single variable, discriminating between presence or absence of that family of traces. Then, the compacted belief structure could be merged with the information coming from CAF tools. The downside of the cascaded structure is the possible over-simplification induced by grouping traces into families. Indeed, relationships between traces searched by IF and CAF tools are not trivial: in the simplest case the adversary, by applying an AF tool, erases the trace searched by a IF tool while introducing the trace searched by a CAF tool, so that these two traces are simply mutually exclusive. However, as we will see in next sections, it may be that due to the application of AF tools the relationships between the standard IF traces are changed: traces that could not coexist in a "standard" forgery scenario, may become compatible due to the application of AF tools and viceversa. This can happen even for traces within the same family, so that the marginalization would cause an over-simplification of the problem, in that it would impede to correctly update relationships in presence of CAF traces. Finally, we can state that it is safer to treat IF and CAF traces at the same level, and the higher complexity seems an unavoidable price to be paid. Moreover, cheaper solutions are not easy to find: for example, the above facts have a heavy impact also on fusion frameworks based on machine-learning. The increasing number of possible combinations of IF and CAF traces makes it hard to devise proper training sets without generating and analysing an exponential number of training samples. This is a serious problem since it has been shown[2] that neglecting some combinations of traces during training could severely impact the performance of the framework.

It is worthy of remarking that the application of an AF tool targeted to remove a particular forensic trace may affect also the detectability of other traces; surprisingly, the effect could be to *increase* the detectability of other traces. Even more, application of a targeted AF tool may introduce an IF trace that was not present beforehand. This means that, by using a multi-clue analysis, the analyst may be able to counter the effect of AF tools without employing CAF tools, and simply relying on the complementary capabilities of the IF tools included in the framework (we will see an example in Section 4.2).

3.3 Managing configuration of tools

When combining the capabilities of multiple tools, there is one practical yet interesting fact that must be accounted for: it may happen that, for a given image, only part of the tools within the framework can be run, while others are not compatible with the image (e.g., due to image format, size, number of channels and so on). Is it possible for the analyst to handle these variants without increasing the complexity and extensiveness of the framework? DST offers a nice way to solve this issue. Given a set Θ, the following BBA, known as *vacuous* BBA, is the neutral element for Dempster's combination rule:

$$m_V^{\Theta}(X) = \begin{cases} 0 & \forall\, X \subset \Theta \\ 1 & \text{for } X = \Theta \end{cases}. \tag{7}$$

We see that $m_V^{\Theta}(X)$ is a valid BBA assigning all of the mass to the whole frame of discernment, which means that no knowledge is brought about elements of Θ. For any BBA m_X^{Θ}, we have:

$$m_X^{\Theta} \oplus m_V^{\Theta} = m_X^{\Theta}, \tag{8}$$

meaning that m_V^{Θ} can be fused an arbitrary number of times without modifying the available information.

Let us go back to our problem, and assume that there is a tool called U, searching for trace α, that cannot be executed on the image under analysis. The analyst will simply write the following:

$$m_{U^{\alpha}}^{\Theta}(X) = \begin{cases} 0 & \text{for } X = \quad \{(t\alpha)\} \\ 0 & \text{for } X = \quad \{(n\alpha)\} \\ 1 & \text{for } X = \{(t\alpha) \cup (n\alpha)\} \end{cases}, \tag{9}$$

and use the framework without changes. Tool U will not contribute at all to the final belief about presence of trace α. In the extreme case where none of the tools available to the analyst can handle the image, we still

obtain a valid BBA, telling that no information is available about searched traces. Also in this case, we point out that the above feature is interesting compared to some machine-learning approaches that would require a different training for every possible combination of tools.

3.4 Interpretation of AF traces

A controversial point about searching traces of anti-forensics is that some processing operators have a double valence: they can both be used "benignly" to enhance the quality of the image, or they can be used "maliciously" as a tool for erasing traces of previous processing. For instance, Kirchner et al. showed how median filtering can be used to erase traces of resampling,[10] that are often used to localize zoomed or rotated areas within a tampered digital image. Due to this fact, the forensic analyst should try to detect application of processing operators that affect traces searched by IF tools. On the other hand, it could seem hasty to classify an image as manipulated when only traces of possibly benign processing operators are detected: although benign processing can be considered a form of manipulation, we believe it is of interest for the analyst to discriminate between global processing, like filtering, and forging operations like splicing. A solution to solve this apparent deadlock is to deepen the analysis of AF traces moving, when it is possible, from a global perspective (i.e., search for the trace over the whole image) to a local perspective (search the trace separately on the suspect region and on the rest of the image). Indeed, while revealing a global application of a processing operator may be considered acceptable, detecting *inconsistent* presence of traces among different regions is much more alarming. Finally, what we are facing with is an interpretation problem: should the presence of traces characterizing "benign" processing operators raise integrity warnings? Should they do that only when they are present in an inconsistent way across the same image? We believe the answer to these questions depends on the field of application, and should be left to the analyst. Therefore, the fusion framework should not force any interpretation, while enabling the possibility for the analyst to select the preferred one.

The above arguments can be included in the adopted fusion framework in a rather intuitive way. When introducing information about the presence of ambivalent processing operators, we introduce *two* traces within the framework: one concerning the presence of operator traces within the suspect region, and one concerning the presence of operator traces within the rest of the image. For example, suppose we have a tool G searching for a CAF trace γ: we propose to use tool G to assign BBAs to the frame of discernment associated to the set $\Theta_\gamma^I = \{t\gamma^I, n\gamma^I\}$ and $\Theta_\gamma^O = \{t\gamma^O, n\gamma^O\}$, where $t\gamma^I$ and $n\gamma^I$ denote, respectively, presence or absence of the trace inside the analysed region, and $t\gamma^O$ and $n\gamma^O$ denote presence or absence of the trace outside the analysed region. If we consider the product set $\Theta_\gamma^I \times \Theta_\gamma^O$, we obtain all possible combinations of presence and absence of the CAF trace inside and outside the analysed region. In this way, the analyst can choose, for example, whether presence of the trace both inside and outside the region, that is the event $(t\gamma^I, t\gamma^O)$, should be considered as an integrity violation or not.

4. TWO CASE STUDIES

In this Section, we consider two case studies whose goal is to evaluate the practical impact of CAF tools when the adversary is equipped with AF technologies. We focus on a widely studied task in image forensics, that is splicing detection: given a digital image, splicing detection aims at understanding whether a suspect region (either specified by the analyst or automatically selected in some way) has been pasted from another image. Interestingly, there are many possible ways to perform a cut-&-paste attack: for example, the forger may start from two JPEG-compressed images, or from one JPEG-compressed and one uncompressed image. The difference between these settings is probably negligible to the attacker's eyes, yet completely different traces are left within the media. This is the main reason enforcing the use of multi-clue analysis: complementary tools are needed to cover a wider range of possible settings.

In devising the two case studies, we start from the original experimental settings adopted by Fontani et al.[2] The analyst employs five different tools for splicing detection based on the analysis of three different JPEG-related traces:

- the tool by Bianchi et al.[11] and the tool by Luo et al.[12] searching for traces of not-aligned double JPEG compression (JPNA);

- the tool by Bianchi et al.[13] and the tool by Lin et al.[14] searching for traces of aligned double JPEG compression (this trace will be called JPDQ from now on);
- the tool by Farid[15] searching for traces of the so-called "JPEG-ghost" (JPGH).

The compatibility relationship between these traces is reported in Table 1.[2]

Comb. num	**JPNA**	**JPDQ**	**JPGH**	**Interpr.**
1	0	0	0	Non-tampered
2	0	0	1	Tampered
3	0	1	0	-
4	0	1	1	Tampered
5	1	0	0	Tampered
6	1	0	1	-
7	1	1	0	-
8	1	1	1	Tampered

Table 1. Trace relationship table: each row forms a combination of presence (1) and absence (0) of traces. In the rightmost column we see the interpretation of each combination, where impossible combinations are denoted by a dash. Notice that only 5 out of 8 combinations are possible.

Switching to the adversary's point of view, regardless of counter-forensic strategies, four different cut-&-paste procedures are considered to create a splicing starting from two images (at least one of which is in JPEG format), that are described in Table 2. As the reader can see from the table, different procedures introduce different combinations of IF traces.

Class	**Procedure**	**Traces in inner region**	**Traces in outer region**
Class 1	Region is cut from a JPEG image and pasted, breaking the 8x8 grid, into an uncompressed one; the result is saved as JPEG.	JPNA	-
Class 2	Region is taken from an uncompressed image and pasted into a JPEG one; the result is saved as JPEG.	-	JPDQ JPGH
Class 3	Region is cut from a JPEG image and pasted into an uncompressed one in a position multiple of the 8x8 grid; result is saved as JPEG.	JPGH	-
Class 4	Region is cut from a JPEG image and pasted (without respecting the original 8x8 grid) into a JPEG image; the result is saved as JPEG	JPNA	JPDQ JPGH

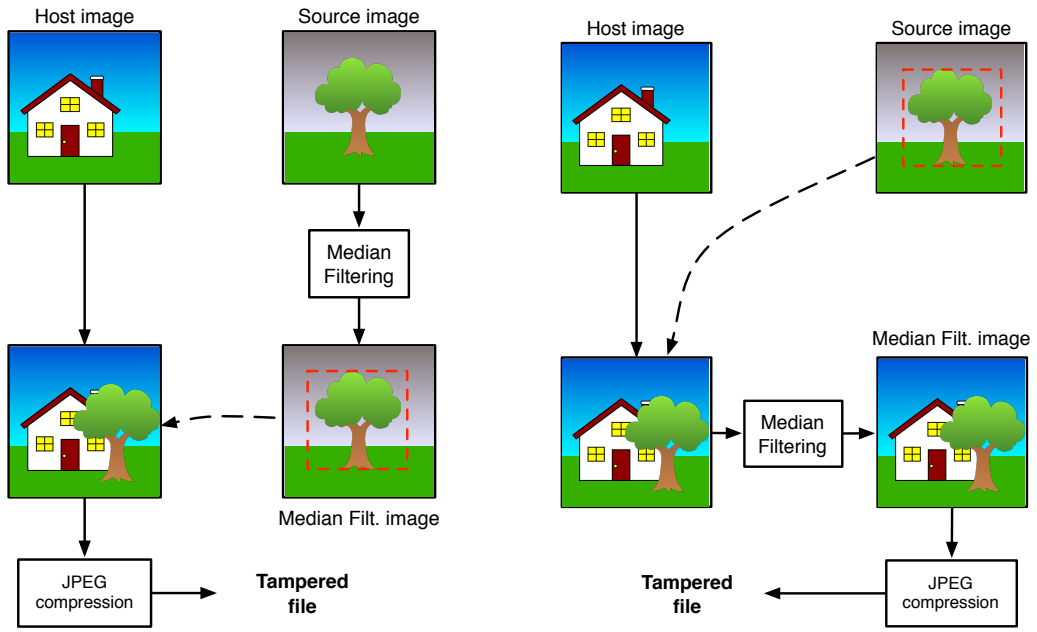
Table 2. Procedure for the creation of different classes of tampering in the training dataset.

Starting from this background, we upgrade both the forger's skills by introducing counter-forensic methods, and the analyst's skills by providing proper CAF tools. The two considered case studies differ in that, in the first one, the forger wants to obtain a spliced JPEG image where traces of double encoding are concealed, while in the second (more complex) case the product of the splicing must show no traces of compression at all (thus erasing all traces that may be used by JPEG-based IF tools).

4.1 Splicing detection in the presence of double encoding concealment

In this case study we consider a forger who wants to generate a splicing starting from two images, at least one of which is in JPEG format, and finally encode the result as a JPEG image. Since it is known that many IF tools exist based on the analysis of double JPEG compression traces, the adversary wants to conceal these traces. A simple but effective way to accomplish such a task is to perform a median filtering between the two compressions, thus avoiding pixels from showing traces of double quantization. Although other kinds of filtering operators could also be used, we believe median filtering is preferable because of its non-linear nature, that complicates the task of the analyst.

The adversary can apply the filter either before of after the cut-&-paste operation, resulting in two different scenarios (Figure 2). Applying MF to the whole image (right figure) gives more chance of disrupting traces but also increases the probability of being detected by CAF algorithms. Since median filtering is known to be a good



(a) (b)

Figure 2. Two possible methods available to the analyst to conceal forensic traces that are left during tampering using median filtering.

AF method,[16] we can reasonably think that the analyst adopts proper counter-measures. Therefore, we include the tool for detecting MF in JPEG compressed images proposed by Kirchner et al.[16] within the analyst's pool of tools. This CAF tool uses SPAM (subtractive pixel adjacency matrix) features[17] to characterize the relationships between neighbouring pixels, and classify the analysed region as median filtered or not using machine learning techniques. We trained the classifier following indications in the original work.[16]

After choosing the set of IF and CAF traces to search for, the analyst needs to reason about the relationships between these traces. First of all, this case study falls within the category described in Section 3.4: median filtering is a "forensically ambiguous" operator because it can be used both benignly to remove noise from the image and maliciously, as an AF tool. Based on the discussion in the aforementioned Section, we find it appropriate to use the CAF tool to analyse *separately* the suspect region and the rest of the image, so to distinguish between a malicious and innocent use of some processing.

Comb. num	JPNA	JPDQ	JPGH	CAF-IN	CAF-OUT	Interpr.
1	0	0	0	0	0	Non-Tampered
2	0	0	0	0	1	Tampered
3	0	0	0	1	0	Tampered
4	0	0	0	1	1	Non-Tampered
5	0	0	1	0	0	Tampered
6	0	0	1	1	0	Tampered
7	0	0	1	1	1	Tampered
8	0	1	1	0	0	Tampered
9	0	1	1	1	0	Tampered
10	0	1	1	1	1	Tampered
11	1	0	0	0	0	Tampered
12	1	0	0	1	1	Tampered
13	1	0	1	0	1	Tampered
14	1	1	1	0	0	Tampered
15	1	1	1	1	1	Tampered

Table 3. Compatibility relationships for IF and CAF traces considered in the first case study. Each row considers a combination of presence (1s) or absence (0s) of considered traces. Notice that the presence of the CAF trace is treated separately for the suspect region (CAF-IN column) and the rest of the image (CAF-OUT). Only plausible combinations are reported in the table, those combinations that are not listed are theoretically incompatible.

We can finally write relationships, updating Table 1 so to account for the presence of traces of median filtering (both inside and outside the analysed region): the result is plotted in Table 3. Notice that, for brevity, only plausible combinations are reported. First of all, notice that combination number 4 reads as follows: when traces of median filtering are detected throughout the whole image, and this is the *only* detected trace, then the image is considered intact. This is the chosen interpretation in this case study; however, nothing prevents the analyst from changing this interpretation, for example to account for a specific setting where any retouch of the image is not acceptable. Moving to the rest of the table, most of the entries are rather intuitive: integrity violation is detected when at least one of the IF tools finds the trace it is looking for, and also when traces of MF are present in only one part of the image (because inconsistent use of filtering is interpreted as a malicious behaviour). The only combination leading to interpret the image as "intact" is the one where none of the traces are present. On the other hand, some combinations exist that raise interest. Let us consider the combination number 13: if we focus only on IF traces (columns 2-4), we recognize one of the impossible combinations according to Table 1, namely number 6. Yet, if we take into account CAF traces (columns 5-6), the combination becomes possible: indeed, median filtering applied to the external region makes it impossible to detect traces of double quantization (JPDQ) there, while the other two traces are still detectable because the inner region was not affected by filtering. This is a very good example supporting the mixed architecture despite its heavier cost compared to the cascaded scheme.

4.1.1 Experimental Results

Given the set of tools defined in this case study, and the compatibility relationships in Table 3, we want to investigate the performance of the multi-clue framework. To do that, we started from 100 uncompressed and heterogeneous TIFF images (indoor, outdoor, landscapes, etc.) of size 1024×1024 pixels, and we generated a dataset of 8000 images, of which:

- 2000 untouched images, obtained by simply applying JPEG compression;
- 500×4 spliced images without AF, generated using the 4 different cut&paste attacks reported in Table 2;
- 500×4 spliced images, to which AF is applied to the spliced region only, according to Figure 2(a).
- 500×4 spliced images, to which AF is applied to the whole image, according to Figure 2(b).

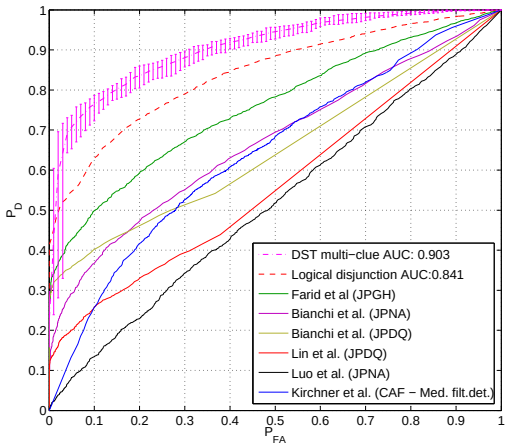


Figure 3. ROC curves obtained on the whole dataset by each of the considered IF and CAF tools (solid lines), and by the two decision fusion methods (dashed lines). The ROC curve relative to the DST-based method shows the maximum and minimum value obtained through all the train-test iterations.

The spliced region has always a size of 256×256 pixels, and is located in the center of the image. Possible values for the quantization quality of the first compression, denoted with Q_1, were chosen so that $Q_1 \in \{40, 45, ..., 80\}$, while the second quality factor Q_2 was defined as $Q_2 = Q_1 + \delta$, with δ randomly chosen from $\{+5, +10, +15, +20\}$.*

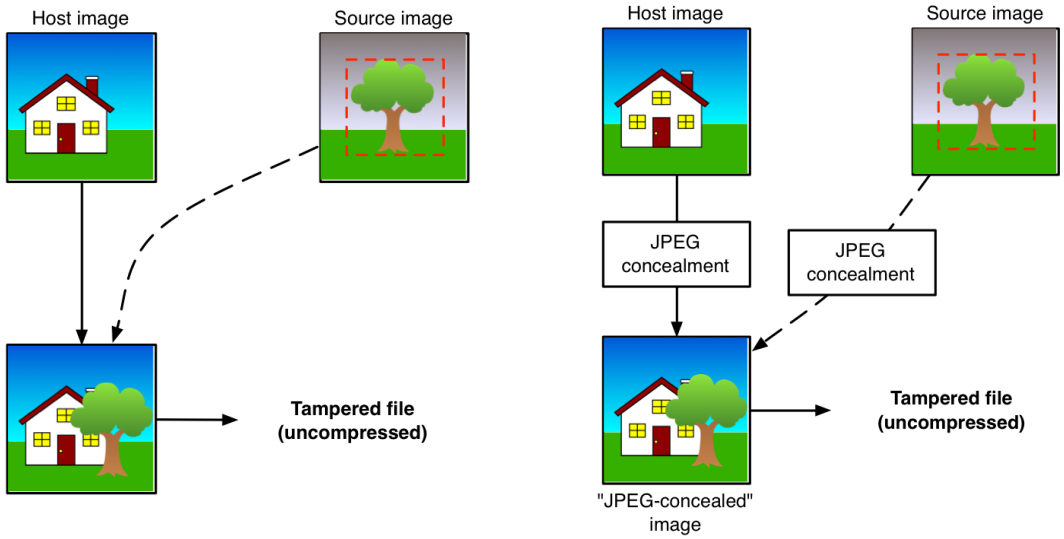
We run the 5 IF tools and the CAF tool for median filtering detection on all images, then we employed the proposed method to calculate merged mass assignments. We used the method proposed by Fontani et al.[8] to map tool outputs into mass assignments, including background information about the mean value, standard deviation and quality compression of the analysed region. Since this kind of BBA mapping requires training information, 80% of the images in the dataset (selected at random) were used to train[8] each tool separately, and the rest were used for testing the system. This procedure was repeated 10 times to increase the statistical significance of the results.

For each image, we evaluated the belief for the set containing all combinations of Table 3 whose interpretation is "tampered". Then, the obtained belief was thresholded so to obtain the final decision. Figure 3 shows the Receiver Operating Characteristic (ROC) curves obtained with: the proposed method (dash-dot curve), with every single IF and CAF algorithm alone (solid curves), and also with a simple decision fusion rule (dashed curve), that is logical disjunction (i.e., the image is classified as tampered if at least one algorithm detects a trace). The most evident fact is that decision fusion strongly helps the analyst in the presence of an adversary: this confirms that AF methods are less effective when the analyst uses a pool of complementary IF tools. We also see that the proposed system outperforms the logical-disjunction method, a rather trivial yet widely used approach.

4.2 Splicing detection in the presence of JPEG coding concealment

In this scenario, the forger starts with two images (at least one JPEG coded) and wants to produce a splicing that is not JPEG compressed. In order to accomplish this task, the forger can either adopt a naive approach (Figure 4(a)) where the spliced image is just stored in an uncompressed format, or a smarter approach where an AF tool is applied to conceal traces of previous JPEG compression (Figure 4(b)) before creating the splicing. A good candidate for this AF task is the tool proposed by Stamm et al,[18] that removes the characteristic trace left by JPEG compression, namely gaps in the histograms of DCT coefficients. Gaps are filled by adding a dithering noise to coefficients so to make their distribution resemble that of an uncompressed image. Of course, the forger applies this AF method only to pixels coming from JPEG-coded images.

*We did not consider the case where the second compression is at a lower quality than the former (that is, $Q_1 < Q_2$) because it is known that JPEG-based tools do not perform well in such a setting.



(a) (b)

Figure 4. Two possible methods available to the analyst to produce an uncompressed spliced image. In figure (b), JPEG concealment is applied only to pixels coming from JPEG-compressed images.

Let us now consider the analyst's side. The IF tools considered in this paper are based on JPEG artefacts, and we may think that the analyst is stuck when an uncompressed image has to be analysed. Yet, the analyst may consider the following possibilities:

1. the image is actually intact;
2. a splicing has been generated starting from two uncompressed images;
3. a splicing has been generated starting from one (or both) JPEG-coded images, and it has been stored with no compression;
4. a splicing has been generated starting from one (or both) JPEG-coded images, traces of JPEG compression have been removed using an AF tool, finally the result has been stored with no compression;

With the available set of IF tools, the analyst can still handle the scenario in point 3 with a rather simple approach: by recompressing images, and searching for traces of aligned or not-aligned double compression. If such traces are found, and they are inconsistent between the suspect region and the rest of the image, then the splicing is properly exposed. However, a clever forger would probably have concealed traces of JPEG compression, like in Figure 4(b). This is where we upgrade the tools of the analyst, introducing the CAF tool,[19] proposed by Valenzise et al., that allows to detect JPEG compression even in presence of the previously mentioned AF attack. By using this CAF tool, the analyst can expose traces of previous JPEG-compression and, most interestingly, search for inconsistent traces inside and outside the suspect region.

We stress again that, in practice, the analyst does not know which processing chain the suspect image underwent. A good strategy, therefore, is to run the available IF and CAF algorithms and properly interpret their outputs. To this end, in this case study we let the analyst perform the following actions:

- when the image is in JPEG format: apply the IF tools, and disable the CAF tool (see Section 3.3);

Comb. num	JPNA	JPDQ	JPGH	CAF-IN	CAF-OUT	Interpr.
1	0	0	0	0	0	Intact
2	0	0	0	0	1	Tampered
3	0	0	0	1	0	Tampered
4	0	0	0	1	1	Tampered
5	0	0	1	0	0	Tampered
6	0	0	1	0	1	Tampered
7	0	0	1	1	0	Tampered
8	0	0	1	1	1	Tampered
9	0	1	1	0	0	Tampered
10	0	1	1	1	0	Tampered
11	1	0	0	0	0	Tampered
12	1	0	0	0	1	Tampered
13	1	0	1	0	1	Tampered
14	1	1	1	0	0	Tampered

Table 4. Compatibility relationships for IF and CAF traces considered in the second case study. Each row considers a combination of presence (1s) or absence (0s) of considered traces. Only plausible combinations are reported in the table, those combinations that are not listed are theoretically incompatible.

- when the image is uncompressed, do both the following:
  - use the CAF tool to expose possible traces of previous JPEG compressions;
  - perform a high-quality JPEG compression and run the IF tools.

In order to maximize the benefits from this joint analysis, the analyst must then provide knowledge about IF and CAF traces relationships, together with the interpretation. Table 4 shows a possible, reasonable, choice for the considered case study. Also in this case, the mixed scheme allows to properly account for non-trivial relationships. For example, let us consider line 13: the combination of IF traces would normally not be possible (see combination 6 in Table 1), but it becomes possible when JPEG traces were removed from outside of the suspect region, and not inside it. Considering Figure 4(b), this would actually happen when the host image was JPEG compressed and the source image was not. Finally, we point out that, in this case study, presence of AF traces is interpreted as a manipulation even when no IF traces are found (row 4 of Table 4): this is is motivated by the fact that hiding traces of JPEG compression can never be considered a "benign" processing, since it does not bring any positive effect to the image.

4.2.1 Experimental results

Based on the above description, we generated a dataset of intact and forged images so to evaluate the performance of IF tools and of the multi-clue analysis system. We followed the same approach described in Section 4.1.1, and we generated:

- 600 untouched images with no compression (TIFF format);
- 600 untouched and JPEG compressed images;
- 100×4 spliced images without AF, generated using the 4 different cut&paste attacks reported in Table 2;
- 100×4 spliced images, that are simply stored after the cut-&-paste, without compression, according to Figure 4(a);
- 100×4 spliced images, to which AF is applied to remove trace of JPEG compression, according to Figure 4(b).

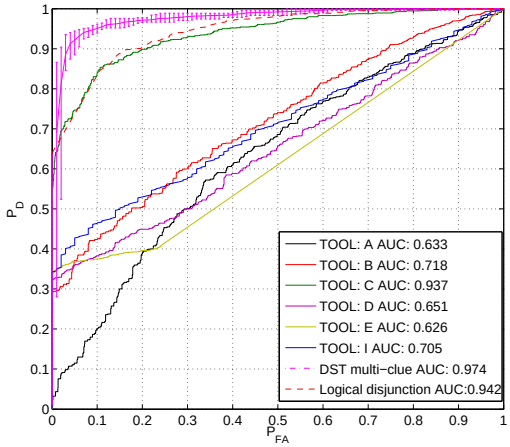


Figure 5. ROC curves obtained on the second case study dataset by each of the considered IF and CAF tools (solid lines), and by the two decision fusion methods (dashed lines). The ROC curve relative to the DST-based method shows the maximum and minimum value obtained through all the train-test iterations.

We used the same parameters (tampering size, compression strengths, etc.) that were described in Section 4.1.1. Uncompressed images were analysed both using the CAF tool[19] and the JPEG-based tools, after compressing them with quality 95.† On the other hand, the CAF tool was not run on JPEG-compressed images, mapping its output to a vacuous belief assignment. We also maintained the same partitioning between training and test samples for evaluating the DST based multi-clue framework. Results obtained by single tools and fusion techniques are plotted in Figure 5, and confirm that multi-clue analysis allows the analyst to effectively counter the presence of AF techniques. Interestingly, the tool based on JPEG ghost[15] yields good performance notwithstanding the presence of JPEG anti-forensics. We actually found that, even after application of the employed AF tool,[18] the algorithm Farid et al. was still able to detect the pasted region in some cases. This is probably due to the way the tool works, namely accumulating in the spatial domain the contribution coming from all DCT coefficients, without explicitly modelling their histograms.

5. CONCLUSIONS

In this paper we investigated the use of data fusion as a tool for countering anti-forensics. We considered a recently proposed framework based on Dempster-Shafer Theory,[2] and we studied how CAF tools can be embedded within such a scheme, highlighting the key points that distinguish CAF traces from standard IF traces. Then, we considered two practical case studies to investigate the impact of data fusion in the presence of an adversary equipped with AF techniques: results show that, as opposed to the unsatisfactory performance allowed by the use of single IF tools, much more interesting results can be obtained by merging them by using the proposed architecture.

ACKNOWLEDGMENTS

This work was partially supported by the European Office of Aerospace Research and Development under Grant FA8655-12-1-2138: AMULET - A multi-clue approach to image forensics, and by the REWIND Project funded by the FP7-Future and Emerging Technologies (FET) Programme under grant 268478. We thank G. Valenzise, M. Tagliasacchi e S. Tubaro for kindly providing us a software implementing their algorithm.[19]

†It has been shown[8] that the best case for JPEG-based forensic analysis is when the last encoding was at high quality, but not the highest possible.

REFERENCES

[1] Piva, A., "An overview on image forensics," *ISRN Signal Processing* **2013** (2013).

[2] Fontani, M., Bianchi, T., De Rosa, A., Piva, A., and Barni, M., "A framework for decision fusion in image forensics based on Dempster-Shafer Theory of Evidence," *Information Forensics and Security, IEEE Transactions on* **8**(4), 593–607 (2013).

[3] Barni, M. and Costanzo, A., "A fuzzy approach to deal with uncertainty in image forensics," *Signal Processing: Image Communication* **27**(9), 998 – 1010 (2012).

[4] Sun, Z.-W., Li, H., and Ji, Z.-C., "Fusion image steganalysis based on Dempster-Shafer evidence theory," *Control and Decision* **26**(8), 1192–1196 (2011).

[5] Böhme, R. and Kirchner, M., "Counter-forensics: Attacking image forensics," in [*Digital Image Forensics*], Sencar, H. T. and Memon, N., eds., 327–366, Springer New York (2013).

[6] Dempster, A. P., "Upper and lower probabilities induced by a multivalued mapping," *Annals of Mathematical Statistics* **38**, 325–339 (1967).

[7] Shafer, G., [*A Mathematical Theory of Evidence*], Princeton University Press, Princeton (1976).

[8] Fontani, M., Argones-Rúa, E., Troncoso, C., and Barni, M., "The Watchful Forensic Analyst: Multi-Clue Information Fusion with Background Knowledge," in [*Proc. of WIFS 2013*], 1–6 (2013).

[9] Barni, M., Fontani, M., and Tondi, B., "A universal attack against histogram-based image forensics," *International Journal of Digital Crime and Forensics (IJDCF)* **5**(3), 35–52 (2013).

[10] Kirchner, M. and Bohme, R., "Hiding traces of resampling in digital images," *Information Forensics and Security, IEEE Transactions on* **3**(4), 582–592 (2008).

[11] Bianchi, T. and Piva, A., "Detection of non-aligned double JPEG compression with estimation of primary compression parameters," in [*Proc. of ICIP 2011*], 1929 –1932 (sept. 2011).

[12] Luo, W., Qu, Z., Huang, J., and Qiu, G., "A novel method for detecting cropped and recompressed image block," in [*Proc. of ICASSP 2007*], **2**, II–217 –II–220 (Apr 2007).

[13] Bianchi, T., De Rosa, A., and Piva, A., "Improved DCT coefficient analysis for forgery localization in JPEG images," in [*Proc. of ICASSP 2011*], 2444–2447 (2011).

[14] Lin, Z. C., He, J. F., Tang, X., and Tang, C. K., "Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis," *Pattern Recognition* **42**, 2492–2501 (Nov. 2009).

[15] Farid, H., "Exposing digital forgeries from JPEG ghosts," *IEEE T. on Information Forensics and Security* **4**(1), 154–160 (2009).

[16] Kirchner, M. and Fridrich, J., "On detection of median filtering in digital images," in [*SPIE Conference Series*], *SPIE Conference Series* **7541** (Feb. 2010).

[17] Pevny, T., Bas, P., and Fridrich, J., "Steganalysis by subtractive pixel adjacency matrix," *Information Forensics and Security, IEEE Transactions on* **5**(2), 215–224 (2010).

[18] Stamm, M., Tjoa, S., Lin, W., and Liu, K., "Undetectable image tampering through JPEG compression anti-forensics," in [*Proc. of ICIP 2010*], 2109–2112 (2010).

[19] Valenzise, G., Tagliasacchi, M., and Tubaro, S., "Revealing the traces of JPEG compression anti-forensics," *Information Forensics and Security, IEEE Transactions on* **8**(2), 335–349 (2013).