[Article] A Framework for Decision Fusion in Image Forensics Based on Dempster-Shafer Theory of Evidence

A Framework for Decision Fusion in Image Forensics based on Dempster-Shafer Theory of Evidence

Abstract—**In this work we present a decision fusion strategy for image forensics. We define a framework that exploits information provided by available forensic tools to yield a global judgment about the authenticity of an image. Sources of information are modeled and fused using Dempster-Shafer Theory of Evidence, since this theory allows to handle uncertain answers from tools and lack of knowledge about prior probabilities better than the classical Bayesian approach. The proposed framework permits to exploit any available information about tools reliability and about the compatibility between the traces the forensic tools look for. The framework is easily extendable: new tools can be added incrementally with a little effort. Comparison with logical disjunction- and SVM- based fusion approaches shows an improvement in classification accuracy, particularly when strong generalization capabilities are needed.**

Index Terms—**Image Forensics, Image Tampering, Image Integrity, Decision Fusion, Dempster Shafer, Forgery Detection**

I. Introduction

Images have always played a key role in the transmission of information, mainly because of their presumed objectivity. However, in the last years the advent of digital imaging has given a great impulse to image manipulation, and nowadays images are facing a thrust crisis. Image Forensics, whose goal is to investigate the history of an image using passive (*blind*) approaches, has emerged as a possible way to solve the above crisis.

The basic idea underlying Image Forensics is that most, if not all, image processing tools leave some (usually imperceptible) traces into the processed image, and hence the presence of these traces can be investigated in order to understand whether the image has undergone some kind of processing or not. In the last years many algorithms for detecting different kinds of traces have been proposed (see [1] for an overview) which usually extract a set of features from the image and use them to classify the content as exposing the trace or not. Very often, the creation of a forgery involves the application of more than a single processing tool, thus leaving a number of traces that can be used to detect the presence of tampering; this consideration suggests to analyze the authenticity of images by using more than one tamper detection tool. Furthermore, existing forensic tools are far from ideal and often give uncertain or even wrong answers, so, whenever possible, it may be wise to employ more than one tool searching for the same trace. On top of that, it may also be the case that the presence of one trace inherently implies the absence of another, because the traces are mutually exclusive by definition. For these reasons, taking a final decision about the authenticity of an image relying on the output of a set of forensic tools is not a trivial task, thus justifying the design of proper decision fusion methods explicitly thought for this scenario. Given that new forensic tools are developed continuously, we would like our decision fusion method to be easily extendable, so that new tools can be included as soon as they become available. Another key issue regards the creation of training datasets for the fusion stage: while producing datasets for training single tools is a rather simple task, creating datasets representing the variety of possible combinations of traces that could be introduced during the creation of a realistic forgery is extremely challenging.

As an answer to the above needs, we propose a decision fusion framework for the image forensics scenario based on Dempster-Shafer Theory of Evidence (DST); the proposed model is easily extendable and, as a key contribution, allows incremental addition of knowledge when new tools become available. With respect to more classical approaches to inference reasoning, the use of DST avoids the necessity of assigning prior probabilities (that would be extremely difficult to estimate) and also provides more intuitive tools for managing the uncertain knowledge provided by the forensic tools. This paper extends a previous work by Fontani et al. [2] both from a theoretical and an experimental points of view. The most significant novelty is that tools and searched traces are modeled in a more flexible way, specifically, a mechanism for hierarchical fusion of traces is introduced, leading to a key improvement of framework extendability. Moreover, the number of implemented tools has been raised to five and tests have been performed also over a realistic (hand-made) forgery dataset. Differences with respect to the previous work will be highlighted when necessary.

The rest of the paper is organized as follows. In the next subsection (I-A), we briefly introduce the problem of decision

M. Fontani and M. Barni are with the Department of Dept. of Information Engineering, University of Siena, via Roma 56, Siena, IT. E-mail: marco.fontani@unisi.it - barni@dii.unisi.it

T. Bianchi is with the Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Corso Duca degli Abruzzi 24, I-10129, Torino, Italy (phone: +39 011 0904070, fax: +39 011 0904099, e-mail: tiziano.bianchi@polito.it)

A. De Rosa and A. Piva are with the Dept. of Electronics and Telecommunications, University of Florence, via Santa Marta 3, Florence, IT. E-mail: tiziano.bianchi@unifi.it - alessandro.piva@unifi.it - alessia.derosa@unifi.it

fusion in an image forensics scenario, then we give some basic notion of DST (section II) and describe in detail the proposed framework (section III). In section IV, we present experimental results regarding a scenario in which the outputs of five image forensic tools ([3], [4], [5], [6] and [7]) are fused to give a global judgement about image authenticity. The results show a clear performance improvement with respect to more classical decision fusion strategies when realistic forgeries are examined.

A. Decision fusion in the image forensics scenario

The problem of taking a final decision about an hypothesis by looking at the output of several different tools is an important task in decision fusion; there are basically three kinds of approaches to tackle with it. The first is to perform fusion at the *feature* level: a subset of the features extracted by the tools is selected and used to train a global classifier. The second is to consider the (usually scalar) output provided by the tools and fuse them (fusion at the *measurement*, or *score*, level). The last approach consists in fusing the binary answers of the tools, usually obtained by binarizing their soft outputs (fusion at the *abstract* level). An effective example of how these three strategies can be applied to a problem similar to the one addressed in this paper is illustrated in [8], where fusion is used cast in a steganalysis framework. In fact, both in steganalysis and image forensics, tools usually extract some features from the image, perform measurements/classification on them and finally produce an output, often probabilistic, which can be thresholded to yield a binary classification.

Although being promising in terms of performance, fusion at the feature level has some serious drawbacks, most importantly the difficulty of handling cases involving a large number of features (commonly addressed as "curse of dimensionality") and the difficulty to define a general approach to feature selection, since ad-hoc solutions are needed for different cases. Furthermore, feature selection in most cases is followed by some machine learning, that by definition is effective only when a training dataset can be prepared that is representative of a large part of the global population of samples. If this can be done for training a single detector, creating a representative dataset of all possible image forgeries is practically unfeasible, especially in the case of photorealistic ones.

Working at the other extreme, the abstract level, suffers from the complementary problem: lots of information is discarded when outputs are thresholded, so the discrimination power of the various tools is not fully exploited. In image forensics, most of the existing works are based on the first approach [9] [10] [11]; an hybrid solution has been investigated in [12], but still focusing on feature fusion.

In order to get around the above problems, we choose to perform fusion at the measurement level. This choice delegates the responsibility of selecting features and training classifiers (or other decision methods) to each single tool, thus keeping the fusion framework more general and easy to extend, while avoiding the loss of important information about tool response confidences. Specifically, we present a fusion framework based on Dempster-Shafer's "Theory of evidence" (DST) [13] that focuses exclusively on fusion at the measurement level. The proposed framework exploits knowledge about reliability of tools and about compatibility between different traces of tampering, and can be easily extended when new tools become available. It allows both a "soft" and a binary (tampered/non-tampered) interpretation of the fusion result, and can help in analyzing images for which taking a decision is critical due to conflicting data. Note that a fusion approach involving DS Theory has already been proposed in [14], but such a scheme applies fusion at the feature level hence inheriting the general drawbacks of feature-level fusion, noticeably the lack of scalability and the need to retrain the whole system each time a new tool is added. Also, in [15] the authors exploit Dempster's combination rule, which provides only a limited part of the expressive capability of the DST framework, to devise an image steganalysis scheme that combines three algorithms to improve detection accuracy; however, our goal is deeply different from that pursued in [15], since we do not aim at providing a specific multi-clue forgery detection tool, but at defining a theoretical model that allows fusing a generic set of tools targeting splicing detection. As we will see later in the paper, the combination rule by itself is not sufficient to address our problem, since we must deal with heterogeneous and evolving sources of information.

II. Dempster-Shafer's theory of evidence

Dempster-Shafer's theory of evidence was firstly introduced by A. Dempster [16] and further developed by G. Shafer [13]. It can be regarded as an extension of the classical Bayesian theory that allows representation of ignorance and of available information in a more flexible way. When using classical probability theory for defining the probability of a certain event A, the additivity rule must be satisfied; so by saying that $Pr(A) = p_A$ one implicitly says that $Pr(\bar{A}) = 1 - p_A$, thus committing the probability of an event A to that of its complementary $\bar{A}$. Most importantly, the additivity rule influences also the representation of ignorance: complete ignorance about a dichotomic event A in Bayesian theory is best represented by setting $Pr(A) = Pr(\bar{A}) = 0.5$ (according to the maximum entropy principle), but this probability distribution also models perfect knowledge about the probability of each event being 0.5 (as for a coin tossing), thus making it difficult to distinguish between ignorance and perfectly known equiprobable events. Since reasoning in a Bayesian framework makes an extensive use of prior probabilities, which are often unknown, a wide usage of maximum entropy assignments is often unavoidable, leading to the introduction of extraneous assumptions. To avoid that, DS theory abandons the classical probability frame and allows to reason without a-priori probabilities through a new formalism.

A. Shafer's formalism

Let the frame $\Theta = \{x_1, x_2, \ldots, x_n\}$ define a finite set of possible values of a variable X; a proposition about variable X is any subset of Θ. We are interested in quantifying how much we are confident in propositions of the form "the true value of X is in H", where $H \subseteq \Theta$ (notice that the set of all possible propositions is the power set of Θ, 2^{Θ}). To give an example, let us think of a patient that can either be affected by cancer or not: we can model this scenario defining a variable C with frame $\Theta = \{ac, nc\}$ where ac is the proposition "patient is affected by cancer", nc is the proposition "patient is not affected by cancer", and $(ac \cup nc)$ is the doubtful proposition "patient is or is not affected by cancer". The link between propositions and subsets of Θ allows to map logical operations on propositions into operations among sets. Each proposition is mapped onto a single subset and is assigned a basic belief *mass* through a Basic Belief Assignment, defined over the frame of the variable.

Definition 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.

Continuing the previous example, a doctor after examining the patient could provide information that lead us to write the following basic belief assignment:

$$m^{\Theta}(X) = \begin{cases} 0.8 & \text{for } X = \quad \{ac\} \\ 0.2 & \text{for } X = \quad \{nc\} \\ 0 & \text{for } X = \{ac \cup nc\} \end{cases} \tag{2}$$

Each set S such that $m(S) > 0$ is called a *focal element* for m. In the following, we will omit the frame when it is clear from the context, writing m instead of m^{Θ}; furthermore, when writing mass assignments only focal elements will be listed (so the last row of eq. (2) would not appear). BBAs are the atomic information in DST, much like probability of single events in probability theory. By definition, $m(A)$ is the part of belief that supports exactly A but, due to lack of knowledge, does not support any strict subset of A, otherwise the mass would "move" into the subsets. In the previous example, if we had assigned mass 0.85 to proposition $\{ac \cup nc\}$ and 0.15 to $\{ac\}$ it would have meant that there is some evidence for the patient being affected by cancer but, basing on current knowledge, a great part of our confidence cannot be assigned to none of the two specific propositions. Whenever we have enough information to assign all of the mass to singletons[1], DST collapses to probability theory.

Intuitively, if we want to obtain the total belief for a set A, we must add the mass of all proper subsets of A plus the mass of A itself, thus obtaining the *Belief* for the proposition A.

Definition 2: Given the BBA in 1, the Belief function $Bel : 2^{\Theta} \rightarrow [0,1]$ is defined as follows:

$$Bel(A) = \sum_{B \subseteq A} m(B)$$

$Bel(A)$ summarizes all our reasons to believe in A with the available knowledge. There are many relationships between $m(A)$, $Bel(A)$ and other functions derived from these; here we just highlight that $Bel(A) + Bel(\bar{A}) \leq 1 \; \forall A \subseteq \Theta$ and $1 - (Bel(A) + Bel(\bar{A}))$ is the lack of information (or the amount of doubt) about A.

B. Combination Rule

If we have two BBAs defined over the same frame, which have been obtained from two independent sources of information, we can use Dempster's *combination rule* to merge them into a single one. Notice that the concept of independence between sources in DST is not rigorously defined (as it is, for example, in Bayesian theory): the intuition is that we require that the different pieces of evidence have been determined by different (*independent*) means [17].

Definition 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:

$$m_{12}(X) = \frac{1}{1-K} \cdot \sum_{\substack{A,B \subseteq \Theta: \\ A \cap B = X}} m_1(A) m_2(B) \tag{3}$$

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 $Bel_1 \oplus Bel_2$.

K is a measure of the *conflict* between m_1 and m_2: the higher the K, the higher the conflict. The meaning of K can be understood from its definition, since K is obtained by accumulating the product of masses assigned to sets having empty intersection (which means incompatible propositions). Furthermore, we see that Dempster's combination rule treats conflict as a normalization factor, so its presence is no longer visible after fusion.

Recall the example in section II-A, and suppose that we obtain evidence coming from another doctor, who is not a cancer specialist, about the variable C. Let us call m_1 the BBA in eq. (2) and m_2 the new assignment; so we have:

$$m_1(X) = \begin{cases} 0.8 & \text{for } X = \quad \{ac\} \\ 0.2 & \text{for } X = \quad \{nc\} \end{cases}$$

$$m_2(X) = \begin{cases} 0.1 & \text{for } X = \quad \{ac\} \\ 0.9 & \text{for } X = \{ac \cup nc\} \end{cases}$$

Note that since the second doctor is not a specialist the information he provides is quite limited: most of the mass is assigned to doubt. Fusing the two pieces of information according to Dempster's rule results in:

$$m_{12}(X) = \begin{cases} \frac{0.8 \cdot 0.1 + 0.8 \cdot 0.9}{1 - (0.1 \cdot 0.2)} = 0.816 & \text{for } X = \quad \{ac\} \\ \\ \frac{0.2 \cdot 0.9}{1 - (0.1 \cdot 0.2)} = 0.184 & \text{for } X = \quad \{nc\} \end{cases}$$

We see that after fusion values are not far from those already assigned by m_1: this is perfectly intuitive, since the second doctor did not bring a clear contribution to the diagnosis. Notice also that for the same reason, and for the low confidence of first doctor about absence of cancer, little conflict is observed ($K = 0.02$).

[1] A singleton is a set with exactly one element.

Dempster's rule has many properties [18], in this work we are mainly interested in its associativity and commutativity, that is:

$$Bel_1 \oplus (Bel_2 \oplus Bel_3) = (Bel_1 \oplus Bel_2) \oplus Bel_3 \qquad (4)$$

$$Bel_1 \oplus Bel_2 = Bel_2 \oplus Bel_1 \qquad (5)$$

Despite its many desirable properties, Dempster's rule is not idempotent; this means that observing twice the same evidence results in stronger beliefs. This is the reason why we need to introduce the hypothesis of independent sources in Dempster's combination rule. In practice, before letting a new source of information enter the system, we must always look at how the new information is collected, to ensure that we are not counting twice the same evidence. In our example, we must be sure that doctors did not talk with each other, did not use the same technology when performing measurements, and so on.

The combination rule expressed in eq. (3) is applicable if the two BBAs, m_1 and m_2, are defined over the same frame, which means that they refer to the same propositions. Whenever we need to combine BBAs defined over different frames, we have to redefine them on the same target frame before the combination. This can be done by using *marginalization* and *vacuous extension*.

Definition 4: Let m^{Θ} be a BBA function defined over a frame Θ, and let Ω be another frame. The vacuous extension of m^{Θ} to the product space $\Theta \times \Omega$, denoted with $m^{\Theta \uparrow \Theta \times \Omega}$, is defined as:

$$m^{\Theta \uparrow \Theta \times \Omega}(X) = \begin{cases} m^{\Theta}(A) & \text{if } X = A \times \Omega, A \subseteq \Theta \\ 0 & \text{otherwise} \end{cases}$$

This allows to extend the frame of a BBA without introducing extraneous assumptions (no new information is provided about propositions that are not in Θ). That said, vacuous extension is not the only possible way to extend a BBA to a larger frame: it just provides the "least informative" extension.

The inverse operation of vacuous extension is marginalization.

Definition 5: Let m^{Θ} be a BBA function defined on a domain Θ, its marginalization to the frame $\Gamma \subseteq \Theta$, denoted with $m^{\Theta \downarrow \Gamma}$, is defined as

$$m^{\Theta \downarrow \Gamma}(X) = \sum_{A \downarrow X} m^{\Theta}(A)$$

where the index of the summation denotes all sets $A \subseteq \Theta$ whose projection on Γ is X.
To outline the projection operator, let us introduce two product frames Θ and Γ, that are obtained as the cartesian product of the frames of some variables. Formally, we have $\Theta = F_1 \times F_2 \times \cdots \times F_k$ and $\Gamma = F_{S_1} \times F_{S_2} \cdots \times F_{S_z}$, where F_j is the frame of the j-th variable and S is a subset of the indices in $\{1, \ldots, k\}$. Each element of Θ will be a vector whose j-th component is a value in F_j.[2] The projection operator maps each element $\theta \in \Theta$ into an element of $\gamma \in \Gamma$ by removing from θ all the components whose indices are not in S.[3]

The importance of extension and marginalization is that they allow to combine over a common frame BBAs originally referring to different frames, hence enabling us to fuse them with Dempster's rule.

III. DST-based decision fusion in image forensics

By using the basic instruments of DST introduced in the previous section, we developed a framework for combining evidence coming from two or more forgery detection algorithms. In particular we focus on the splicing detection problem, which consists in determining if a region of an image has been pasted from another. As already stated, during this process some traces are left into the image, depending on the modality used to create the forgery. The presence of each of these traces can be revealed by using one (or more) image forensic tools, each of which provides information about the presence of the trace it is looking for. Note that, in splicing *detection* tasks, most forensic tools assume knowledge of the suspect region. That said, if no information is available, we could still run all tools in a block-wise fashion, and fuse their outputs at the block level. As we will highlight in Section V, forgery *localization* is a different problem, that we will consider from the decision fusion point of view in future works.

A. Assumptions

Our framework for decision fusion relies on the basic assumptions listed below:

1) Each tool outputs a number in [0,1], where higher values indicate higher confidence about the analyzed region containing the searched trace;
2) Compatibility relations among some or all of the considered traces are known, at least theoretically (for instance, we may know that two tools search for mutually-exclusive traces).
3) Information about tools reliability, possibly image dependent, is available (for instance such an information could derive from published results or from experimental evidence);
4) Each tool gathers information independently of other tools (i.e. a tool is never employed as a subroutine of another, and no information is exchanged between tools), and by different means (each tool relies on a different principle or effect);

These assumptions are very reasonable in the current image forensics scenario; nevertheless, some of them can be relaxed with a limited impact on our framework. For example, in Section III-D we discuss how to handle the case where some relationships between traces are not known, and in Section IV-B we show that errors in estimating tool reliabilities do not affect overall performance significantly.

Notice that assumption 4 is needed to ensure that we can fuse tool responses using Dempster's rule. Intuitively, it means

[2] For instance, if $\Theta = X \times Y \times Z$ one possible element of Θ is (x_1, y_3, z_1), where $x_1 \in X$, $y_3 \in Y$ and $z_1 \in Z$.

[3] For example, if we project the set $\Theta = X \times Y \times Z$ to $\Gamma = X \times Z$ the element $(x_1, y_3, z_1) \in \Theta$ reduces to $(x_1, z_1) \in \Gamma$

that if we observe two different tools supporting the same proposition, we are more confident than observing only one. On the other hand, if two tools that search for the same trace exploiting the same model are available, it makes sense to discard the less reliable one, since its contribution will be limited or null. That said, and also considering that the concept of independence in DST is not equivalent to statistical independence, we believe that possible limited dependencies between algorithms would not undermine the developed framework.

B. *Formalization for the single-tool case*

For sake of clarity, we start by formalizing the DST framework when only one tool is available, let us call it *ToolA*, which returns a value $A \in [0,1]$ and has a reliability $R \in [0,1]$. While in our previous work [2] we directly modeled the output of the tool with a variable, here we propose a different point of view, that improves the extendability and generality of the framework: we focus on the *trace* searched by the tool, and we consider the information coming from $ToolA$ about the trace, by introducing a variable α, with frame $\Theta_\alpha = \{t\alpha, n\alpha\}$, where $t\alpha$ is the proposition "trace α is present" and $n\alpha$ is the proposition "trace α is not present". We model the information provided by *ToolA* about the presence of α with the following BBA over the frame Θ_α:

$$m_A^{\Theta_\alpha}(X) = \begin{cases} A_T & \text{for } X = \quad \{(t\alpha)\} \\ A_N & \text{for } X = \quad \{(n\alpha)\} \\ A_{TN} & \text{for } X = \quad \{(t\alpha) \cup (n\alpha)\} \end{cases} \tag{6}$$

where A_T, A_N and A_{TN} are functions (see next section and Fig.1) of the response A of the tool. We see that this BBA assigns a mass to every element of the power set of Θ_α; $\{(t\alpha) \cup (n\alpha)\}$ is the doubt that *ToolA* has about the presence of the trace, so it refers to the proposition "trace α is either present or not".

1) Detection mapping: The way A is mapped into A_T, A_N and A_{TN} is an *interpretation* of *ToolA* response and is simply models the behavior of the tool. For example we may know, either from theory or from experimental results, that any value above 0.5 should be interpreted as a strong belief about the trace being present, so A_T should be near to 1 when tool output crosses 0.5. We formalize this concept introducing three functions $\mu_T(\cdot)$, $\mu_N(\cdot)$ and $\mu_{TN}(\cdot)$, all from [0,1] to [0,1], which map the detection score (A in this case) to a value for A_T, A_N and A_{TN} respectively. These functions can be either obtained from theoretical analysis or from experiments (training); since each tool will probably distribute its output in the interval [0,1] in a characteristic way, we do not impose a general rule for performing these assignments. What is important is that they depend only on the specific tool, so they do not require any cross-tool information. An example of mapping is given in Fig.1.

2) Incorporation of reliability within the framework: When we combine evidence using Dempster's rule, it is assumed that masses are assigned by reliable sources; if we have some information about the reliability of the sources, then it should be taken into account. This can be done through a mechanism called *discounting* [13], which permits to weigh each source by its reliability. If we denote with A_R the reliability of $ToolA$, applying discounting to the BBA in eq. (6) yields[4]:

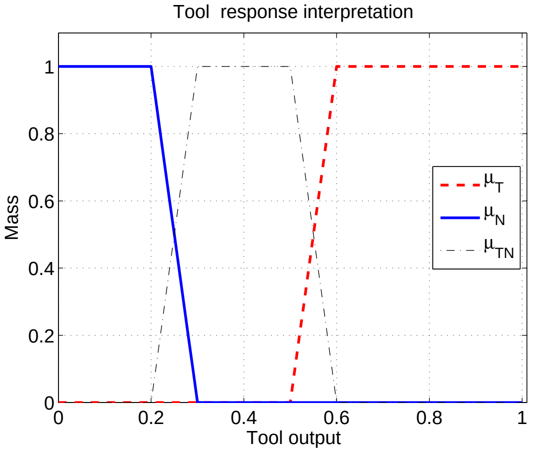


Fig. 1. An example of mapping the tool answer to mass assignments. Such an example is typical for tools featuring a good separation between positive and negative examples: most training examples yield very low (< 0.2) or high (> 0.6) values, while nothing is observed in the middle range. Therefore, should the tool provide 0.5 as output, we can not say anything about the class of the sample (i.e., we have only doubt)

$$m_{A_{tot}}^{\Theta_\alpha}(X) = \begin{cases} A_R \cdot A_T & \text{for } X = \quad \{(t\alpha)\} \\ A_R \cdot A_N & \text{for } X = \quad \{(n\alpha)\} \\ C_A & \text{for } X = \{(t\alpha) \cup (n\alpha)\} \end{cases} \tag{7}$$

where $C_A = (1 - A_R(A_T + A_N))$. The effect of discounting is better understood by considering the extreme cases for reliability: if the tool is fully reliable ($A_R = 1$) its response is totally believed, while if it is fully unreliable ($A_R = 0$) we do not say anything about the presence of the trace since all the mass moves to doubt.

C. *Introducing new tools*

Suppose we want to introduce in our framework a new tool, *ToolB*, that satisfies the assumptions in section III-A. Differently from our previous work [2], we distinguish two cases: the new tool may either search for a trace that is already considered in the framework, or for a novel trace; these cases are addressed differently in our framework.

1) Introduction of a tool looking for a known trace: If the trace searched by the new tool is already present in the framework (let us call it α, as in section III-B), applying the procedure in section III-B will produce $m_{B_{tot}}^{\Theta_\alpha}$, which can be directly fused with $m_{A_{tot}}^{\Theta_\alpha}$ by using Dempster's rule, yielding:

$$m_{AB_{tot}}^{\Theta_\alpha}(X) = \frac{1}{1-K} \cdot \begin{cases} \begin{matrix} A_R \cdot A_T \cdot C_B + C_A \cdot B_R \cdot B_T \\ + A_R \cdot A_T \cdot B_R \cdot B_T \end{matrix} & \text{for } X = \quad \{(t\alpha)\} \\ \begin{matrix} A_N \cdot A_R \cdot C_B + C_A \cdot B_N \cdot B_R \\ + A_N \cdot A_R \cdot B_N \cdot B_R \end{matrix} & \text{for } X = \quad \{(n\alpha)\} \\ C_A \cdot C_B & \text{for } X = \{(t\alpha) \cup (n\alpha)\} \end{cases} \tag{8}$$

where $K = A_N \cdot A_R \cdot B_T \cdot B_R + A_T \cdot A_R \cdot B_N \cdot B_R$. This BBA contains the information about the trace α gathered by the two distinct tools. We see that conflict is non-null, and is obtained

[4] The formal derivation of this formula is provided in Appendix A.

by summing the masses for propositions in which the tools are reliable but provide conflicting information about the presence of the trace. It is worth repeating that before introducing a new tool into the framework, the user should understand how the tool works and ensure that it does not replicate the job of a tool that is already present, since this would violate the request of independence of sources, and lead to a overestimation of the presence of the trace the new tool is looking for.

2) Introduction of a tool looking for a new trace: If $ToolB$ searches for a novel kind of trace, say β, we have to introduce it into the framework defining a new frame $\Theta_\beta = \{t\beta, n\beta\}$, where the propositions have the same meaning as in section III-B. The response of $ToolB$ will be used to assign masses to the variable Θ_β, and application of discounting will lead us to $m^{\Theta_\beta}_{B_{tot}}$. Since α and β are defined over different frames, $m^{\Theta_\alpha}_{A_{tot}}$ and $m^{\Theta_\beta}_{B_{tot}}$ cannot be fused directly. We first need to define a common frame $\Theta_\alpha \times \Theta_\beta$, so that we can (vacuously) extend both $m_{A_{tot}}$ and $m_{B_{tot}}$ to it and finally fuse them, yielding:

$$m^{\Theta_\alpha \times \Theta_\beta}_{AB_{tot}}(X) = \begin{cases} A_R{\cdot}A_T{\cdot}B_R{\cdot}B_T & \text{for } X{=} \quad \{\ (t\alpha,\ t\beta)\} \\ A_R{\cdot}A_T{\cdot}B_R{\cdot}B_N & \text{for } X{=} \quad \{(t\alpha,\ n\beta)\ \} \\ A_R{\cdot}A_T{\cdot}C_B & \text{for } X{=} \ \{(t\alpha,\ t\beta) \cup (t\alpha,\ n\beta)\ \} \\ A_R{\cdot}A_N{\cdot}B_R{\cdot}B_T & \text{for } X{=} \quad \{(n\alpha,\ t\beta)\ \} \\ A_R{\cdot}A_N{\cdot}B_R{\cdot}B_N & \text{for } X{=} \quad \{(n\alpha,\ n\beta)\ \} \\ A_R{\cdot}A_N{\cdot}C_B & \text{for } X{=} \ \{(n\alpha,\ t\beta) \cup (n\alpha,\ n\beta)\ \} \\ C_A{\cdot}B_R{\cdot}B_T & \text{for } X{=} \ \{(t\alpha,\ t\beta) \cup (n\alpha,\ t\beta)\ \} \\ C_A{\cdot}B_R{\cdot}B_N & \text{for } X{=} \ \{\ (t\alpha,\ n\beta) \cup (n\alpha,\ n\beta)\} \\ C_A{\cdot}C_B & \text{for } X{=} \quad \{(t\alpha,t\beta)\cup(n\alpha,t\beta) \\ & \qquad \cup(t\alpha,n\beta)\cup(n\alpha,n\beta)\} \end{cases}$$

Notice that we are not considering whether traces α and β are compatible or not: we will take this information into account only later on, exploiting the associativity and commutativity of Dempster's rule. Consequently, as confirmed by the fact that $K = 0$ in the above formula, there is no reason why the two tools should be conflicting, since by now we are looking for "unrelated" traces.

The Procedures in section III-C1 and III-C2 can be repeated when another tool *ToolX* becomes available. The associativity of Dempster's rule, defined in eq. (4), allows to combine directly the BBA $m_{X_{tot}}$ of the new tool with the one currently available (that takes into account all the tools in the framework), so we will always need to extend the frame of, at most, two BBAs: this is a considerably smaller effort with respect to extending the BBA and computing the combination rule for all the tools.

We stress that, compared to [2], using traces as basic entities (instead of tools responses) strongly improves the extendability of the framework: as a matter of fact, while new tools are being released quite often, many of them search for an already known trace; if this is the case, introducing a new tool is very simple since only its BBA has to be extended.

D. Compatibility among traces

So far we have considered traces as if they were unrelated from each other. However, as we noted in III-A, this is not always the case in real applications. Suppose, for instance, that we have two traces α and β and suppose that, ideally, only some of their combinations are possible. For example, it may be that the presence of α implies the absence of β, so, at least ideally, two tools searching for these traces should never detect tampering simultaneously.

This information induces a *compatibility relation* between frames Θ_α and Θ_β, meaning that some of the elements of the cartesian product $\Theta_\alpha \times \Theta_\beta$ are impossible (and hence should be removed from the frame of discernment, because by definition it contains only *possible* values of the variables, see section II-A). However, since we do not know in advance which traces will be introduced in our framework, we need a way to include this knowledge only in the late stage of fusion. Fortunately, in DST we can easily model this information by using a standard belief assignment: we define a BBA on the domain $\Theta_\alpha \times \Theta_\beta$, that has only one focal set, containing the union of all propositions (i.e, combination of traces) that are considered possible, while all others have a null mass. For example the following BBA:

$$m_{comp}(X) = \begin{cases} 1 \text{ for } X{=}\ \{(t\alpha,n\beta)\cup(n\alpha,t\beta)\cup(n\alpha,n\beta)\} \\ 0 \text{ for } X{=} \qquad\qquad \{(t\alpha,t\beta)\} \end{cases} \tag{9}$$

models the incompatibility between traces α and β. Thanks to the commutative property of Dempster's combination rule, this BBA can be combined with those coming from traces in the final stage of fusion. In such a way, information about tools relationships are exploited only at the very end and hence do not hinder model extendability.

Notice that the given formulation encompasses also the case where the relationship between two traces is not known: it is sufficient to put those propositions where the two traces are present in both the focal set and the impossible set of m_{comp}, and this will automatically result in a void contribution for that combination of traces during fusion.

The last step of our decision fusion process consists in fusing the compatibility BBA defined above with the BBA obtained combining evidences from all the available tools, yielding a global BBA m_{FIN}. Notice that in this last application of Dempster's rule all the conflict that may arise is due to incompatibilities between traces. Although this conflict is normalized away by Dempster's rule, the value of K can be recorded and used to evaluate how "unexpected" the output of tools were. Very high values of conflict may indicate that the image under analysis does not respect the working assumptions of one or more tools. The overall decision fusion approach described so far is summarized in Fig.2 for the case of two tools.

It is worth noting that, we did not need to introduce a-priori probabilities about an image being original or forged, or prior probabilities of presence of traces: in a Bayesian framework, this would have been difficult to obtain.

E. Dealing with many traces: hierarchical modeling

Since the extension to novel traces is based on cartesian product of sets, the number of variables in the framework grows exponentially with the number of different traces. However, this consideration holds only if the user is interested in a fusion approach that fully preserves the granularity of information, meaning that, after fusing several different traces, the user wants to get the beliefs about presence/absence of

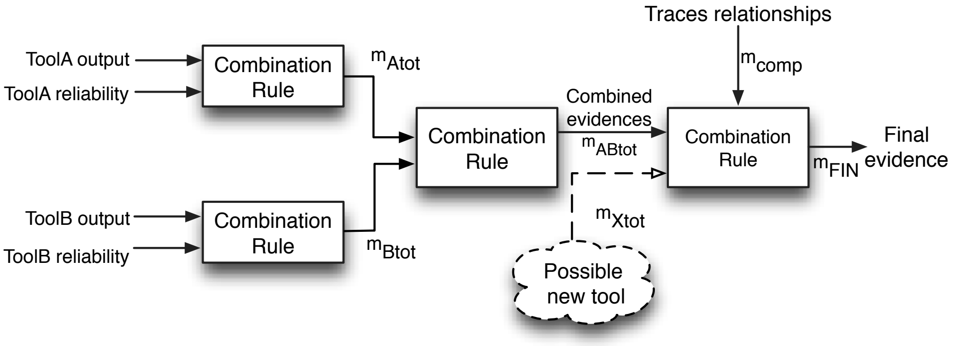


Fig. 2. Block diagram of the proposed fusion approach. Notice that, when a new tool becomes available (represented in the dashed cloud), its BBA directly enters the final stage of the fusion, without needing to recombine information from previous tools.

each single trace separately. In practice, however, the presence of many traces is probably due to the fact that the framework is taking into account different classes of phenomena, e.g. traces related to camera artifacts, to JPEG coding, to geometrical inconsistencies, and so on. In such a scenario, it makes sense to treat each class of traces as a whole, and directly consider the contribution of each class when taking the final decision. This hierarchical fusion can be easily implemented within the proposed framework by using belief marginalization (see Definition 5) to collapse the contribution of several traces of the same class into a single variable, thus reducing the granularity of the information without hindering performance in terms of splicing detection. In Fig.3 we draw an example of hierarchical fusion applied to three different kinds of traces. Furthermore, compatibility among classes of traces can be introduced as well, at the end of the fusion chain.

F. Final decision

We are now ready to define the final output of the fusion procedure: we want to know whether a given region of an image has been tampered with or not. To do so we consider the belief of two sets: the first one, T, is the union of all propositions in which at least one trace is detected, the second one, N, is the single proposition in which none of the traces is found (in the previous example it would be $N = (n\alpha, n\beta)$). The output of the fusion process therefore consists of two belief values, $Bel(T)$ and $Bel(N)$, calculated over the BBA m_{FIN} defined in section III-D. Optionally, we may also consider the normalization factor K (as defined in section II-B) of the last fusion step, involving the compatibility table. These outputs summarize the information provided by the available tools, without forcing a final decision. If a binary decision about image authenticity is required, an interpretation of these outputs has to be made; the most intuitive binarization rule is to classify an image as tampered with when the belief for the presence of at least one trace is stronger than the belief for the total absence of traces, that is to say when $Bel(T) > Bel(N)$. Of course, we will probably want to meet a minimum distance requirement between the two: a Receiver Operating Characteristic (ROC) curve can thus be obtained by classifying images according to $Bel(T) > Bel(N) + \delta$, sampling δ in [-1,1].

It is worth noting that evaluating belief values is a very simple task: only elementary operations among scalar values in [0,1] have to be calculated (see for example mass assignments in equation (8)), since the model is built only once for a fixed set of tools, and need to be extended only when new sources of information become available.

IV. Experimental results

In order to validate the effectiveness of the proposed approach, we compared it with three other methods. The first is one of those proposed in [12], where image manipulations are detected by taking the logical disjunction (OR) of the outputs of single tools. Logical disjunction is indeed one of the simplest and most widely used methods for decision fusion, and is quite well-suited to the proposed case study[5]. Furthermore, since we know the logical relationships between traces, the second method we compare with is a rule-based logical disjunction: in this case, only the combinations of binarized outputs that are consistent with known traces relationships are considered, while the others are discarded. By comparing with this technique, we want to investigate whether the proposed framework actually yields some advantages with respect to this simpler hard-reasoning method, based on the same prior knowledge about trace relationships.

As we mentioned in the Introduction, several methods have been proposed for decision fusion at the feature level in image forensics [9] [10] [11] [14], but they are typically based on feature selection and are therefore not directly comparable to the method proposed in this work. On the other hand, since most methods end up using a classifier (usually an SVM) the best we can do to compare our framework with them without exiting the measurement level is to train an SVM by using the scalar output of the tools as input features, and see how the SVM performs in discriminating between tampered and original images. Finally, we observe that our framework is not comparable with [15], because we are considering a complex scenario, where tools may search for compatible or incompatible traces, and more than one tool may be available

[5] Actually, taking the OR of binarized outputs is an "*abstract* level" approach. However, logical disjunction is one of the most used approaches among the post-classification ones [8], so we decided to compare our method against it.

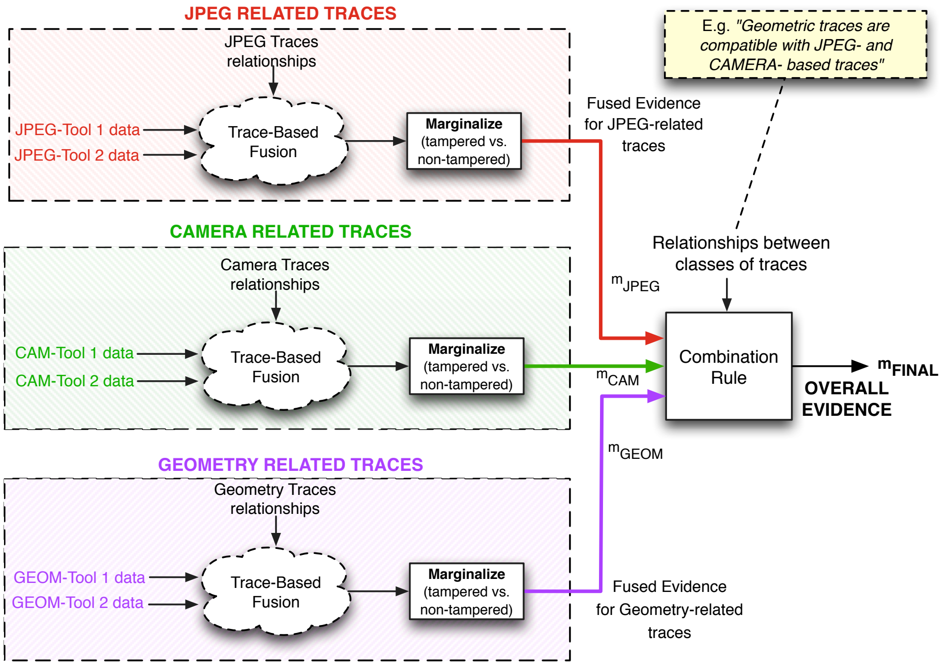

JPEG RELATED TRACES
JPEG Traces relationships
JPEG-Tool 1 data
JPEG-Tool 2 data
Trace-Based Fusion
Marginalize (tampered vs. non-tampered)
E.g. "Geometric traces are compatible with JPEG- and CAMERA- based traces"
Fused Evidence for JPEG-related traces
CAMERA RELATED TRACES
Camera Traces relationships
CAM-Tool 1 data
CAM-Tool 2 data
Trace-Based Fusion
Marginalize (tampered vs. non-tampered)
Relationships between classes of traces
m_JPEG
Combination Rule
m_CAM
m_FINAL
OVERALL EVIDENCE
m_GEOM
GEOMETRY RELATED TRACES
Geometry Traces relationships
GEOM-Tool 1 data
GEOM-Tool 2 data
Trace-Based Fusion
Marginalize (tampered vs. non-tampered)
Fused Evidence for Geometry-related traces

Fig. 3. Block diagram illustrating the proposed approach to hierarchical fusion of traces of different kind. The "Trace-Based Fusion" bubble represents the schema in Fig.2. For the sake of clarity, in the figure we write "Tool data" instead of separately drawing output and reliability assignments for each tool.

for the same trace, so applying directly Dempster's rule would lead to erroneous conclusions.

A. *Experimental setup*

As already stated, we evaluated the validity of the new DST fusion framework by focusing on the detection of splicing attacks: a portion of an image (source) is cut and pasted into another image (host), thus producing a new content that is finally saved. Because most images are stored in JPEG format, a great deal of research has been carried out for the identification and detection of traces left by splicing attacks in JPEG images, so several tools are available to search for them. In our experiments we fused the outputs obtained from five of these tools, searching for a total of three different traces.

1) Selected traces and tools: To explore all the features of the proposed scheme, we chose a set of algorithms such that some of them search for the same trace, and for which some combination of traces is not possible. Namely, we are considering the following traces (see Fig.4 for a graphical explanation):

1) *Misaligned JPEG compression* (JPNA): this trace shows up when the investigated region is cropped from a JPEG image and pasted into the target picture without preserving JPEG grid alignment, performing a final JPEG compression. Therefore, pasted pixels undergo two misaligned compressions, while others do not.
2) *Double quantization* (JPDQ): when a portion of uncompressed pixels[6] is pasted into a JPEG image, and the final result is JPEG saved, the untouched region undergoes a double compression. This causes its DCT coefficients to be doubly quantized, leaving a characteristic trace in their statistics.
3) *JPEG ghost* (JPGH): this trace is left when a region is cut-and-pasted, respecting grid alignment, from a JPEG source image into the host one (which has not been JPEG compressed). When the obtained splicing is JPEG saved, the inserted part undergoes a second compression, while the outer is compressed for the first time, thus introducing an inconsistency.

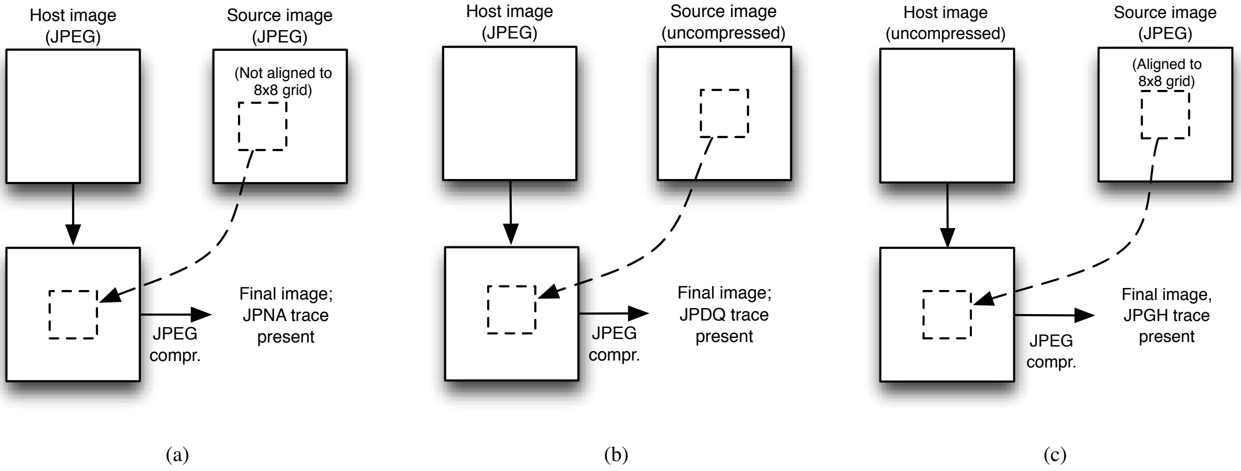
Given the above definitions, some combination of traces are not possible. For example an attack that introduces the JPDQ trace also introduces the JPGH, while the contrary is not necessarily true; but, if both JPGH and JPNA are introduced, then also JPDQ must be present. These facts are best represented by using a tabular form (see section III-D) with the compatibility relations, as in Tab. I.

TABLE I
DETECTION COMPATIBILITY: EACH COLUMN OF THE TABLE FORMS A COMBINATION OF PRESENCE (Y) AND ABSENCE (N) OF TRACES. WE SEE THAT ONLY 5 OUT OF 8 COMBINATIONS ARE POSSIBLE.

Trace	Possible					Excluded		
JPNA	Y	N	N	Y	N	Y	Y	N
JPDQ	N	Y	N	Y	N	Y	N	Y
JPGH	N	Y	Y	Y	N	N	Y	N

Now that we have introduced the traces considered in our experiments, we list the adopted forensic tools (see Tab. II). We employed two tools looking for JPNA, namely the one from Luo et al. [5] (*ToolA*) and the one from Bianchi et al. [4] (*ToolD*); two tools looking for JPDQ, the one from Lin et al.[6] (*ToolB*) and the one from Bianchi et al. [3] (*ToolE*); and the tool from Farid that searches for ghost traces [7] (*ToolC*).

[6] or pixels that have been compressed according to a different grid.



(a) (b) (c)

Fig. 4. In these schemes three different configuration of cut&paste attacks are reported. The attack in (a) introduces a misaligned double compression, the one in (b) introduces the double quantization effect in the untouched part of the final image and the attack in (c) introduces the ghost effect in the pasted region.

TABLE II
COUPLING BETWEEN TRACES AND TOOLS: FOR EACH TRACE, THE LIST OF ADOPTED TOOLS ABLE TO DETECT IT IS GIVEN.

Trace	**Tools**
JPNA	*ToolA* [5], *ToolD* [4]
JPDQ	*ToolB* [6], *ToolE* [3]
JPGH	*ToolC* [7]

In section III-A we assumed that each tool outputs a value in [0,1], where values near 1 indicate high confidence about the analyzed region containing the searched trace. Although not strictly necessary, normalization of tool outputs is desirable also for other fusion techniques, so we choose to adopt it as a common step before applying any fusion method in our experiments. In the following we give a brief description of how each of the selected tools works and define the approach we adopted to obtain a scalar output from it:

- *ToolA* searches for misaligned compression by measuring inconsistencies in blocking artifacts in the spatial domain. Because features are classified by using an SVM (which we trained on a separated dataset, according to the original work) we train a model supporting probability estimates [19];
- *ToolB* and *ToolE* search for double quantization effect employing two different statistical models to analyze the histogram of the DCT coefficients of the image. Both tools provide a probability map which gives, for each 8×8 pixel block, its probability of being original (i.e. showing double quantization) or tampered (not showing double quantization). The final detection value is taken as the median (over the suspect region only) of the probability map;
- *ToolC* searches ghost artifacts by re-compressing the image at several different qualities and taking the difference between the given image and the re-compressed one. Ghost effect is detected when the difference is small for the suspect region and not for the rest of the image. To evaluate how much the two regions are separated the KS statistic is used [7]. We directly take the value of this statistic as the detection value;
- *ToolD* searches for misaligned double compression exploiting the fact that DCT coefficients exhibit an integer periodicity when the DCT is computed according to the grid of the primary compression. Being the shift of the grid unknown, the algorithm searches among all possible shifts the one that minimizes a specific metric (see [4] for details). We scale and invert this metric from [0,6] to [0,1].

As mentioned at the beginning of this section, usually the simple presence of a trace does not imply a splicing attack, but just that a common processing over the image has taken place (for example, cropping a couple of rows from the top of the image would introduce a JPNA trace). Instead, *inconsistencies* in the presence of a trace through the image (i.e. high detection values for the suspect region and low for the other or vice-versa) are far more suspect. For this reason, each tool[7] is run both on the suspect region and on the remaining part of the image, and the absolute difference between the two outputs (which will still be in [0,1]) is considered.

Notice that since DST does not require that the masses assigned to propositions have a probabilistic meaning, we do not need to use more complex approaches to cast the output of each tool.

2) Training procedure: For all the fusion techniques used in the tests we need to run a training phase; however, in the proposed framework training is performed one tool at a time. The key idea is that if we can perform training separately for each tool, then we neither need a complicated learning technique, which would probably overfit on the training examples, nor we need huge datasets, since generating a dataset representing *one kind* of forgery is typically not difficult. In the following, the training procedures for each method are explained:

[7] *ToolC* is excluded since it already considers inconsistencies over the image.

- *DST fusion.* The DST based framework requires only to specify the mapping functions (μ_T, μ_N and μ_{TN}) of each algorithm, as defined in section III-B1. We found that it is not necessary to use a fitting technique for tuning these curves: it is sufficient to qualitatively consider how the output of a tool is distributed in the range [0,1] when the tool is run on forged and original images respectively[8], and draw trapezoidal functions consequently (see Fig.5). By relying on published results and on our tests about tools performance, we also defined the reliability of the various tools as follows: for *ToolA*, *ToolD* and *ToolE* reliability is a function of quality factor QF_2, according to published results (see, respectively [5], [4] and [3]), for *ToolB* we experimentally determined a B_R=0.4 and for *ToolC* we set C_R=0.85 according to experiments in [7].
- *SVM fusion.* A training example for the SVM is obtained from each image. The array of features is obtained by concatenating the outputs given for the image by each tool and the last quality factor[9], estimated from JPEG quantization tables. We use a RBF kernel with parameters (obtained through a 5-fold cross validation) $\gamma = 2.48$ and $C = 0.1$. For all the experiments, we repeated 20 times the training-testing phase, choosing different train- and test- datasets from the available examples (datasets are described later on).
- *OR-based fusion.* Since we are going to use Receiver Operating Characteristic (ROC) curves to compare the various methods, we need to train an *aggregate* ROC for the five algorithms, which represents their behavior in terms of probability of detection (p_D) and false alarm (p_{FA}) after being combined with the OR operator. To obtain these curves we uniformly sample (with precision 10^{-3}) the ROC of each algorithm, considering only images that satisfy the corresponding working assumptions, as reported in Tab. III. For each algorithm we save the threshold associated with each p_{FA}. During the test phase, given a target overall probability of false alarm $\hat{p}_{FA}$, we choose for each algorithm the threshold that gave a probability of false alarm of $\hat{p}_{FA}/5$, and we use that threshold to binarize its output. Binarized outputs for each image are combined with the OR operator, giving the final classification, that allows drawing a point of the overall ROC. Also in this case, the train-test procedure is repeated 20 times.
- *OR-based fusion with Hard Reasoning.* The same approach described for OR-based fusion is also used in the experiment with hard-reasoning logical disjunction (that will be abbreviated with OR-HR in figures). The difference is that, after binarization, only the combinations of outputs that are consistent with traces relationships will contribute to classify an image as tampered.

It is worth stressing that, training of the SVM for the considered decision fusion task requires not only to create a training dataset for each tool: a set of examples must be created for each possible *combination of traces* that the system should recognize. As we will show in the experiments, if the SVM is asked to classify a forgery containing a never-seen combination of traces, misclassifications are likely to occur. This fact has two consequences: first, the size of the training dataset grows exponentially with the number of different traces; secondly, only synthetically generated forgeries can be reasonably employed to create such a dataset. Unfortunately, synthetic examples are not really representative of the real world forgeries. This necessity is avoided with the DST framework, since determining mapping functions does not require any *cross-tool* training: a (usually quite small) dataset suffices to understand the behavior of the tool when is run on images that show exactly the trace the tool is looking for. The functions mapping the detection values into BBAs (eq. 6) for the selected tools are reported in Fig.5. These curves have been obtained considering the histogram of each algorithm outputs on the training dataset (both for original and tampered images) and mimicking it with trapezoidal functions, thus introducing a sort of smoothness constraint. Another possibility could be to fit a function to the cumulant of the histograms, however small variations in the shape of curves do not affects results significantly.

As shown in Fig.5, doubt is used only for regions where *few* examples from both tampered and original classes are observed. On the contrary, doubt is not employed when detection values overlap for a consistent number of examples, because this indicates that, on average, tools are “equally sure” about those images being tampered or original instead of unsure about both.

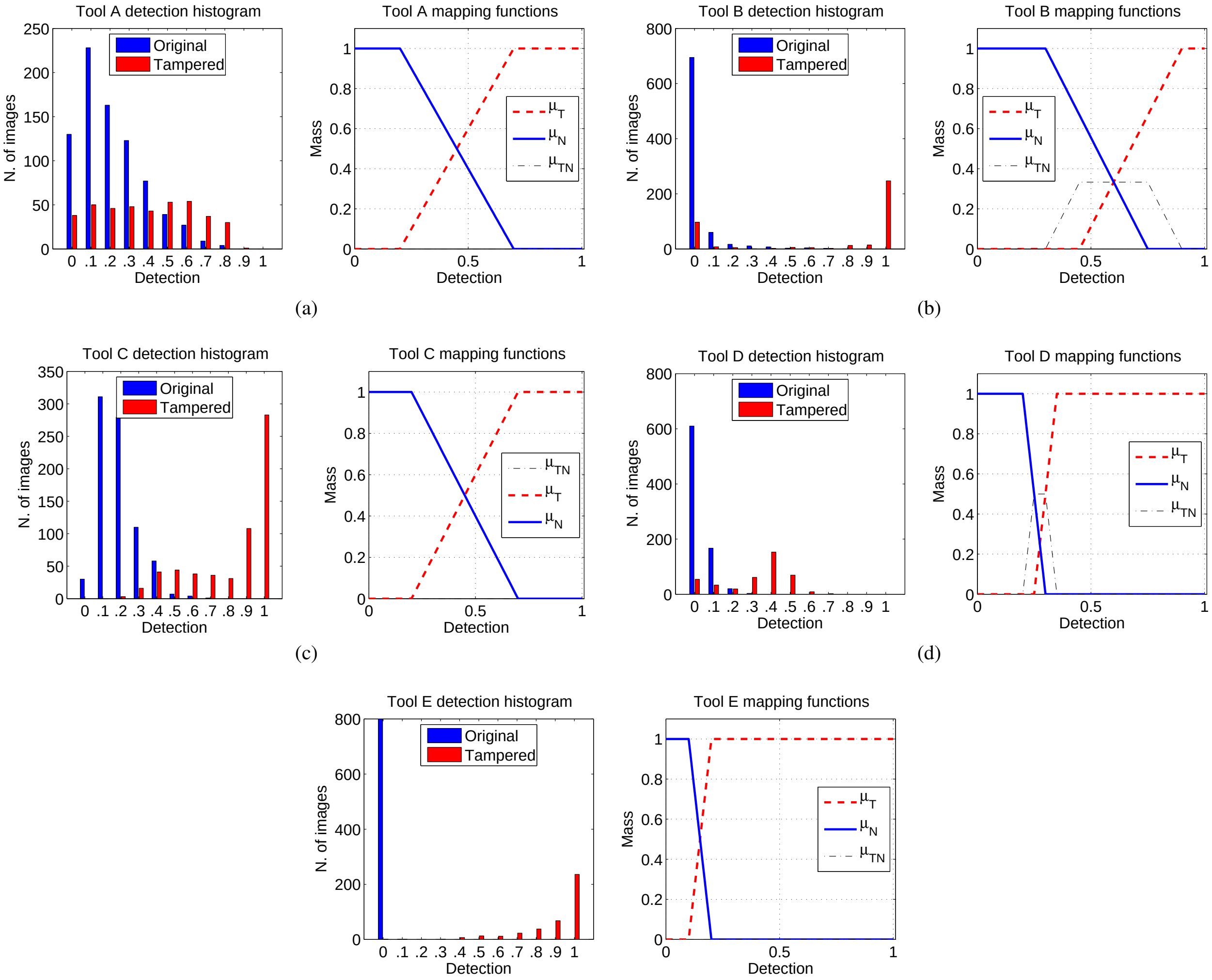
3) Datasets: As stated in section IV-A2, each of the three compared methods requires a training phase, so we created a dataset of 4800 tampered and 4800 original images. We considered four different tampering procedures (described in Tab. III) that start from an uncompressed image and automatically produce a forgery by cutting a portion (256x256 pixel) of the image and pasting it into a copy of itself, exactly in the same position. So doing, the forged image is perceptually above suspicion, and abrupt changes in content, that could influence the algorithms performance, are avoided. This also mimics the work of an image editing expert, which would limit discontinuities along the boundary of the tampered region. In every procedure the created splicing is JPEG compressed and saved. Notice that the four tampering classes we used cover all the possible combinations of presence/absence of the considered traces[10], as can be seen by comparing Tab. IV and Tab. I. The quality factor of the first compression, QF_1, is chosen randomly from the set {40, 50, ..., 80}, while the quality of the second compression is set to QF_2 = QF_1+20. Forgeries of the training set are equally distributed among these classes (1200 images per class).

The original, non-tampered, images are obtained by ap-

[8] In this phase we consider only images that match the working assumptions the tool is built on.

[9] We introduce this information because last quantization value strongly affects the reliability of tools. Since this information is used in the proposed model, not providing it to the SVM would be unfair.

[10] Notice also that a generic JPEG spliced image will almost surely fall in one of these classes.



(a)

(b)

(c)

(d)

(e)

Fig. 5. For each tool, the histogram of detection values on training dataset are shown on the left image. Right images show how this histogram is interpreted to define a mapping from tool detection value (x-axis) to mass assignment (y-axis). See eq. 6 for an explanation of each line meaning.

plying JPEG compression to uncompressed TIFF images (1024x1024 pixels), choosing randomly the quality factor from the set {40, 50, ..., 100}.

It is interesting to evaluate the performance of the considered decision fusion methods in two different scenarios:

1) in the first experiment, we use the synthetic images both to train and test the methods, of course with separated test and train datasets. The main goal of this experiment is to check if the training is effective for each method; we will refer to this dataset as the “synthetic” dataset. To show that the SVM really needs to have examples of all possible combinations of traces during training, we also train an “handicap”-SVM, where the handicap consists in removing all images belonging to Class 1 from the training set.
2) the second experiment mimics a realistic scenario. A team of students created 70 forgeries using common photo editing software, respecting only a constraint about JPEG quality factors: the quality factor of the final compression is always higher than the one of the host image. Students were asked to provide both tampered images (along with an indication of the attacked region) and original ones, for a total of 140 images. Although being rather small (creating good forgeries is a time consuming procedure) this dataset is crucial to understand how well the considered frameworks generalize to unseen cases. We will refer to this dataset as the “realistic” dataset. According to a realistic scenario, this dataset is used only for testing, training is still performed on synthetical images.

The train- and test- datasets we used in our experiments are available at the website http://clem.dii.unisi.it/~vipp/index.php/download/imagerepository

B. *Results and discussion*

We ran the five forensic tools on each dataset, then we combined their responses by the different fusion methods. To allow a comparison between tool performance we use ROC curves. However, both the SVM and the DST frameworks have been trained in order to maximize the *overall* accuracy,

TABLE III
PROCEDURE FOR THE CREATION OF DIFFERENT CLASSES OF TAMPERING IN THE TRAINING DATASET.

Class	Procedure	Result
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.	Inner region shows JPNA trace, external region does not. *Only tool A detects this trace.*
Class 2	Region is taken from an uncompressed image and pasted into a JPEG one; the result is saved as JPEG.	Outer region shows both JPDQ and JPGH traces, inner does not. *Tools B, E and C detect this trace*
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.	The inner region shows JPGH effect, the outer does not. *Only Tool C detects.*
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	The inner region shows JPNA, the outer shows JPDQ and JPGH. *All tools detect this trace.*

TABLE IV
THIS TABLE SPECIFIES WHETHER THE CLASS OF TAMPERING (COLUMN) SHOWS (Y) OR NOT (N) THE TRACE ON THE LEFT ROW. COMPARING THIS TO TAB. I SHOWS THAT OUR DATASET HAS A CLASS FOR EACH OF THE *possible* COMBINATIONS OF TRACES PRESENCE.

Trace	Class 1	Class 2	Class 3	Class 4	Original
JPNA	Y	N	N	Y	N
JPDQ	N	Y	N	Y	N
JPGH	N	Y	Y	Y	N

given by the percentage of correctly classified examples[11]. This urges us to consider only the portion of ROCs with reasonably low probabilities of false alarm; we fix the limit to 30%, a value that is far higher than acceptable ones in a standard forensic scenario. For each of the test datasets we report and comment the ROC curve (averaged over the 20 train-test iterations) obtained with each fusion method along with the ROC curves obtained by the single tools. Notice that since the proposed method does not require a training phase, no cross-fold validation is performed for it and we can use all the images to test it. For the other methods, we also plot uncertainty bars showing the maximum and minimum probability of detection obtained within the 20 iterations for several probability of false alarms. Values for the Area Under Curve (AUC) are normalized to the considered interval.

a) Results on the synthetic dataset: Fig.6 illustrates the results of the experiments taken on the synthetic dataset. Although being trained without cross-tool information (except for the traces compatibility table) the proposed method almost retains the same performance showed by the SVM (Fig. 6a). Considering that the synthetic test dataset is very similar to the training one, and considering also the high ratio of examples versus features (9600 images for 6 scalar normalized features), retaining the same performance of a SVM classifier is an undoubtedly good result. Both the SVM and the DST framework overcome the logical disjunction method, which nevertheless shows good performance. On the other hand, performance of the handicap-SVM are seriously hindered by the fact that some of the test images contain a combination of traces (namely, images belonging to Class 1) that were not in the training set. We considered the average performance (over 20 experiments) of the SVM in classifying test images belonging to Class 1: when the handicap is not present, the average accuracy is 77.2%; when the handicap-SVM is used, performance drops to 18.7%. This fact is extremely important since, as the number of traces increases, creating datasets with a sufficient number of forgery examples for each possible combination becomes complicated. On the other hand, the proposed method does not exhibit a significant performance deterioration. Notice that, in this experiment, the hard-reasoning method (blue curve in Fig.6a) yields the worst performance, meaning that application of rules from Tab. IV does not provide any help to the logical disjunction: this is reasonable, since images of this dataset are synthetically generated according to tool working hypotheses, and will unlikely expose unexpected combination of traces. However, looking at performance of single tools, in Fig. 6b, we see a clear benefit from the use of each of the decision fusion techniques.

We also used the synthetic dataset to investigate the sensitiveness of the DST-based framework to fluctuations of tool reliabilities. We repeated 20 times the classification, perturbing the reliability of tools with a gaussian error ($\mu = 0$, $\sigma = 0.15$), and collecting from each experiment the resulting AUC. We observed a standard deviation for the AUC of 0.02, thus showing the robustness of the framework against inaccurate estimation of reliabilities.

b) Results on the realistic dataset: Fig.7 shows results obtained using the realistic dataset. The inherent difference between synthetic training examples and real-world splicings has a direct implication on performance of single tools (compare Fig.6b with Fig.7b), and this is not surprising. What is really important is that performance of decision fusion methods are even more affected: the DST framework now clearly overcomes the SVM classifier, and also the hard-reasoning logical disjunction method behaves better than the SVM. This suggests that when knowledge about relationships between traces is available, we should introduce this knowledge as directly as possible instead of using machine learning methods, that are more suited to scenarios where knowledge is somewhat hidden in data. In forensics, such relationships are known most of the times, because they depend on some physical or logical phenomenon (e.g., camera interpolation or noise, JPEG quantization effects, shadow consistency, etc.) whose compatibility with other effects can be easily argued or measured with a targeted experiment.

Finally we point out that, except for very low probabilities of false alarms ($< 2\%$), the performance of the DST method is always better than those provided by each single tool.

We conclude this section discussing the computational time of our system. As stated in previous sections, framework definition and belief evaluation are two separated and different tasks: the first one is executed off-line when a new tool enters

[11] A training targeted to obtain a specified false alarm probability would require for the SVM to find an appropriate balance of the weights assigned to misclassified examples; and for the DST framework to adjust accordingly the mapping functions.

the system, producing a formula that is stored; this formula is then used for belief evaluation, which represents the on-line phase of the system and is much faster. Table V shows the time[12] needed to build the framework and to perform belief evaluation for the 5 tools used in this section. We also show how times would change if two more tools searching for one new trace enter the system.

TABLE V
EXECUTION TIMES (IN SECONDS) FOR FRAMEWORK DEFINITION AND BELIEF EVALUATION.

Num. of Tools	**2**	**3**	**4**	**5**	**6**	**7**
Definition	0.11	0.27	0.75	0.70	2.7	39.8
Evaluation	0.02	0.03	0.21	0.23	0.81	1.3

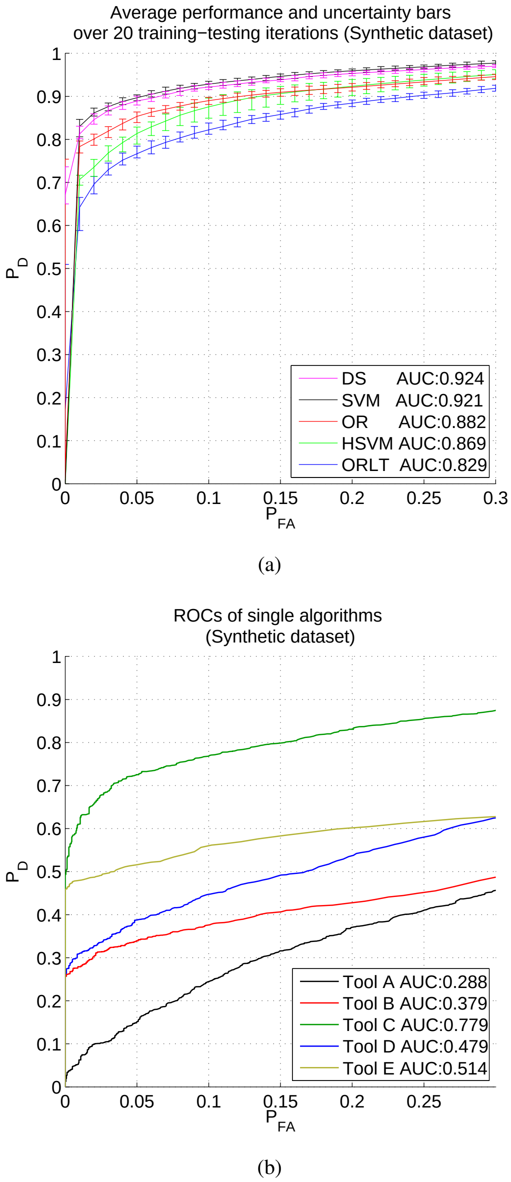
V. CONCLUSIONS

In this paper we have proposed a framework and discussed a system for data fusion in image forensics. The proposed system operates by fusing the output of a pool of forensic tools at the measurement level thus permitting to retain as much information as possible with regard to the single tool analysis, without incurring in the problems typical of feature fusion techniques (curse of dimensionality, training complexity, redundant features and so on).

The peculiarities of the proposed framework include: i) the use of a soft reasoning approach, based on Dempster-Shafer theory of evidence, to cope with the lack of a priori-information about the kind of tampering the image may have undergone, and the possibility that the available forensic tools provide incomplete and even conflicting evidence, ii) the ease with which new information can be included as soon as it becomes available, iii) the hierarchical structure of the framework that allows to trade-off between granularity of the information provided by the fusion system and the complexity of the update procedure when the information becomes available.

Experimental results are encouraging: the proposed model gives significantly better results than fusion approaches based on logical disjunction, and outperforms SVM-based fusion when tested against a realistic dataset.

We believe data fusion is a key ingredient to go beyond the current state of the art in image forensics, making it suitable to work in a real world setting where the strictly controlled conditions typical of laboratory experiments can not be enforced. Even more, we are convinced that data fusion can also help to cope with the proliferation of counter-forensics techniques. As a matter of fact, it is quite easy to fool a forensic analysis based on a single detection tool, especially if the algorithm the tool relies on is known to the attacker; however, facing with a pool of forensic tools, whose output is fused in a clever way, is likely to be a much harder piece of work, given that hiding a tampering trace may introduce additional traces whose presence can be spotted by other detectors in the pool.



(a)

(b)

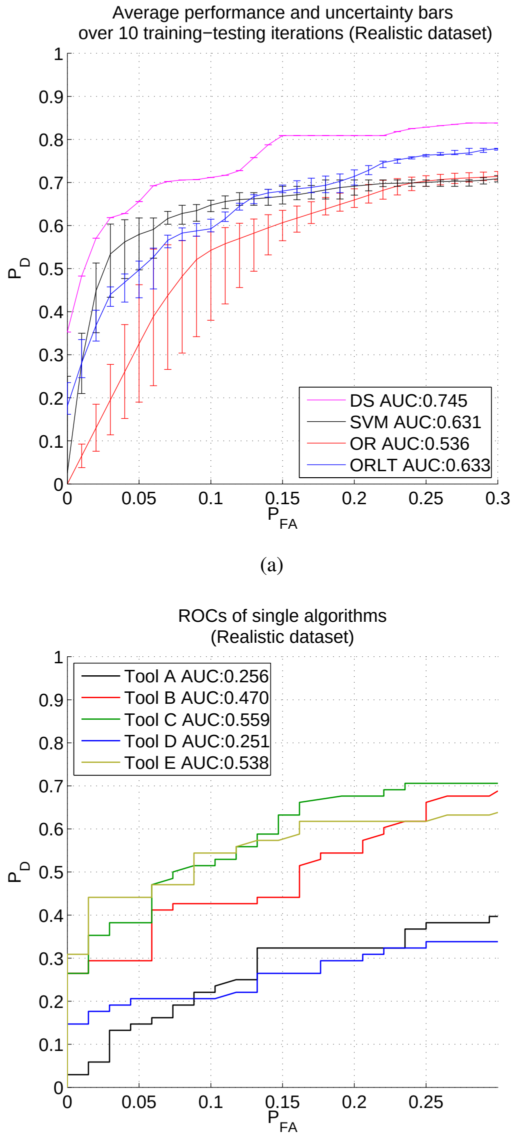
Fig. 6. Results on the synthetic test dataset both using the decision fusion methods (a) and each algorithm separately (b). For all methods involving a training phase, we plot the average performance along with uncertainty bars, showing the maximum and minimum values obtained. AUC values are normalized to the considered interval.

For this reason in the future we are going to extend our research in several directions including: considering spatial information in the fusion process, fusion of forensic tools explicitly thought to cope with counter-forensics, evaluation of the performance of the fusion process on large realistic datasets, comparison of the Dempster-Shafer framework with other reasoning approaches, including fuzzy theory, Bayesian inference, imprecise probability.

APPENDIX

In the following, the formal derivation behind the belief discounting principle is given. We assume that the reliability of *ToolA* is known through R. We introduce a new variable a, with frame: $\Omega_a = \{ra, ua\}$ where *ra* is the proposition "*ToolA* is reliable" and *ua* is the proposition "*ToolA* in unreliable". In our framework we model information about reliability by

[12] Values have been obtained running a Matlab implementation of the fusion framework on a laptop computer equipped with a Pentium Core2 Duo 2.26GHz CPU, 4GB RAM.



(a)

(b)

Fig. 7. Results on the realistic test dataset both using the decision fusion methods (a) and each algorithm separately (b). Curves and AUC values have the same meaning of those in Figure 6.

using a BBA that has only two focal elements:

$$m_A^{\Omega_a}(X) = \begin{cases} A_R & \text{for } X = \{(ra)\} \\ 1 - A_R & \text{for } X = \{(ua)\} \end{cases}$$

This BBA does not assign a mass to doubt: we are saying that knowing that a tool is unreliable and ignoring whether it is reliable or not are considered in the same way. Consequently, the most intuitive mapping from R to this BBA assignment is to choose $A_R = R$.

Being defined on different frames, $m_A^{\Theta_\alpha}$ and $m_A^{\Omega_a}$ cannot be combined directly. We need to extend them to a common domain: the simplest one is $\Theta_\alpha \times \Omega_a$, which contains propositions on both *ToolA* response and reliability. We use vacuous extension to find $m_A^{\Omega_a\uparrow\Theta_\alpha\times\Omega_a}$:

$$m_A^{\Omega_a\uparrow\Theta_\alpha\times\Omega_a} = \begin{cases} A_R & \text{for } X = \{(t\alpha, ra)\cup(n\alpha, ra)\} \\ 1-A_R & \text{for } X = \{(t\alpha, ua)\cup(n\alpha, ua)\} \end{cases} \tag{10}$$

while, for extending $m_A^{\Theta_\alpha}$ to $m_A^{\Theta_\alpha\times\Omega_a}$, we use a different approach, to give a specific interpretation of what tool reliability should mean: we assume that if a tool is unreliable, its detection should not modify beliefs when is fused. This can be easily expressed by putting all elements representing propositions in which the tool is not reliable (i.e. all $(\cdot, ua)$) elements) in every focal element of the combined BBA:

$$m_A^{\Theta_\alpha\uparrow\Theta_\alpha\times\Omega_a}(X) = \begin{cases} A_T & \text{for } X = \{(t\alpha, ra)\cup(t\alpha, ua)\cup(n\alpha, ua)\} \\ A_N & \text{for } X = \{(n\alpha, ra)\cup(t\alpha, ua)\cup(n\alpha, ua)\} \\ A_{TN} & \text{for } X = \{(t\alpha, ra)\cup(n\alpha, ra)\cup(t\alpha, ua)\cup(n\alpha, ua)\} \end{cases} \tag{11}$$

Now, using the combination rule (section II-B) we can combine BBAs in (10) and (11) to yield $m_{A_{tot}}^{\Theta_\alpha\times\Omega_a}$, which summarizes all the knowledge we have about *ToolA*:

$$m_{A_{tot}}^{\Theta_\alpha\times\Omega_a}(X) = \begin{cases} A_R{\cdot}A_T & \text{for } X = \{(t\alpha, ra)\} \\ A_R{\cdot}A_N & \text{for } X = \{(n\alpha, ra)\} \\ A_R{\cdot}A_{TN} & \text{for } X = \{(t\alpha, ra)\cup(n\alpha, ra)\} \\ 1-A_R & \text{for } X = \{(t\alpha, ua)\cup(n\alpha, ua)\} \end{cases}$$

Notice that in the above formula there is no conflict ($K = 0$). This agrees with the intuition that information about the reliability of a tool cannot be conflicting with information about its output.

If we are only interested in taking decisions on the presence or absence of traces (as it is in our case) it is useless to keep trace of variable a, because it does not bring any direct information about traces. Therefore, we choose to marginalize $m_A^{\Theta_\alpha\times\Omega_a}$ with respect to this variable, yielding:

$$m_{A_{tot}}(X)^{\Theta_\alpha\times\Omega_a\downarrow\Theta_\alpha} = \begin{cases} A_R{\cdot}A_T & \text{for } X = \{(t\alpha)\} \\ A_R{\cdot}A_N & \text{for } X = \{(n\alpha)\} \\ C_A & \text{for } X = \{(t\alpha)\cup(n\alpha)\} \end{cases} \tag{12}$$

where $C_A = (1 - A_R(A_T + A_N))$.

Acknowledgment

This work was partially supported by the REWIND Project, funded by the Future and Emerging Technologies (FET) programme within the 7FP of the EC under grant 268478, and by the European Office of Aerospace Research and Development under Grant FA8655-12-1-2138: AMULET - A multi-clue approach to image forensics.

References

[1] J. A. Redi, W. Taktak, and J.-L. Dugelay, "Digital image forensics: a booklet for beginners," *Multimedia Tools Appl.*, vol. 51, pp. 133–162, January 2011. [Online]. Available: http://dx.doi.org/10.1007/s11042-010-0620-1

[2] M. Fontani, T. Bianchi, A. De Rosa, A. Piva, and M. Barni, "A Dempster-Shafer framework for decision fusion in image forensics," in *Information Forensics and Security (WIFS), 2011 IEEE International Workshop on*, 29 2011-dec. 2 2011, pp. 1 –6.

[3] T. Bianchi, A. De Rosa, and A. Piva, "Improved DCT coefficient analysis for forgery localization in JPEG images," in *ICASSP*. IEEE, 2011, pp. 2444–2447.

[4] T. Bianchi and A. Piva, "Detection of non-aligned double JPEG compression with estimation of primary compression parameters," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, sept. 2011, pp. 1929 –1932.

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

[6] Z. C. Lin, J. F. He, X. Tang, and C. K. Tang, "Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis," *Pattern Recognition*, vol. 42, no. 11, pp. 2492–2501, Nov. 2009.

[7] H. Farid, "Exposing digital forgeries from JPEG ghosts," *IEEE T. on Information Forensics and Security*, vol. 4, no. 1, pp. 154–160, 2009.

[8] M. Kharrazi, H. T. Sencar, and N. D. Memon, "Improving steganalysis by fusion techniques: A case study with image steganography," *T. Data Hiding and Multimedia Security*, vol. 4300, pp. 123–137, 2006.

[9] Y.-F. Hsu and S.-F. Chang, "Statistical fusion of multiple cues for image tampering detection," in *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, oct. 2008, pp. 1386 –1390.

[10] G. Chetty and M. Singh, "Nonintrusive image tamper detection based on fuzzy fusion," *IJCSNS*, vol. 10, no. 9, pp. 86–90, Sep 2010.

[11] D. Hu, L. Wang, Y. Zhou, Y. Zhou, X. Jiang, and L. Ma, "D-S Evidence Theory based digital image trustworthiness evaluation model," in *Proc. of MINES 2009*, ser. MINES '09, vol. 1, 2009, pp. 85–89.

[12] S. Bayram, I. Avcibas, B. Sankur, and N. Memon, "Image manipulation detection," *J. Electronic Imaging*, vol. 15, no. 4, 2006.

[13] G. Shafer, *A Mathematical Theory of Evidence*. Princeton: Princeton University Press, 1976.

[14] P. Zhang and X. Kong, "Detecting image tampering using feature fusion," *Proc. of ARES 2009*, vol. 0, pp. 335–340, 2009.

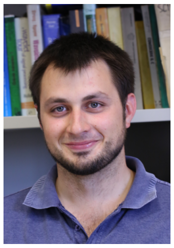
[15] Z.-W. Sun, H. Li, and Z.-C. Ji, "Fusion image steganalysis based on Dempster-Shafer evidence theory," *Control and Decision*, vol. 26, no. 8, pp. 1192–1196, 2011.

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

[17] R. Yager, "Aggregating non-independent Dempster-Shafer belief structures," in *Proc. of IPMU'08*, 2008, pp. 289–297.

[18] A. Benavoli, L. Chisci, B. Ristic, A. Farina, and A. Graziano, *Reasoning under uncertainty: from Bayesian to Valuation Based Systems*. ISBN: 978-8886658430, 2007.

[19] 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.



Marco Fontani Marco Fontani received the M.Sc. degree (Laurea) in informatics engineering from the University of Florence in 2010. Since October 2010, he is a Ph.D. student at the Department of Information Engineering of the University of Siena, following the Visual Image Processing and Protection group under the supervision of prof. M. Barni. He also collaborates with the National Inter-University Consortium for Telecommunications (CNIT), working on research projects funded by the European Commission. His research activity is mainly focused on multimedia forensics, reversible watermarking and decision fusion.

Tiziano Bianchi Tiziano Bianchi (S'03-M'05) received the M.Sc. degree (Laurea) in electronic engineering and the Ph.D. degree in information and telecommunication engineering from the University of Florence, Italy, in 2001 and 2005, respectively.

Since December 2012, he is with the Department of Electronics and Telecommunications, Politecnico di Torino as an Assistant Professor. From 2005 to 2012, he has been with the Department of Electronics and Telecommunications, University of Florence as a Research Assistant. His research interests have involved signal processing in communications, multicarrier modulation techniques, and ultra-wideband systems. Current research topics include multimedia security technologies, signal processing in the encrypted domain, and processing of SAR images. He has published more than 80 papers on international journals and conference proceedings.

Alessia De Rosa Alessia De Rosa received the electronic engineering degree and the Ph.D. degree in informatics and telecommunications from the University of Florence (Italy) in 1998 and 2002 respectively. Since 2002, she has worked at the University of Florence as a Research Assistant with the Department of Electronics and Telecommunications. Her main research interests are in the field of image processing and protection, including digital watermarking, human perception models for watermarking and quality assessment, image processing for Cultural Heritage applications and image forensics. She holds an Italian patent regarding digital watermarking.

Alessandro Piva Alessandro Piva (M'04-SM'10) received the Ph.D. degree in computer science and telecommunications engineering from the University of Florence in 1999.

From 2002 to 2004, he was a Research Scientist at the National Inter-University Consortium for Telecommunications. He is at present Assistant Professor at the University of Florence, Firenze. Italy. His current research interests are the technologies for multimedia content security, and image processing techniques for the Cultural Heritage field. He is coauthor of more than 100 papers published in international journals and conference proceedings.

Dr. Piva holds three Italian patents and an international one regarding watermarking. He serves as Associate Editor of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, of the *EURASIP Journal on Information Security*, and of the *LNCS Transactions on Data Hiding and Multimedia Security*.

Mauro Barni Mauro Barni graduated in electronic engineering at the University of Florence in 1991. He received the PhD in informatics and telecommunications in October 1995. He has carried out his research activity for over 15 years at the Department of Electronics and Telecommunication of the University of Florence, then at the Department of Information Engineering of the University of Siena where he works as associate professor. During the last decade he has been studying the application of image processing techniques to copyright protection and authentication multimedia. He is author/co-author of about 180 papers published in international journals and conference proceedings, and holds three patents in the field of digital watermarking. He was the founding editor of the EURASIP Journal on Information Security and serves as associate editor of the IEEE Trans. on CSVT, and the IEEE Trans. on Information Forensics and Security. He is a Fellow of the IEEE.