Multi-Clue Image Tampering Localization

Lorenzo Gaborini*, Paolo Bestagini†, Simone Milani†, Marco Tagliasacchi†, Stefano Tubaro†
*Dipartimento di Matematica “Francesco Brioschi”
†Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133, Milano, Italy

***Abstract*—Image tampering is nowadays at everyone's reach. This has determined an urgent need of tools capable of revealing such alterations. Unfortunately, while forgeries can be operated in many different ways, forensic tools usually focus on one specific kind of forgeries. Therefore, an effective strategy for tampering detection and localization requires to merge the output of many different forensic tools. In this paper, we propose an algorithm for image tampering localization, based on the fusion of three separate detectors: i) one based on PRNU, working when we have at least a few of pictures shot with the same camera; ii) one based on PatchMatch; iii) one exploiting image phylogeny analysis, in case we have a set of near-duplicate images to analyze. The method is validated against the dataset released by the IEEE Information Forensics and Security Technical Committee for the First Image Forensics Challenge. Results show that the proposed algorithm can beat the challenge with the highest score achieved at paper submission time.**

***Index Terms*—image forensics, tampering localization, PRNU, near duplicates, PatchMatch**

I. Introduction

Thanks to the increasing availability of cheap portable devices, such as cameras and smartphones, the acquisition of multimedia digital contents is at everyone's reach. Moreover, the development of user-friendly image editing software, enables anyone to easily alter digital images creating realistic forgered images able to fool human eyes. In fact, any content downloadable from multimedia sharing platforms (such as Flickr, YouTube, etc.) has been possibly tampered with.

The widespread diffusion of altered media has severe implications on many social and legal aspects. As an example, fake images diffused by newscasts convey false information that could manipulate the public opinion. From these premises, there is a urgent need of forensic tools that are able to uncover the history and prove the authenticity of a digital content.

In the last few years, the multimedia forensics community has developed a series of algorithms to deal with any kind of multimedia objects, and especially images [1]. Many of these algorithms rely on the fact that every non-invertible operation leaves peculiar footprints on multimedia objects. These footprints, or fingerprints, are exploited as an asset by forensic detectors enabling the identification of the applied operation.

Among the algorithms developed to uncover the history of still images, are those aiming at revealing the presence of tampering. These methods can be broadly split into two categories: i) *tampering detection* algorithms that aim to detect whether an image has been modified, or it is authentic [2]; ii) *tampering localization* algorithms that aim to detect which region of the image has been tampered with [3].

Many of the proposed algorithms focus on detecting traces left by a specific operation. As an example, [4] shows how to localize a forged region detecting inconsistencies in the Photo Response Non Uniformity (PRNU) pattern, i.e., the characteristic noise left by the acquisition sensor on images. In [5], the authors propose an algorithm capable of estimating the Color Filter Array (CFA) interpolation strategy, to detect local tampering. In [3], tampering localization is performed exploiting traces left by JPEG coding. Alternatively, [6] and [7] report methods to uncover a global resampling operation. Copy-move forgeries can be detected using [8].

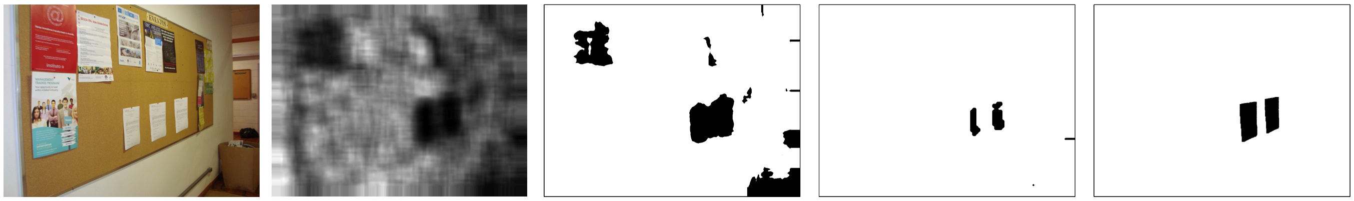
However, tampering can be operated using many different techniques together on the same picture. For this reason, an analyst cannot rely on a single detector. Instead, a more reliable way to proceed is to merge the output of different detectors [9]. This strategy was followed by [10], where the authors used a fusion technique on three different forgery localization detectors: i) one based on PRNU; ii) one based on PatchMatch for copy-move forgery detection [11]; iii) one based on a statistical method [12]. It is worth noting that the approach in [10] was validated against the dataset released by the IEEE Information Forensics and Security Technical Committee (IFS-TC) for the First Image Forensics Challenge[1], leading the authors to the win.

In this paper, we propose a multi-clue image tampering localization algorithm inspired by the work in [10]. We use a fusion strategy to merge results obtained from three detectors. The first one is a PRNU-based detector developed to be unsupervised. The second one is a PatchMatch-based detector as the one proposed in [10]. The third one is based on near-duplicate image detection, exploiting concepts borrowed from the image phylogeny research field [13].

The proposed method is validated against the IEEE IFS-TC First Image Forensics Challenge as [10]. Note that, on this dataset, techniques based on JPEG compression artifacts do not work due to the kind of operated forgeries. Experimental

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

[1] http://ifc.recod.ic.unicamp.br/



(a) image (b) PCE-field (c) binarized PCE-field (d) $\mathbf{M}_{\text{PRNU}}$ (e) ground truth

Fig. 1: A forged image (a), the PCE-field (b), the PCE-field binarized with a not optimized threshold (c), $\mathbf{M}_{\text{PRNU}}$ obtained with the proposed approach (d), and the ground truth mask (e). Notice that choosing a wrong threshold to binarize the PCE-field may lead to a bad tampering mask. Dark colors represent low values.

results show that we are able to reach a very high localization accuracy, enabling the proposed algorithm to beat the challenge with the highest score at paper submission time.

The rest of the paper is structured as follows. Section II, Section III and Section IV present the proposed image forgery localization algorithms based on PRNU, PatchMatch and near-duplicates, respectively. In Section V we show how to merge all the obtained information in order to effectively localize the tampering. Section VI reports the conducted experiments and the achieved results. Finally, in Section VII we draw some conclusive remarks.

II. PRNU-based approach

Camera sensors are greatly affected by the presence of noise. A significant non-random contribution to this noise is given by the Photo Response Non Uniformity (PRNU). Note that PRNU is pretty much stable over a camera lifetime, and is unique to each camera instance. These facts make it a robust fingerprint to identify a specific acquisition device [14].

More formally, let us consider an image $\mathbf{I}$, whose pixels are denoted as $\mathbf{I}(i, j)$, generated by a digital camera. For the sake of simplicity, let us consider $\mathbf{I}$ to be either gray scale or a single color component from a color image. This image can be modeled as

$$\mathbf{I} = \mathbf{I}_{\text{o}} + \mathbf{K}\mathbf{I}_{\text{o}} + \mathbf{\Theta}, \tag{1}$$

where $\mathbf{I}_{\text{o}}$ is the ideally acquired noise-free image, $\mathbf{\Theta}$ is an additive noise term, and the multiplicative term $\mathbf{K}$ is the PRNU.

The estimation of $\mathbf{K}$ is usually problematic, since we cannot easily separate $\mathbf{I}_{\text{o}}$ from $\mathbf{I}$. PRNU is then typically extracted from a set of images $\mathcal{I} = \{\mathbf{I}_n\}$, $n \in 1, ..., N$, acquired with the same camera. To this purpose, let us define the noise fingerprint of each image as

$$\mathbf{W}_n = \mathbf{I}_n - \mathcal{D}(\mathbf{I}_n), \tag{2}$$

where $\mathcal{D}$ is a denoising operator [15]. The PRNU is estimated as

$$\mathbf{K} = \frac{\sum_{n=1}^{N} \mathbf{W}_n \mathbf{I}_n}{\sum_{n=1}^{N} \mathbf{I}_n^2}. \tag{3}$$

To detect whether an image $\mathbf{I}_n$ has been acquired with a specific camera, a correlation test is performed between $\mathbf{K}$ and $\mathbf{W}_n$. As an example, in [16] this is done by computing a measure called Peak to Correlation Energy ratio (PCE) and comparing it to a threshold. Images whose PCE is higher than the threshold are detected as acquired with the camera associated to $\mathbf{K}$.

However, PRNU can be also used for tampering localization. To this purpose, a typical approach is to block-wise perform the correlation test between $\mathbf{K}$ and $\mathbf{W}_n$, and populate a PCE-field. This field is thresholded, and pixels whose PCE is lower than the threshold are considered fake. Many algorithms are based on slight modification of this pipeline [4], [10], [16].

The method we propose is based on the same rationale (i.e., block-wise PRNU compatibility check), but we also exploit information given by blocks correlation offsets.

Let us consider an image $\mathbf{I}$, and the PRNU $\mathbf{K}$ that we associate to the same camera. We assume the image and the PRNU to be pixel-wise aligned, i.e., no scaling or cropping operations have been applied. The image $\mathbf{I}$ is split into overlapping blocks $\mathbf{I}^b$, $b \in \{1, ..., B\}$, each one centered on pixel coordinates (i_b, j_b). If the image is pristine, each block $\mathbf{I}^b$ must pass the PRNU correlation test only when centered on PRNU pixel in coordinates (i_b, j_b).

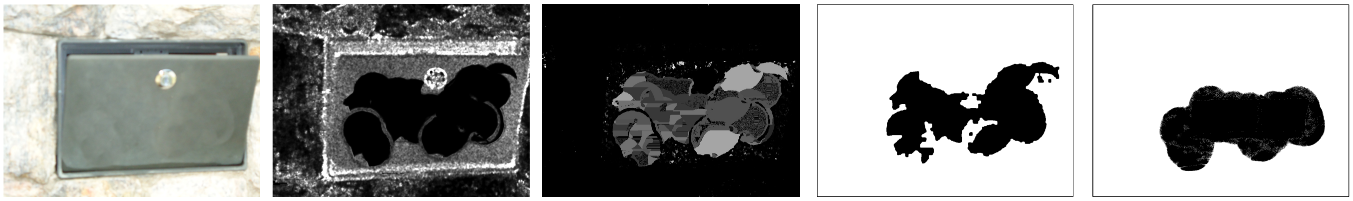
To verify this condition, we compute for each block the noise fingerprint $\mathbf{W}^b$ according to (2). We then compute the phase-correlation with the PRNU as

$$\mathbf{R}_b(i, j) = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(\mathbf{W}^b)\mathcal{F}(\mathbf{IK})^*}{|\mathcal{F}(\mathbf{W}^b)\mathcal{F}(\mathbf{IK})^*|}\right), \tag{4}$$

where $\mathcal{F}$ is the Discrete Fourier transform (zero-padded if needed), $\mathcal{F}^{-1}$ is its inverse, * denotes the complex conjugate, and the PRNU $\mathbf{K}$ is multiplied by the image pixels $\mathbf{I}$ as suggested in [17]. Notice that $\mathbf{R}_b$ is a 2D map, showing the correlation between the PRNU and the b-th block shifted of i and j pixels in the horizontal and vertical dimensions, respectively. Therefore, we can compute the offset estimate between $\mathbf{K}$ and $\mathbf{I}^b$ as

$$(\hat{i}_b, \hat{j}_b) = \underset{(i,j)}{\operatorname{argmax}} \left(\mathbf{R}_b(i, j)\right). \tag{5}$$

If $(\hat{i}_b, \hat{j}_b) = (i_b, j_b)$, the b-th block is compatible with the underlying PRNU, hence the block is considered pristine. On the other hand, if $(\hat{i}_b, \hat{j}_b) \neq (i_b, j_b)$, the b-th block is not aligned with the PRNU, thus it is considered tampered with. This condition is usually not considered in baseline PRNU-based algorithms, which are based on thresholding PCE (or



(a) image (b) $\mathbf{E}$ (c) $\mathbf{D}$ (d) $\mathbf{M}_{\text{PM}}$ (e) ground truth

Fig. 2: A forged image (a), $\mathbf{E}$ (b), $\mathbf{D}$ (c), $\mathbf{M}_{\text{PM}}$ obtained with the proposed approach (d) and the ground truth mask (e). Dark colors represent low values.

correlation) values. The proposed PRNU tampering mask is then built as

$$\mathbf{M}_{\text{PRNU}} = \begin{cases} 1, & \text{if} \quad (\hat{i}_b, \hat{j}_b) = (i_b, j_b), \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where 1 denotes a pristine pixel, and 0 a forged one.

Fig. 1 shows the comparison between $\mathbf{M}_{\text{PRNU}}$ and the PCE-field used by many methods [4], [10], [16]. Notice that the proposed approach has an inherent advantage over PCE-based approaches. This is, $\mathbf{M}_{\text{PRNU}}$ does not need any training, since no thresholds must be defined. Opposite, PCE-based approaches need some training images to chose a good threshold to compute a binary mask from the PCE-field. A wrong threshold choice may severely degrade tampering localization accuracy.

III. PatchMatch-based approach

A common way to perform an attack is to clone part of an image over another region of the same image. This can be either done with simple copy-move techniques,i.e., replicating an object or small patches from the same image with some optional additional filtering to minimize the discontinuity between original and copied pixels.[2] If the copied object or patch is big enough, SIFT-based copy-move forgery detectors [8] achieve good results. On the other hand, if cloned patches are very small, SIFT-based methods are not effective. In order to be able to detect both small and big replicated regions, we rely on the PatchMatch algorithm presented in [11] and used also in [10]. PatchMatch enables to detect whether a small patch (e.g., a 7×7 pixels block) can be replaced with another small patch found in the same image, at a very low computational complexity.

More formally, an image $\mathbf{I}$ is split into non overlapping 7×7 pixels blocks $\mathbf{I}^b$, $b \in \{1, ..., B\}$, each one centered on pixel coordinates (i_b, j_b). For each block $\mathbf{I}^b$, PatchMatch returns the block that is most similar to it as

$$\hat{\mathbf{I}}^b = \underset{\mathbf{I}^\beta \in \mathcal{B}}{\operatorname{argmin}} \, \mathcal{E}\left(\mathbf{I}^b, \mathbf{I}^\beta\right), \tag{7}$$

where $\mathcal{E}$ is a certain distance metric (mean squared error in our experiments), and $\mathcal{B}$ is a set of possible $\hat{\mathbf{I}}^b$ candidates selected by PatchMatch to avoid full-search and patches too close to $\mathbf{I}^b$.

We then store the information about the matching patches into two matrices: i) $\mathbf{D}$ is a map of the distances between the centers of each matching pair $\langle \mathbf{I}^b, \hat{\mathbf{I}}^b \rangle$; ii) $\mathbf{E}$ is a map of the Mean Squared Error (MSE) introduced if we actually substitute a patch with its matching one. These maps are built as

$$\mathbf{D}(i_b, j_b) = |[i_b, j_b] - [\hat{i}_b, \hat{j}_b]|, \tag{8}$$

$$\mathbf{E}(i_b, j_b) = \text{MSE}\left(\mathbf{I}^b, \hat{\mathbf{I}}^b\right), \tag{9}$$

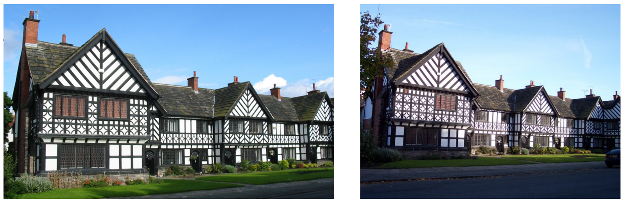
where $|\cdot|$ is the L^2-norm, $[i_b, j_b]$ and $[\hat{i}_b, \hat{j}_b]$ are vectors collecting the coordinates of $\mathbf{I}_b$ and $\hat{\mathbf{I}}_b$ central pixels respectively, and $\text{MSE}(\cdot, \cdot)$ compute the MSE between two patches. Fig. 2 shows an example of these two maps computed on a picture forged using Healing Brush [18].

To compute the binary tampering mask $\mathbf{M}_{\text{PM}}$, we segment $\mathbf{D}$ in regions with the same $\mathbf{D}$ value. These are areas that can be substituted with pixels at a fixed distance. Among these areas, we select only those larger than a given size (fixing the smallest tampered block we want to detect) and with a low $\mathbf{E}$ value. The mask $\mathbf{M}_{\text{PM}}$ can be optionally refined using morphological operators. Fig. 2 shows an example of $\mathbf{M}_{\text{PM}}$ compared to the ground truth mask. It is worth noting that $\mathbf{M}_{\text{PM}}$ suffers of ambiguity problems when copy-move attack is used, i.e., it is not possible to disambiguate between the original and copied objects. If more sophisticated attacks are used (e.g., Healing Brush), this problem is less pronounced.

IV. Near-duplicate-based approach

When dealing with user-generated content distributed online, forged objects are seldom created starting only from undistributed original material [19]. In fact, a common image tampering pipeline is to collect and reuse pictures found on different media sharing platforms. A typical example is that of image copy-paste forgery operated to substitute the face of a person (e.g., a friend of the forgery creator) with that of another (e.g., a famous artist). In Fig. 3a, this kind of attack has been used to replicate a window. It is then possible to search for near-duplicate copies of the image under analysis (i.e., versions of the same image differing due to processing operations, or pictures of the same scene captured from a slightly different point of view) and compare them to find the differences. This search can be either performed via Web crawling, or in the dataset under analysis. An example of a

[2] An example of copy-move with small patches is the Healing Brush Tool of the Photoshop software [18].



(a) dataset (b) web crawled

Fig. 3: A fake image coming from the IFS-TC dataset (a) and a near-duplicate version of it found online (b). The analysis of the differences reveals the tampering.

fake picture coming from the IEEE IFS-TC dataset and a near-duplicate version found online is shown in Fig. 3.

The idea of studying the relationship between pairs of near-duplicate images to find which one has possibly been used to generate the others is at the base of the image phylogeny research field [13], [20], [21]. Starting from this idea, we propose a near-duplicate-based image tampering localization approach.

The first step consists in determining which images are actually near-duplicates. To this purpose, let us consider a set of images to analyze. We describe each image by means of a robust hash, obtained modifying the hash proposed for near-duplicate video matching in [19]. To build the hash, we resize each image $\mathbf{I}_n$ to a fixed dimension (256×256 pixels in our experiments). We then compute $\mathbf{Y}_n$ as the 2D Discrete Cosine Transform (DCT) of the resized image. We select a given number of DCT coefficients (in our experiments 256 coefficients $\mathbf{Y}_n(i,j),\ i \in \{2,...,17\}, j \in \{2,...,17\}$ discarding horizontal and vertical components whose $i = 1$ or $j = 1$). The selected coefficients are binarized with respect to their median value to obtain the binary hash $\mathbf{h}_n$ (a 256 bit string in our case, composed by 128 zeros and 128 ones). Hashes are then pairwise compared by computing hamming distance between each pair. If this distance is below a threshold (4 in our experiments), the images related to the compared hashes are considered near-duplicates.

After we identify a near-duplicate $\mathbf{I}_m$ of an image $\mathbf{I}_n$ under analysis, we compare them pixel-wise to find the differences. To this purpose, we register $\mathbf{I}_m$ to $\mathbf{I}_n$ in order to compensate for geometrical transformations such as cropping and resizing. This is done using SIFT matching as suggested in [13]. Then we subtract $\mathbf{I}_n$ to the registered version of $\mathbf{I}_m$ obtaining a difference map. Notice that differences between $\mathbf{I}_n$ and $\mathbf{I}_m$ are due to the presence of tampering, the presence of noise introduced by processing operations such as JPEG compression, and errors introduced in the registration step. For this reason, the estimation of the binary tampering localization mask $\mathbf{M}_{\text{ND}}$ requires thresholding the difference map and optionally processing it with some morphological operators. Fig. 4 shows an example of near-duplicate images, and the obtained $\mathbf{M}_{\text{ND}}$ mask. Note that, if both the compared images contain tampered areas, $\mathbf{M}_{\text{ND}}$ suffers of ambiguity problems as $\mathbf{M}_{\text{PM}}$. If we find K near-duplicates of a reference image, we obtain a set $\mathbf{M}_{\text{ND}}^k,\ k \in \{1,...,K\}$ of masks (i.e., one for each near-duplicate) that can be merged to solve the ambiguity.

V. Fusion

For each image $\mathbf{I}_n$ we can estimate a set of different tampering masks (i.e., $\mathbf{M}_{\text{PRNU}}$, $\mathbf{M}_{\text{PM}}$ and $\mathbf{M}_{\text{ND}}^k,\ k \in \{1,...,K\}$), each one set to zero to denote forged pixels, and to one to denote pristine pixels. In order to reach a final decision, we must merge these masks into a single one $\mathbf{M}_{\text{FUS}}$. To this purpose, let us take into account the inherent properties of each kind of masks.

$\mathbf{M}_{\text{PRNU}}$ reveals many kind of tampering, and revealed forged areas are unambiguous. In other words, we can strongly trust areas detected as tampered with. On the other hand, $\mathbf{M}_{\text{PRNU}}$ hardly reveals forgeries smaller than the block size used to analyze $\mathbf{I}_n$ during the PRNU analysis. For this reason, some forged areas may not be revealed by $\mathbf{M}_{\text{PRNU}}$.

$\mathbf{M}_{\text{PM}}$ is tailored to a specific kind of attack (i.e., copy-move-like), moreover it presents ambiguous regions. The ambiguity is due to the fact that both the original and copied patches are detected (e.g., if an object is replicated, both replicas appear in $\mathbf{M}_{\text{PM}}$).

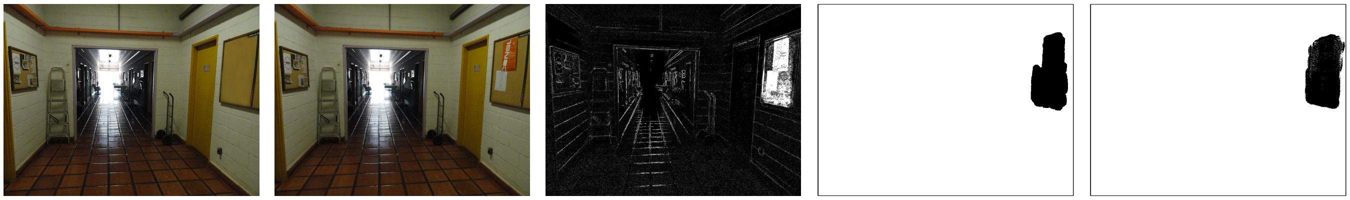
$\mathbf{M}_{\text{ND}}^k$ reveals many kind of forgeries (as $\mathbf{M}_{\text{PRNU}}$), nonetheless it suffers from ambiguity problems (as $\mathbf{M}_{\text{PM}}$). Indeed, each $\mathbf{M}_{\text{ND}}^k$ contains information about forgeries on both compared images (i.e., $\mathbf{I}_n$ and its near-duplicate).

Since each mask embeds information that might not be present into the others, a natural method to merge them is the use of the AND operator [10]. However, due to ambiguity in certain masks, this is a suboptimal choice. The pipeline we propose for mask fusion is then the following: i) we solve the ambiguity problem where possible (i.e., for $\mathbf{M}_{\text{ND}}$); ii) we select which masks (i.e., $\mathbf{M}_{\text{PRNU}}$, $\mathbf{M}_{\text{PM}}$ and $\mathbf{M}_{\text{ND}}$) to merge with the AND operator, according to a confidence value obtained evaluating the masks on a training set.

Ambiguity. When dealing with masks $\mathbf{M}_{\text{ND}}^k,\ k \in \{1,...,K\}$, the ambiguity problem can be solved. Indeed, if $K > 1$, we can resolve this ambiguity by comparing all the $\mathbf{M}_{\text{ND}}^k$ masks. Each mask reveals the forged area in both $\mathbf{I}_n$ and a near-duplicate version of it, therefore the only forged region that appears in every mask is attributed to $\mathbf{I}_n$. More formally, we compute the binary mask $\mathbf{M}_{\text{ND}} = \mathbf{M}_{\text{ND}}^1 \vee \mathbf{M}_{\text{ND}}^2 \vee ... \vee \mathbf{M}_{\text{ND}}^K$, where $\vee$ is the OR operator.

Mask selection. We define a set $\mathcal{M}$ of possible masks obtained according to different strategies. In our experiments $\mathcal{M} = \{\{\mathbf{M}_{\text{PRNU}}\}, \{\mathbf{M}_{\text{PM}}\}, \{\mathbf{M}_{\text{ND}}\}, \{\mathbf{M}_{\text{PRNU}} \wedge \mathbf{M}_{\text{PM}}\}, \{\mathbf{M}_{\text{PRNU}} \wedge \mathbf{M}_{\text{ND}}\}, \{\mathbf{M}_{\text{PM}} \wedge \mathbf{M}_{\text{ND}}\}, \{\mathbf{M}_{\text{PRNU}} \wedge \mathbf{M}_{\text{PM}} \wedge \mathbf{M}_{\text{ND}}\}\} = \{\mathbf{M}^p\},\ p \in \{1,...,7\}$, where $\wedge$ is the AND operator. Notice that different sets can be defined as well.

We consider a training set $\mathcal{S}_{\text{train}}$ of forged images $\mathbf{I}_n,\ n \in \{1,...,N\}$, whose ground truth tampering mask $\mathbf{M}_n^{\text{GT}}$ is known. For each image $\mathbf{I}_n \in \mathcal{S}_{\text{train}}$, we compute the set of possible masks $\mathcal{M}_n = \{\mathbf{M}_n^p\}$. Notice that some strategies cannot be applied on some images (e.g., in case of unknown



(a) reference image (b) near-duplicate (c) difference (d) $\mathbf{M}_{\text{ND}}$ (e) ground truth

Fig. 4: Reference image $\mathbf{I}_n$ (a), near-duplicate $\mathbf{I}_m$ (b), difference between reference and registered near-duplicate (c), $\mathbf{M}_{\text{ND}}$ (d), and ground truth mask. Dark colors represent low values.

PRNU), thus resulting in a different cardinality of $\mathcal{M}_n$ for each image.

For each image $\mathbf{I}_n$, we select the strategy that gives the best estimated mask as

$$p_n = \underset{p}{\operatorname{argmax}}\, \mathcal{R}\left(\mathbf{M}_n^p, \mathbf{M}_n^{\text{GT}}\right), \tag{10}$$

where $\mathcal{R}(\cdot,\cdot)$ is a metric of similarity between the compared masks (in our experiments we used the F-measure score adopted for the Image Forensics Challenge described in Section VI). We then compute a confidence value for each strategy as

$$C_p = \frac{|\{\mathbf{I}_n \in \mathcal{S}_{\text{train}} : p_n = p\}|}{|\{\mathbf{I}_n \in \mathcal{S}_{\text{train}} : \exists\, \mathbf{M}_n^p\}|},\ p \in \{1, ..., 7\}, \tag{11}$$

where the numerator represents the number of images for which the p-th strategy is the best one, and the denominator is the number of images on which the p-th strategy could actually be applied.

When a new image $\mathbf{I}_m \notin \mathcal{S}_{\text{train}}$ is evaluated, we use the strategy

$$\hat{p} = \underset{p}{\operatorname{argmax}}\, \left(C_p \mid \exists\, \mathbf{M}_m^p\right), \tag{12}$$

which is the strategy associated to the mask $\mathbf{M}_m^{\hat{p}}$ (among those that can be computed for $\mathbf{I}_m$) that maximizes the confidence $C_{\hat{p}}$. We refer to the mask obtained with the fusion procedure as $\mathbf{M}_{\text{FUS}}$.

VI. Experimental Results

In order to validate the proposed approach, we tested it on the dataset distributed for the IEEE IFS-TC First Image Forensics Challenge. This dataset is composed by four sets of images: i) $\mathcal{S}_{\text{train}}^{\text{f}}$ composed by 450 tampered images whose ground truth mask is known; ii) $\mathcal{S}_{\text{train}}^{\text{p}}$ composed by 1050 pristine images; iii) $\mathcal{S}_{\text{test}}^{1}$ composed by 5713 images, both fake and pristine, without annotation; iv) $\mathcal{S}_{\text{test}}^{2}$ composed by 700 fake images without annotation, actually used to officially test tampering localization methods submitted to the challenge. The total number of images in $\mathcal{S} = \{\mathcal{S}_{\text{train}}^{\text{f}} \cup \mathcal{S}_{\text{train}}^{\text{p}} \cup \mathcal{S}_{\text{test}}^{1} \cup \mathcal{S}_{\text{test}}^{2}\}$ is 7913, of which 1500 annotated (i.e., we know whether they are pristine or fake, and the tampering masks are known too). A large amount of images is 1024×768 pixels each, nonetheless other image sizes (either smaller or bigger) are present. Notice that we present results for $\mathcal{S}_{\text{train}}^{\text{f}}$ (that can be evaluated thanks to the available ground truth) and $\mathcal{S}_{\text{test}}^{2}$ (that can be evaluated submitting masks to the challenge system, which only returns one F-measure score a day). The other datasets are used to search for near-duplicates and PRNU estimation.

TABLE I: Comparison between PRNU-based forgery localization methods.

	TPR	TNR	FPR	FNR	ACC
$\mathbf{M}_{\text{PCE}}$ (supervised)	61.36%	**82.51%**	**17.49%**	38.64%	71.93%
$\mathbf{M}_{\text{PRNU}}$	68.44%	74.39%	25.61%	31.56%	71.41%
$\mathbf{M}_{\text{PRNU}}$ (supervised)	**82.82%**	68.45%	31.55%	**17.18%**	**75.63%**

In order to compute $\mathbf{M}_{\text{PRNU}}$, we first needed the reference PRNU for each image. Considering a specific image $\mathbf{I}_n$, we selected all the images $\mathbf{I}_m \in \mathcal{S}$ of the same size of $\mathbf{I}_n$. We computed the noise fingerprint $\mathbf{W}_m$ for each one of them using (2). We computed the PCE between every pair of fingerprints $\langle \mathbf{W}_n, \mathbf{W}_m \rangle$, keeping n fixed. We selected all the images $\mathbf{I}_m$ with PCE greater than 50 and fingerprint $\mathbf{W}_m$ aligned with $\mathbf{W}_n$ as images taken from the same camera used for $\mathbf{I}_n$. Then we computed the PRNU $\mathbf{K}_n$ related to the n-th image using (3), considering all the images $\mathbf{I}_m$ associated to $\mathbf{I}_n$.

With this procedure, we identified the PRNU for the 45% of images in $\mathcal{S}_{\text{train}}^{\text{f}}$ and more than the 54% in $\mathcal{S}_{\text{test}}^{2}$, just using 1024×768 images. For these images, we computed both $\mathbf{M}_{\text{PRNU}}$ using the proposed method, and a PRNU-based tampering mask $\mathbf{M}_{\text{PCE}}$ obtained by thresholding the PCE-field as suggested in [10]. We evaluated both techniques in terms of: i) *True Positive Rate* (TPR) as the fraction of fake pixels correctly identified; ii) *True Negative Rate* (TNR) as the fraction of pristine pixels correctly identified; iii) *False Positive Rate* (FPR) as the fraction of pristine pixels identified as fake; iv) *False Negative Rate* (FNR) as the fraction of fake pixels identified as pristine; v) *Accuracy* (ACC) as $(\text{TPR} + \text{TNR})/2$.

This evaluation was carried on only on $\mathcal{S}_{\text{train}}^{\text{f}}$, since we only have the ground truth for this set, and the submission system (working on $\mathcal{S}_{\text{test}}^{2}$) does not provide such statistics. Notice that the PCE threshold for $\mathbf{M}_{\text{PCE}}$ was chosen as the optimal one for the used dataset, therefore reported results can be considered as an upper bound of $\mathbf{M}_{\text{PCE}}$ performance. On the other hand, $\mathbf{M}_{\text{PRNU}}$ is completely unsupervised. In order to give an upper bound to $\mathbf{M}_{\text{PRNU}}$ results too, we also used a supervised version, whereby we optimized on $\mathcal{S}_{\text{train}}^{\text{f}}$ the use of morphological opening. Table I shows results obtained comparing these strategies. Notice that, in terms of TPR, both supervised and unsupervised versions of $\mathbf{M}_{\text{PRNU}}$ beats $\mathbf{M}_{\text{PCE}}$.

TABLE II: Percentage of computed tampering masks.

	$\mathcal{S}^{\text{f}}_{\text{train}}$	$\mathcal{S}^{2}_{\text{test}}$
$\mathbf{M}_{\text{PRNU}}$	45%	54%
$\mathbf{M}_{\text{PM}}$	45%	48%
$\mathbf{M}_{\text{ND}}$	22%	29%
$\mathbf{M}_{\text{FUS}}$	**64%**	**66%**

TABLE III: F score for the obtained masks. Scores are presented for the whole dataset (*global*) and averaging it only on the number of effectively computed masks (*per image*).

	$\mathcal{S}^{\text{f}}_{\text{train}}$		$\mathcal{S}^{2}_{\text{test}}$	
	per image	*global*	*per image*	*global*
$\mathbf{M}_{\text{PRNU}}$	45.14%	24.83%	33.17%	25.35%
$\mathbf{M}_{\text{PM}}$	45.41%	25.63%	41.90%	27.84%
$\mathbf{M}_{\text{ND}}$	76.03%	26.79%	84.48%	33.31%
$\mathbf{M}_{\text{FUS}}$	55.52%	38.53%	56.69%	**45.33%**

In terms of accuracy, supervised $\mathbf{M}_{\text{PRNU}}$ achieves better results than the other techniques.

After validating the proposed PRNU-based approach, we focused on the challenge evaluation. Table II shows how many masks we obtained with each method on each dataset. Notice that after fusion, we have a mask estimate for the 64% and 66% of images in $\mathcal{S}^{\text{f}}_{\text{train}}$ and $\mathcal{S}^{2}_{\text{test}}$, respectively.

According to the challenge rules, tampering localization methods are evaluated according to F-measure defined as

$$F = \frac{2\ \text{TPR}}{2\ \text{TPR} + \text{FNR} + \text{FPR}}, \qquad (13)$$

which is zero if evaluated masks are all white (i.e., every pixel is detected as pristine). This fact allowed us to evaluate the masks in two different scenarios: i) *per image* - computing the score only for images we could actually obtain a mask for (see Table II); ii) *global* - computing the score on all the images in each set, setting as black the masks not available. The first criterion is an estimate of masks' goodness. The second one is obtained as the official Challenge score, taking into account the fact we could not estimate a mask for each image. Table III shows these results. *Global* results are higher on $\mathcal{S}^{2}_{\text{test}}$ than on $\mathcal{S}^{\text{f}}_{\text{train}}$ because we actually estimated more masks on $\mathcal{S}^{2}_{\text{test}}$ than on $\mathcal{S}^{\text{f}}_{\text{train}}$ (see Table II).

Notice that results in the last column (highlighted in red) were obtained using the official Challenge submission system. The $\mathbf{M}_{\text{FUS}}$ score of **45.33%** (publicly obtained as Gabor team[3]) is higher than that obtained by the official challenge winners (i.e., 40.72%), fully validating our approach.

VII. Conclusions

In this paper we presented a multi-clue image tampering localization algorithm that merges information from three different strategies. The proposed approach builds upon [10], introducing an alternative PRNU-based detector, detailing the PatchMatch-based one, and presenting a tool based on image phylogeny. To the best of our knowledge, results on the IEEE IFS-TC First Image Forensics Challenge dataset achieve the highest score (at paper submission time). Future works will be devoted to fine-tune the fusion strategy, as well as to integrate tools based on other footprints.

[3] http://tinyurl.com/onandky or http://tinyurl.com/nvc8dn4

References

[1] A. Piva, "An overview on image forensics," *ISRN Signal Processing*, vol. 2013, p. 22, 2013.

[2] G. K. Birajdar and V. H. Mankar, "Digital image forgery detection using passive techniques: A survey," *Digital Investigation*, vol. 10, pp. 226–245, 2013.

[3] T. Bianchi and A. Piva, "Image forgery localization via block-grained analysis of JPEG artifacts," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 7, pp. 1003–1017, 2012.

[4] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 3, pp. 74–90, 2008.

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

[6] M. Kirchner, "Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue," in *ACM workshop on Multimedia and Security (MM&Sec)*, 2008.

[7] D. Vazquez-Padin and P. Comesana, "ML estimation of the resampling factor," in *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2012.

[8] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "A SIFT-based forensic method for copy-move attack detection and transformation recovery," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 6, pp. 1099–1110, 2011.

[9] M. Fontani, T. Bianchi, A. De Rosa, A. Piva, and M. Barni, "A framework for decision fusion in image forensics based on dempster-shafer theory of evidence," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 8, pp. 593–607, 2013.

[10] D. Cozzolino, D. Gragnaniello, and L. Verdoliva, "A novel framework for image forgery localization," in *IEEE IFS-TC Image Forensics Challenge (phase-2)*, 2013, (presented at WIFS'13). [Online]. Available: http://arxiv.org/abs/1311.6932

[11] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized PatchMatch correspondence algorithm," in *2010 European Conference on Computer Vision (ECCV)*, 2010.

[12] D. Cozzolino, D. Gragnaniello, and L. Verdoliva, "Image forgery detection based on the fusion of machine learning and block-matching methods," in *IEEE IFS-TC Image Forensics Challenge (phase-1)*, 2013, (presented at WIFS'13). [Online]. Available: http://arxiv.org/abs/1311.6934

[13] Z. Dias, A. Rocha, and S. Goldenstein, "Image phylogeny by minimal spanning trees," *IEEE Transactions on Information Forensics and Security*, vol. 7, pp. 774–788, 2012.

[14] M. Goljan, J. Fridrich, and T. Filler, "Large scale test of sensor fingerprint camera identification," in *SPIE Confernece on Media Forensics and Security (MFS)*, 2009.

[15] G. Chierchia, S. Parrilli, G. Poggi, C. Sansone, and L. Verdoliva, "On the influence of denoising in PRNU based forgery detection," in *ACM Workshop on Multimedia in Forensics, Security and Intelligence (MiFor)*, 2010.

[16] M. Goljan, "Digital camera identification from images estimating false acceptance probability," in *Digital Watermarking*. Springer Berlin Heidelberg, 2009.

[17] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Source digital camcorder identification using sensor photo-response nonuniformity," in *SPIE Confernece on Electronic Imaging*, 2007.

[18] T. Georgiev. (2014, Jun.) Photoshop healing brush: a tool for seamless cloning. [Online]. Available: http://www.tgeorgiev.net/Photoshop_Healing.pdf

[19] S. Lameri, P. Bestagini, A. Melloni, S. Milani, A. Rocha, M. Tagliasacchi, and S. Tubaro, "Who is my parent? reconstructing video sequences from partially matching shots," in *IEEE International Conference on Image Processing (ICIP)*, 2014.

[20] A. De Rosa, F. Uccheddu, A. Piva, M. Barni, and A. Costanzo, "Exploring image dependencies: A new challenge in image forensics," in *SPIE Conference on Media Forensics and Security (MFS)*, 2010.

[21] Z. Dias, A. Rocha, and S. Goldenstein, "First steps toward image phylogeny," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2010.