HIDING TRACES OF MEDIAN FILTERING IN DIGITAL IMAGES

M. Fontani, M. Barni

Dept. of Information Engineering - University of Siena
via Roma 56, Siena (IT)

ABSTRACT

Detection of median filtering is an important task in image forensics, since this operator is frequently used both for benign and malicious processing. In this paper we introduce a counter-forensic technique that allows to conceal traces left by median filtering while preserving the quality of the processed image. The work aims to hide traces searched by state-of-the-art tools, and does not require JPEG compression of the image to hide traces.

Index Terms— Image forensic, counter-forensic, anti-forensic, median filtering detection.

1. INTRODUCTION

In the last years a great effort has been put into the development of digital image forensics techniques, which allow to investigate the origin and integrity of a given image in a blind fashion. These techniques are usually based on the assumption that the acquisition and editing processes leave some (usually invisible) fingerprints into the image, that can be leveraged to expose tampering or to infer some information about the originating device. In its beginnings, image forensics research was mainly focused on detecting "malicious" editing undergone by images: this includes cut&paste attacks (where two or more images are spliced), copy-move attacks (where a portion of the image is replicated to another location), and so on. More recently, however, also the detection of benign editing (such denoising filtering, rotating, resizing) has come to interest, since these processes affect the history of the data as well. Furthermore, benign filtering can be used after a malicious processing in order to conceal the traces introduced during the first step.

Along with research on image forensic methods, counter-forensics has emerged as the dual discipline: its goal is to devise processing techniques that allow to edit a digital image without leaving in it those fingerprints that would be revealed by a forensic algorithm (see [1] for a thorough definition). In this work we propose a generalizable counter-forensic approach that, leveraging on the knowledge of the to-be-countered forensic technique, moves the processed image to a similar (i.e. constrained to have a low distortion) one that no longer exposes the fingerprint searched by the detector. We focus on the case of median filtering footprints concealment. Median filtering has an overwhelming importance in image processing: it is known to be a good denoising filter (excellent for salt-and-pepper noise removal) as well as a smoothing operator that does not introduce new values in the signal. To the best of our knowledge, three methods have been proposed in the literature for detecting median filtering, that will be briefly described in the following. Among these, the most recent one [2] yields the best performance. The paper is structured as follows: Section 2 introduces the currently available median filtering detection tools [2] [3] [4]; Section 3 defines the proposed counter-forensic method and finally in Section 4 we validate our approach. We focus on removing footprints defined in [2], and then show that this attack also hinders the performance of the detectors in [3] and [4].

This work was partially supported by the REWIND project funded by the Future and Emerging Technologies (FET) programme within the 7FP of the European Commission, under FET-Open grant number: 268478.

2. DETECTION OF MEDIAN FILTERING

Median filtering detection is considered an important task in image forensics, but is also known to be a hard problem because of the non-linear nature of the median operator, which limits the usefulness of common statistical tools. Nevertheless, some characteristic footprints left by median filtering have been studied and exploited in the literature. We briefly describe each one of the proposed techniques, assuming that the analyzed image is grayscale and has not been JPEG compressed after median filtering (this would facilitate footprints concealment).

2.1. Kirchner et al. method

Kirchner et al [4] propose a simple yet effective detector based on the "streaking artifact" that characterizes median filtered images: the basic idea is that the probability of two adjacent pixels being equal is greatly increased by median filtering. The authors therefore compute the difference D between the image and a 1-pixel shifted version of it, and calculate the histogram h_D of this difference. It turns out that, in median filtered images, the ratio between the bin centered in 0, h_D^0 and the adjacent ones, h_D^1, h_D^{-1} is far higher than it

is in natural images. Therefore, the value

$$\varrho = h_D^0 / h_D^1$$

that is expected to be approximatevely 1 for natural images, will be much greater for filtered ones and can be considered a good discriminating feature. Since classification based on ϱ is less reliable on highly saturated images, this feature is evaluated block-wise, compensating for this effect.

2.2. Cao et al. method

A similar consideration leads the work by Cao et al [3]. The authors observe that, in presence of median filtering, it is much more likely that the difference between two adjacent pixel is exactly zero. To explore the presence of this footprint, they compute and binarize the row-based first order difference ΔI_r as follows:

$$\Delta I_r(i,j) = \begin{cases} 1 & \text{if } I(i+1,j) - I(i,j) = 0 \\ 0 & \text{if } I(i+1,j) - I(i,j) \neq 0 \end{cases} \quad (1)$$

where I is the image under analysis. In the same way, they also compute column-based difference $\Delta I_c(i,j)$ for each pixel. Obviously highly textured regions will rarely show equal adjacent pixels, independently of median filtering, and this must be taken into account: a map $V(i,j)$ is computed evaluating for each pixel the variance of the surrounding region. Using the first order difference and the variance map, the actual features of the scheme are computed as:

$$f_r = \frac{\sum_{i,j} \Delta I_r(i,j) \cdot V(i,j)}{\sum_{i,j} V(i,j)}$$

the same is done for column differences (substituing $\Delta I_c(i,j)$ to $\Delta I_r(i,j)$ in 2.2) yielding f_c. The final scalar feature ρ is obtained as $\rho = [f_r, f_c] \bullet [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$, where $\bullet$ denotes the dot product.

2.3. Yuan method

The last, more recent, approach from Yuan [2] is far more elaborate. Since median is an order operator, its block-wise application to images will obviously affect the ordering of pixels within each block. Furthermore, since the filter is applied to overlapping blocks, some kind of dependencies between neighboring blocks is expected: the authors state that this local dependence is an artifact that characterizes median filtering. To account for these two footprints, a set of five features is extracted from $s \times s$ non overlapping blocks; we limit ourself to give an informal description of each of them, since the formal definition would be notationally heavy. All features compute a value for every pixel in the block (so each block is represented by five $1 \times (s \times s)$ arrays) and their average value among all blocks is considered.

- Distribution of block median (DBM), denoted with $\mathbf{h}^{DBM}$: accounts for the fact that, in median-filtered images, gray levels in a small block tend to be equal to the block median;
- Occurrence of the Block-Center Gray Level (OBC), denoted with $\mathbf{h}^{OBC}$: accounts for the fact that the gray level of the block center should occur more frequently in the block after median filtering;
- Quantity of Gray Levels in a Block (QGL), denoted with $\mathbf{h}^{QGL}$: since the median filter reduces noise without introducing new gray levels, it is likely that after filtering the number of different gray levels in each block is decreased.

While above features do not consider the sorting of gray levels in the block, the following take it into account:

- Distribution of the Block-Center Gray Level in the Sorted Gray Levels (DBC), denoted with $\mathbf{h}^{DBC}$: considers the frequency of the block-center gray level in the *sorted* gray levels;
- First Occurrence of the Block-Center Gray Level in the Sorted Gray Levels (FBC), denoted with $\mathbf{h}^{FBC}$: simply considers the first occurrence of the block-center gray level in sorted gray level.

Having defined these features, Yuan proposes an effective way to merge them together into a single scalar value f, and this fusion aims to best exploit the discriminant properties of each of the five features:

$$f = \frac{h_5^{DBM} h_2^{OBC} h_6^{QGL} (h_3^{DBC} + h_7^{DBC} - h_2^{DBC} - h_8^{DBC}) h_3^{FBC}}{h_1^{OBC} h_9^{QGL} (h_2^{DBC} + h_8^{DBC} - h_1^{DBC} - h_9^{DBC}) h_2^{FBC} h_9^{FBC}} \quad (2)$$

Experimental evidence shows that values for f near to the unit are typical of non-filtered images, while median filtered ones yield values greater by three order of magnitude. Median filtering detection is then obtained by simply thresholding f.

3. COUNTER-FORENSIC FOR MEDIAN FILTERING

In this section we present the proposed counter-forensic technique, that aims at removing traces left by median filtering while preserving a high fidelity of the image. We believe that the forensic detection proposed by Yuan is more elaborate then the previously existing detection methods: for this reason, we initially focus in removing traces searched in [2], and then show that this also hinders the performances of tools in [3] and [4].

3.1. Problem formulation

The basic idea of our counter-forensic technique is to *automatically* search a processing operation p (among a class P) that, starting from a median filtered image (which we will call "processed"), produces another image that is similar to

the processed one while not showing characteristic traces of median filtering. This can be thought of as an optimization problem, where we want to maximize the fidelity between the processed image and the counter-processed one while removing footprints searched by the forensic detector.

Formally, we start from a median filtered image M, from which the algorithm by Yuan extracts the feature f_M (computed as in eq. 2). Therefore, we define a cost function $c : \mathbb{R} \to \mathbb{R}$ that maps each value of the feature f to a cost, which grows as f increases. Then, given M, we are looking for a processing $p \in \mathrm{P}$ that produces an image $W = p(M)$ whose extracted feature f_W has a cost $c(f_W)$ as low as possible, while introducing as low distortion between M and W as possible. We choose PSNR as a measure of similarity, and define the following optimization problem:

$$\min_{p} \left[\mathrm{c}(f_W) - \mathrm{PSNR}(M, W)\right] \tag{3}$$

$$\text{subject to}$$

$$p \in P, W = p(M)$$

We limit the class P of admissible processing to linear convolution filters of size 3×3, without any constraint on their components. Notice that, in another formulation, we may set a constraint on the distortion induced by filtering, forcing it to be under a threshold, instead of embedding this measure into the objective function. We choose the solution in eq. (3) because it significantly simplifies the optimization problem; the attacker will check a-posteriori if the obtained filter yields an image that satisfies the quality requirements and, if it does not, he may run another optimization choosing a different starting point.

Instead of weighting the two components of the objective function with scalars, we directly choose the function c in such a way that a good tradeoff between footprint concealment and quality is retained; since very small displacements of f from typical values for unfiltered images allow the detector to discriminate well, we need a cost that grows rapidly when f moves away from a desired value f_0. Therefore, we choose an exponential cost function $c(f)$:

$$c(f) = \exp(f - f_0) \tag{4}$$

and determine the value for f_0 experimentally. We found that taking $f_0 = 2.2$ yields good results.

3.2. Optimization algorithm and strategy

The objective function in eq. 3 is strongly non-linear, independently on how the cost function $c(\cdot)$ is chosen and on how the PSNR is evaluated (logarithmic or linear scale). Actually, the non-linearity is induced by the way f is defined; looking back at Section 2.3 it is clear that features are discontinuous: some of them involve counting the number of occurrences or the number of different gray levels within each block. Looking at their formal definition in [2] this is definitely confirmed by the extensive use of the "equality function" δ. Therefore, we cannot even attempt to calculate any derivative of the objective function, and must consequently exclude from eligible optimization algorithms those based on gradient computation.

However, since the problem has been formulated without constraints other than fixing a class of processing, we can use the Nelder-Mead optmization algorithm [5] to search for a solution to the problem in eq. 3. The Nelder-Mead algorithm uses an iterative approach that only requires evaluations of the objective function: given a function of n unknowns and a starting point, the algorithm forms an $(n-1)$-dimensional simplex whose vertices are slight perturbations of the starting point, then it evaluates the function over each one of these vertices and moves the worst one to another point. The simplex will keep moving along the direction where function evaluates to lower values, and once the final minimum is inside the simplex, it will contract on it. This algorithm proves to be effective and computationally compact (obviously, the complexity of the objective function plays a key role in determining the optimization time), although its convergency has been proved only for a limited number of variables and classes of functions [6].

As often happens in optimization, choosing a good starting point is usually as important as not easy to do. In our specific case, we must specify a linear 3×3 filter whose elements will form the nine-dimensional argument of the objective function. On the other hand, we can reasonably expect the optimal filter not to be very dissimilar for different images. Furthermore, since median filtering has a smoothing effect over the signal, we argue that a sharpening filter would probably delete median footprints; of course, it would also revert (to some extent) the desired effect, so it cannot be an acceptable solution as it is. Based on these considerations, we use this approach to practically find a counter-filtered image:

1. Select a set of "estimation images";
2. Solve the optimization problem in 3, using a standard sharpening filter as starting point, for all estimation images;
3. Choose the experiment in which the best objective value was obtained, and use that filter as starting point for all other images.

Notice that these steps are not providing a filter that is to be used for all images: they give a hopefully good starting-point for the ad-hoc optimization, that has to be run separately for each different image. As a matter of fact, this last optimization will probably just "fine-tune" the given filter to yield the best tradeoff between distortion and footprint concealment over the specific image.

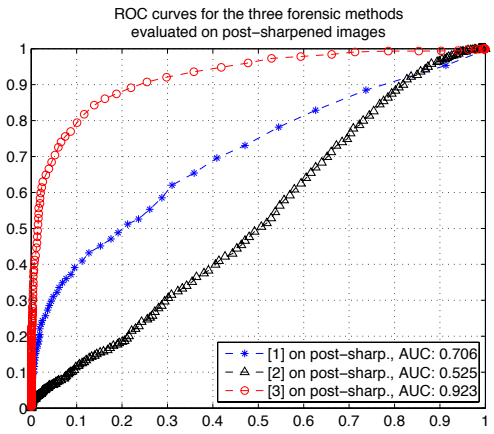

Fig. 1. ROC curves for the three median filtering detection methods evaluated on images of the UCID dataset, along with the resulting Area Under Curve (AUC). Images have been 3×3 median-filtered and post-sharpened. Performances of detectors on simply median-filtered images are almost ideal (see table 1) and therefore are not reported.

4. EXPERIMENTAL RESULTS

In this section we test the proposed counter-forensic technique over the well known UCID [7] dataset of images, which has also been used in all the cited works about median filtering detection. As stated in previous sections, we focused the design of the counter-forensic filter to conceal the "overall" footprint f defined in Section 2.3, eq. 2; however, we test the performances of all the described detector over both median-filtered and counter-filtered images. First of all, let us show that using a sharpening filter over the image is a good instrument to delete traces: we apply median filtering to the whole UCID dataset, then we filter each image with the following sharpening filter:

$$\begin{bmatrix} -0.1667 & -0.6667 & -0.1667 \\ -0.6667 & 4.3333 & -0.6667 \\ -0.1667 & -0.6667 & -0.1667 \end{bmatrix} \quad (5)$$

and plot ROC curves for the three detection algorithms (figure 1): it is clear that this filtering actually hinders significantly the performance of detection algorithms.

However, sharpened images are not good for the forensic adversary: the mean PSNR between median- and counter-filtered images is 23.3 dB on the UCID dataset. Using the Structural Similarity perceptual metric [8] to evaluate quality, a significant degradation is confirmed (average is 0.81). This motivates our search for a more cautious filtering: we select the first 40 images from the UCID dataset and solve the optimization problem in eq. 3 on each of them, using the filter in eq. 5 as starting point. The best result obtained is an image with $f = 0.4643$ and PSNR = 31.65 (SSIM = 0.97), corresponding to the following filter:

Method	PSNR	SSIM	[1] AUC	[2] AUC	[3] AUC
None	-	1	0.972	1.000	0.999
Sharpen.	23.3	0.815	0.706	0.525	0.923
Proposed	30.77	0.940	0.709	0.679	0.924

Table 1. Mean values for PSNR (in dB) and SSIM between median- and counter- filtered images (UCID dataset), and AUC obtained with the three forensic detectors.

$$\begin{bmatrix} -0.1447 & -0.1253 & 0.2541 \\ 0.2557 & 1.2685 & -0.4993 \\ -0.1715 & -0.2050 & 0.3660 \end{bmatrix} \quad (6)$$

It should be noted that, although not being constrained to do so, the filter in 6 yields almost unitary sum (0.998). This is not surprising (maximization of PSNR strongly favors this kind of filters) but supports our choice of introducing the distortion measure in the objective function of the optimization problem and not as a constraint. We also point out that optimization leads to a filter that is not symmetric: this characteristic contributes to the disruption of traces introduced by filtering, and therefore should not be eliminated by imposing additional constraints. For each image in the dataset, we use the filter in 6 as a starting point for the Nelder-Mead algorithm. This time, we obtain values for PSNR and SSIM reported in table 1, and ROC curves reported in figure 2: we see that counter-filtered images are a good approximation of median filtered ones (especially from a perceptual point of view) and that performances of forensic detection methods are still seriously hindered, even though the detector proposed in [4] retains an acceptable discrimination rate. By way of example, we report in fig. 3 one median filtered image and

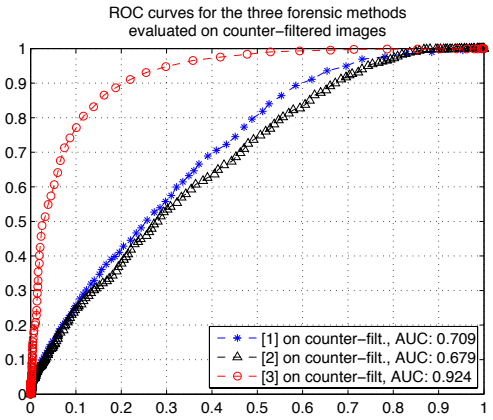

Fig. 2. ROC curves for the three median filtering detection methods on counter-filtered images. Performance of detectors on simply median-filtered images are almost ideal (table 1).

Median filtered | Counter-filtered | Difference (logarithm)

Fig. 3. An example of median filtered image and its counter-filtered version. As visible in the difference (logarithm is taken to enhance visibility) between the two, filtering mostly affects regions with higher variance, resulting in a perceptual high fidelity.

its counter-filtered version the perceptual fidelity is satisfactory, while fingerprints have been removed. Although only results from experiments with 3×3 median filtering are reported here, the method yields equivalent performances even in concealing 5×5 median filtering. This is an important fact because, being the Nelder-Mead optimization computationally heavy, the optimization of big kernels would be rather problematic. On the other hand, future work may explore the possibility of reducing the search space by imposing more constraints to the problem, so to mitigate its complexity.

As a last consideration, it may be objected that the proposed counter-forensic approach is not useful when median filtering was meant to remove traces of resampling (as suggested in [2]), since applying the linear counter-filtering would reintroduce correlations between neighboring pixels. While this is true in principle, median filtering does not seem to be the best way to hide resampling, since a slight JPEG compression would suffice to disrupt correlations [9].

5. CONCLUSION

We propose a simple yet effective method for concealing traces of median filtering in uncompressed digital images. The method exploits knowledge of features used by existing techniques for median filtering detection [2] [3] [4]: an optimization problem is devised that, for a given image, yields a linear filter that allows to remove footprints while keeping the fidelity between the processed image and the counter-processed one as high as possible. Experiments show that performance of detectors in [2] and [3] are seriously hindered by filtering with such a kernel, while the detector in [4] is still affected but in a slighter way. Future work may consider the use of composed objective functions for the optimization problem, so to obtain a filter that simultaneously conceal traces searched by different detectors.

6. REFERENCES

[1] Matthias Kirchner and Rainer Bhme, "Tamper hiding: Defeating image forensics.," in *Information Hiding'07*, Jun. 2007, pp. 326–341.

[2] H. Yuan, "Blind forensics of median filtering in digital images," *Information Forensics and Security, IEEE Transactions on*, vol. 6, no. 4, pp. 1335 –1345, Dec. 2011.

[3] G. Cao, Y. Zhao, R. Ni, L. Yu, and H. Tian, "Forensic detection of median filtering in digital images," in *Multimedia and Expo (ICME), 2010*, Jul. 2010, pp. 89 –94.

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

[5] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, 1965.

[6] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM Journal on Optimization*, vol. 9, pp. 112–147, May 1999.

[7] G. Schaefer and M. Stich, "UCID: an uncompressed color image database," in *SPIE Conference Series*, M. M. Yeung, R. W. Lienhart, & C.-S. Li, Ed., Dec 2003, vol. 5307, pp. 472–480.

[8] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[9] A.C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *Signal Processing, IEEE Transactions on*, vol. 53, no. 2, pp. 758 – 767, Feb. 2005.