FAST CAMERA FINGERPRINT MATCHING IN VERY LARGE DATABASES

Samet Taspinar[1], *Husrev T. Sencar*[1,2], *Sevinc Bayram*[3] *and Nasir Memon*[1,4]

[1]Center for Cyber Security, New York University, Abu Dhabi, UAE
[2]Computer Engineering Department, TOBB University, Ankara, Turkey
[3]Hitachi Europe Ltd., London, UK
[4]Department of CSE, New York University, New York, USA

ABSTRACT

Given a query image or video, or a known camera fingerprint, there is a lack of capabilities for fast identification of media, from a large repository of images and videos, that match the query fingerprint. This work introduces a new approach that improves the computation efficiency of pairwise camera fingerprint matching and incorporates group testing to make the search more effective. More specifically, we jointly leverage the individual strengths of composite fingerprints and fingerprint digests in a novel manner and design two methods that are superior to existing approaches. The results show that under very high-performance requirements, where the probability of correct identification is close to one with a false-positive rate of zero, the proposed search methods are **2-8** times faster than the state-of-art search methods.

Index Terms— Sensor noise, PRNU noise, camera fingerprint, composite fingerprint, fingerprint digests

1. INTRODUCTION

Research in multimedia forensics has established that a digital camera leaves a distinguishing fingerprint-like characteristic in every image it captures due to a phenomenon called *photo response non-uniformity (PRNU)* noise associated with its imaging sensor. PRNU noise exists as a result of the varying sensitivity of each sensing element (pixel) to light. Because this is an artifact of the manufacturing process, PRNU noise appears to be random and unique to each imaging sensor. The procedure for obtaining the PRNU noise profile of a camera, which we refer to as the camera fingerprint or for brevity just fingerprint, from one or more photos or videos and the method for matching two camera fingerprints is now well-known [1, 2, 3, 4, 5]. It has also been shown that this camera fingerprint is hard to remove or forge and survives a multitude of operations performed on the image such as blurring [6], scaling [7], compression [6, 8], and even printing and scanning [9]. Even after anonymization of PRNU noise [10, 11, 12], there are cases where these operations can be de-anonymized [13] or countered [14, 15].

Experimental studies have shown that the accuracy in correctly attributing an image to a particular camera is quite high (with false-positive rates in the order of 10^{-6}) [16]. Therefore, this method of attribution has established itself as a very powerful tool that applies to any source camera verification setting.

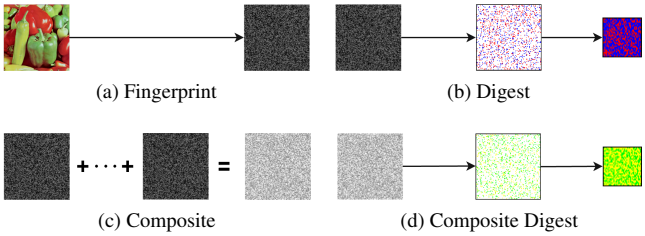

Fig. 1: a) PRNU based fingerprint extracted from an image. b) *Fingerprint Digest* is a short fingerprint obtained from pixels with the highest sensitivity and omitting the rest of pixels. c) *Composite Fingerprint* is obtained by combining multiple fingerprints together. d) *Composite Digest* is obtained as the digest of a composite fingerprint.

However, the ability to search the source of one or more query images in a large dataset consisting of millions of images needs to be developed due to the complexity of the problem. Fundamental improvements are required in two areas; the computational cost of pairwise fingerprint matching and the need for scalable and efficient ways to organize and index PRNU based fingerprints in a database.

A number of approaches have been proposed for efficient camera fingerprint identification over large collections of images. To improve computational efficiency of pairwise matching, Fridrich and Goljan [17] proposed the use of *fingerprint digests*, where only a subset of fingerprint elements that exhibit higher levels of sensitivity (*i.e.,* pixels with higher PRNU) are utilized rather than the whole fingerprint (see Figure 1 a and b). Their results showed that fingerprint digests allow for up to two orders of magnitude dimension reduction without significantly affecting the reliability of the fingerprint verification step. In an alternative approach, Bayram *et al.* [18] proposed reducing the resolution of the PRNU noise intensity measurements by quantizing each fingerprint element into a single bit (which essentially signifies whether a pixel is over- or under-sensitive to light). The reduction in matching accuracy is shown to have a negligible effect while reducing complexity significantly.

More recently, Valsesia *et al.* [19] introduced the idea of applying random projections (followed by binary quantization of projected values) to reduce fingerprint dimension. Their re-

sults revealed that when subjected to same storage limitations, randomly projected fingerprints outperform fingerprint digests in matching accuracy. With this approach, the dimension of the projection subspace (*i.e.*, the size of the compressed fingerprint) is observed to be around two orders of magnitude higher than typical digest sizes (of lengths 512K *vs.* 9.2K), making fingerprint matching computationally more costly.

Approaching the problem from the search efficiency point of view, [20, 21] utilized the idea of a *composite fingerprint* to organize a database of fingerprints into an unordered binary search tree such that each internal node in the tree represents a fingerprint composited from all the fingerprints of its descendant leaf nodes. The ability to evaluate the match of a query fingerprint to a composite fingerprint makes it possible to reject multiple fingerprints at once. However, error tends to increase when so many images are used to create a composite tree ($i.e.$ more than 2^{14}). Results obtained indicated that this approach offers some benefits over the use of fingerprint digests.

To effectively address the source identification problem, one must essentially combine methods proposed for both reducing the complexity of pairwise fingerprint matching and increasing the search efficiency. In this work, we take a step towards this direction by proposing two new strategies: first, we organize a dataset of fingerprints into one or more one-level search trees, where the root node is a composite fingerprint and all the leaves include digests of the fingerprints that composited the root node. In the second strategy, the root-level composite fingerprint is also substituted by its digest. We compare the performance of the proposed search approach to the use of fingerprint digests alone [17] and the composite fingerprint based search tree approach [21]. Results show that the proposed search methods can outperform both schemes.

2. PROPOSED APPROACH

The complexity of linear search, involving pairwise comparison of camera fingerprint with each PRNU noise in a database can be reduced by a group search technique. The underlying idea here is that if none of a group of fingerprints matches with a query, then a fingerprint composited from those PRNU noise patterns is also expected to not match with the query. Thereby, a whole group of fingerprints can be eliminated from the search with only one mismatch decision. However, when the composite fingerprint yields a match decision, the group of fingerprints must either be individually queried or they can be further divided into smaller sub-groups until a matching fingerprint is identified as described. In many practical settings, it is intuitively plausible to assume that, there will only be a limited number of matching entries in the database; therefore, use of composite fingerprints will allow rejecting most fingerprints without actually performing a one-to-one match with them.

The most straightforward way to construct a composite fingerprint is by computing the mean of all PRNU noise values in each fingerprint in the composite. With the increase in the number of fingerprints, the share of each fingerprint in the composite will decrease and the match decision will become less reliable. This is usually measured in terms of the probability of correctly identifying all composites that include a matching fingerprint (*i.e.*, probability of true positive) and the probability of incorrectly selecting composites that have no matching fingerprint in them (*i.e.,* probability of false positive). Since it is not possible to combine an arbitrary number of fingerprints in a dataset into a single composite while achieving high identification rate and low false positive rates, a more viable approach is to divide the dataset into smaller groups. The size of a group can be determined by making a trade-off between the performance and computational efficiency.

The use of composite fingerprints may reduce the number of matches to be performed, but the computational cost of each match remains unchanged. Fingerprint digests, on the other hand, decrease the dimension of camera fingerprints considerably without incurring a significant performance penalty. Essentially, digests can be easily obtained by retaining the highest valued fingerprint elements, and the reduction in computational complexity due to their use is almost directly proportional to the ratio of reduction in the fingerprint size. Although one also needs to store and retrieve indexes for each fingerprint element in the digest for matching (so that the corresponding elements of the query can be selected prior to matching), the simplicity and achievable performance make fingerprint digests suitable replacement for the full-length fingerprints.

Since fingerprint digests and composite fingerprints reduce different aspects of computational complexity we explore two different ways of incorporating their strengths. One approach is to create composite fingerprints by partitioning the dataset into groups of fingerprints and to perform a two-stage search. The first stage is to perform a linear search on the fingerprint composites rather than on all fingerprints in the database. The second stage is launched only if a composite fingerprint is identified to potentially include a matching fingerprint. In that case, digests of contributing fingerprints, as opposed to their full-length versions, are correlated with the query. To further improve the search efficiency, we also considered generating digests of the composite fingerprints, *i.e.*, composite digests, used in the first stage of the search. (See Fig. 1 for a depiction of how different fingerprint forms are constructed.)

In the rest of the paper, we will refer to the search strategy that leverages composite fingerprints and fingerprint digests as *composite-digest search tree* (CDST) method, and the latter one that also generates digests of *composite fingerprints as full-digest search tree* (FDST) method. The performances of proposed search methods are compared against linear search over fingerprint digests (LSFD) and against utilizing composite fingerprints based search trees (CFST). Figure 2 provides a graphical representation of all the methods.

Comparison of these schemes on an equal footing ideally requires obtaining performance curves at a fixed level of computation for all schemes. This, unfortunately, is non-trivial as the analytical model involves many free parameters that need to be optimized (*e.g.,* thresholds selected at each level of CFST) and also because it requires taking both processing and I/O related tasks into account in determining the overall complexity.

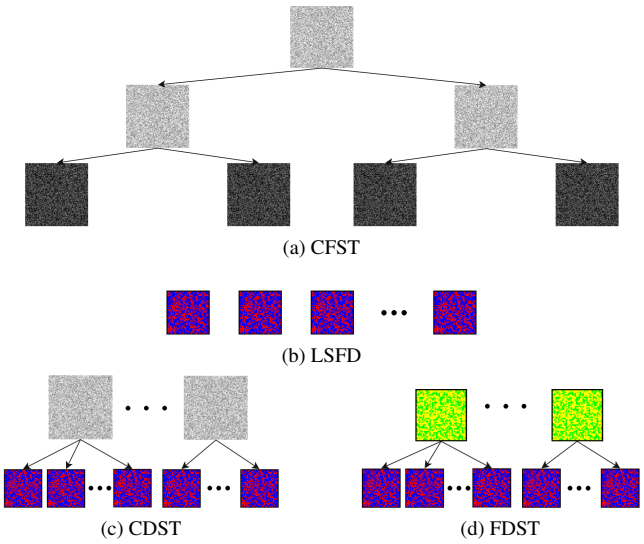

Fig. 2: a) Composite Fingerprint based Search Tree (CFST): Query is matched against the tree of composite fingerprints with fingerprints at the leaves. b) Linear Search over Fingerprint Digests (LSFD): Query is compared with each digest in a linear fashion. c) Composite-Digest Search Tree (CDST): One-level CFST with fingerprint digests at the leaves instead of fingerprints. d) Full Digest Search Tree (FDST): One-level CFST comprised of composite digests and fingerprint digests.

To deal with this problem, in our comparisons, we measure the computational cost for each scheme in attaining the same level of performance, in terms of overall computation time and the number of correlations (which determine the number of multiplications to be performed). Next, we describe our test settings in detail and provide comparison results obtained on actual fingerprints extracted from a large number of images.

3. EXPERIMENTAL RESULTS

We measure and compare the performance composite-digest and full-digest search trees with CFST and LSFD. The performance is determined based on the ability to identify images in the database that are captured by known cameras. The goal is to achieve a high level of identification with very low incorrect matches. This essentially corresponds to the case where the probability of detection (P_D) is set close to one while the probability of false positives (P_{FA}) is set close to zero.

We chose three *known* cameras (Samsung GT-I9300, Sony DSC-W130, and Apple-iPhone 5) for positive queries and 18 other cameras for negative queries. 100 images captured from each camera were used to create fingerprints. We downloaded a large collection of images from the Flickr and 20 images were added from known cameras and none from 18 other cameras. All images were cropped from $(0, 0)$ indexes to the size of 1024×1024 pixels to eliminate variations in sensor sizes due to the different camera makes and models.

All search methods were implemented in C++ with Visual Studio 2010 and OpenCV libraries. The testing environment consisted of a dual-core (Intel Xeon E5-2637 v2 3.50 GHz) CPU system with 32 GB memory, running on Windows 7. To allow for an easier comparison all methods were implemented as a single thread.

3.1. Test Setting

The primary objective of this test was to evaluate the efficiency of four search methods (CDST, FDST, CFST and LSFD) and determine the computational costs of each when $P_D = 1$ and $P_{FA} = 0$, which is the best achievable performance for a search scheme. Since the search tree based methods partition the database of fingerprints into groups and create a composite fingerprint from each group, we examined the impact of group size on search efficiency. We considered composites of different sizes ($64, 128, 256, 512, 1024, 2048$, and 4096) where the size indicates the number of fingerprints in a composite.

In both of the proposed methods perform a search over digests of fingerprints that contribute to each composite when a composite yields a match. In our tests, we set the length of digests to 120×120 (with more than 70 times reduction in original size) which was experimentally determined to be the minimum length satisfying the performance requirement for our dataset using LSFD. In the composite-digest search tree method, the size of each composite fingerprint is the same as the original fingerprint length (*i.e.,* 1024×1024); whereas in the full-digest search tree, the size of the composites are further reduced by utilizing digests of composite fingerprints. In tests, we considered various digest lengths where the size of composite fingerprints is decreased by a factor of 16 to 64 (*i.e.* from 1024×1024 elements to 128×128, 144×144, 192×192 or 256×256 elements).

To evaluate the performance of the methods under these settings, we used a total of $131,072$ images that were randomly divided into eight equal-sized subsets (*i.e.,* eight clusters of $16,384$ images). In five subsets we inserted 1 image from each known camera, and in the last three subsets, we inserted 5 images from the known cameras, overall inserting 20 images from each known camera into the dataset. The rest included images downloaded from Flickr with no images from the 18 non-matching cameras. For comparison with the LSFD method, we picked $14,400$ pixels with highest sensitivity from each fingerprint and created 120×120-sized digests from each fingerprint. (We also stored the coordinates of selected pixels as indexes) Similarly, for the CFST, eight search trees were generated and thresholds needed to make a decision at each level of the search tree were determined according to the designated performance goal.

3.2. Performance Comparison

For all methods, a comparison was performed on the basis of average computation time that takes to search each set of $16,384$ images using 21 camera fingerprints as search queries. Comparison results are reported considering null and alternative hypotheses. In the null hypothesis, *i.e.*, H_0, 18 non-matching camera fingerprints are used as queries to search all image sets. The alternative hypothesis, denoted by H_1, represents the search of image sets, each of which includes one or

five matching fingerprints from each of three known cameras, with corresponding camera fingerprints used as queries.

Average computation times for using composite−digest search tree to search $16,384$ fingerprints while $P_D = 1$ and $P_{FA} = 0$ are given in Table 1 for varying sizes of composites. Search times under H_1^1 and H_1^5 are obtained from images sets including one and five matching fingerprints. Similarly, H_0 measurements are obtained from all image sets using the same thresholds as under H_1^1 with search queries only including non-matching camera fingerprints. (We must note here that H_0 measurements corresponding to H_1^5 case yielded similar values. To keep all the tables less cluttered we only present H_0 measurements corresponding to H_1^1.) Results show that under both H_0 and H_1, composite-digest search tree is most efficient when the composite size is set to 1024.

This finding is intuitive since selecting a small composite size results with the creation of a large number of composite fingerprints and does not provide a considerable gain. (*e.g.,* when composite fingerprints are of size 64, the query has to be matched with 256 full-length composite fingerprints.) On the other extreme, where composite size is large, match decisions become less reliable; hence, many composites cannot be eliminated from the search to ensure high identification accuracy.

Table 1: Average Search Times for Composite-Digest Search Tree Method for varying Composite Sizes (in seconds)

	64	*128*	*256*	*512*	*1024*	*2048*	*4096*
H_0	23.0	15.0	12.4	11.8	11.2	14.4	17.4
H_1^1	23.6	18.7	17.8	16.2	14.4	16.7	18.1
H_1^5	24.3	19.4	18.6	17.6	16.3	17.7	18.1

To further improve the search speed, in full-digest search tree method, digests of the composite fingerprints are utilized in the first phase of the search. We consider digest lengths of 16K, 20K, 36K, and 64K elements as opposed to the use of full-length (1024×1024) composite fingerprints. Table 2 provides the average search times for full-digest search tree method in comparison to composite-digest search tree (see Table 1) when $P_D = 1$ and $P_{FA} = 0$. These measurements show that when the digest length for composite fingerprints is set to 36K and the size of composites to 256, 512 or $1,024$ fingerprints, full-digest search tree performs the search much faster than composite-digest search tree.

Table 3 provides the average number of correlations for each method to search a query in a set of $16,384$ PRNU noise patterns as well as the overall computation time. It can be seen that full-digest search tree performs fastest under H_0 due to its ability to reject multiple fingerprints at once. However, under H_1^1 and H_1^5 there is an increase in the computation for all search trees based methods as composite fingerprints are more likely to yield a match and further search has to be conducted. This incurred a more significant slowdown with CFST as its computations involve full-length fingerprints. It must be noted that for large datasets with relatively few matching fingerprints, the performance of all searc tree based methods are expected to converge to their H_0 case. Overall, since the proposed approaches incorporate the use of composite fingerprints and fingerprint digests, it continues to perform $2-8$ times faster than CFST and LSFD methods. (For better comparison, performance results for the original linear search (LS) scheme that uses the full-length fingerprints during the search are also included.)

Table 2: Average Search Times for Full-Digest Search Tree Method for Varying Composite Sizes and Digest Lengths (in seconds)

Digest		*Composite Size*						
		64	*128*	*256*	*512*	*1024*	*2048*	*4096*
16K	H_0	4.0	5.5	5.2	3.3	4.6	5.8	6.7
	H_1^1	7.1	11.8	15.5	16.1	16.5	17.7	20.6
	H_1^5	17.5	20.1	21.1	14.0	17.2	18.3	18.4
20K	H_0	4.2	3.9	4.1	3.5	3.7	4.9	4.8
	H_1^1	9.7	14.1	13.6	9.6	10.3	12.7	12.9
	H_1^5	16.8	14.5	13.7	10.6	10.7	11.8	15.7
36K	H_0	4.0	3.2	3.1	2.9	3.8	5.4	5.3
	H_1^1	7.6	7.4	10.1	10.9	9.3	10.2	12.4
	H_1^5	16.7	12.6	10.8	10.1	10.9	12.6	14.0
64K	H_0	5.6	4.0	3.9	3.8	4.4	5.9	6.0
	H_1^1	8.1	7.5	11.9	13.1	12.0	14.5	20.0
	H_1^5	19.4	20.6	13.3	18.3	18.5	21.0	17.8

Table 3: Comparison of search methods

Method		Fing. length	# of corr.	time (sec)
Linear Search		1048576	16384	1163
CFST	H_0	1048576	267	18.8
	H_1^1	1048576	578	41.5
	H_1^5	1048576	668	46.9
LSFD		14400	16384	22.1
CDST	H_0	1048576/14400	16/7596	11.2
	H_1^1	1048576/14400	16/9980	14.4
	H_1^5	1048576/14400	16/11394	16.3
FDST	H_0	20480/14400	32/2653	3.5
	H_1^1	20480/14400	32/7346	9.6
	H_1^5	20480/14400	32/8115	10.6

4. CONCLUSION

The results show that incorporation of composite fingerprint and fingerprint digest ideas speed up the task of source camera identification with no performance drop. The composite-digest search tree approach is approximately 2 times faster than CFST and LSFD, whereas the full-digest search tree based approach yields an 8 times speed up, which is close to 400 times faster than conventional linear search. In terms of storage proposed methods are comparable to LSFD, requiring only 1.4 percent of the total space needed by CFST in storing search trees. Overall, our results demonstrate that proposed methods are superior than CFST and LSFD methods.

5. REFERENCES

[1] Jan Lukáš, Jessica Fridrich, and Miroslav Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, 2006.

[2] Miroslav Goljan, "Digital camera identification from images–estimating false acceptance probability," in *International Workshop on Digital Watermarking*. Springer, 2008, pp. 454–468.

[3] Wiger Van Houten and Zeno Geradts, "Source video camera identification for multiply compressed videos originating from youtube," *Digital Investigation*, vol. 6, no. 1, pp. 48–60, 2009.

[4] Samet Taspinar, Manoranjan Mohanty, and Nasir Memon, "Source camera attribution using stabilized video," in *Information Forensics and Security (WIFS), 2016 IEEE International Workshop on*. IEEE, 2016, pp. 1–6.

[5] Jan Lukáš, Jessica Fridrich, and Miroslav Goljan, "Detecting digital image forgeries using sensor pattern noise," in *Electronic Imaging 2006*. International Society for Optics and Photonics, 2006, pp. 60720Y–60720Y.

[6] Erwin J Alles, Zeno JMH Geradts, and Cor J Veenman, "Source camera identification for low resolution heavily compressed images," in *Computational Sciences and Its Applications, 2008. ICCSA'08. International Conference on*. IEEE, 2008, pp. 557–567.

[7] Miroslav Goljan and Jessica Fridrich, "Camera identification from scaled and cropped images," *Proc. SPIE, Electronic Imaging, Forensics, Security, Steganography, and Watermarking of Multimedia Contents X*, vol. 6819, pp. 68190E–68190E–13, 2008.

[8] Kurt Rosenfeld and Husrev T. Sencar, "A study of the robustness of prnu-based camera identification," in *Media Forensics and Security*, Edward J. Delp, Jana Dittmann, Nasir D. Memon, and Ping Wah Wong, Eds. 2009, vol. 7254 of *SPIE Proceedings*, p. 72540, SPIE.

[9] Miroslav Goljan, Jessica Fridrich, and Jan Lukáš, "Camera identification from printed images," *Proceedings of SPIE*, vol. 6819, pp. 68190I, 2008.

[10] Sevinç Bayram, Husrev T Sencar, and Nasir D Memon, "Seam-carving based anonymization against image & video source attribution," in *Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on*. IEEE, 2013, pp. 272–277.

[11] Ahmet Karaküçük and Ahmet Emir Dirik, "Adaptive photo-response non-uniformity noise removal against image source attribution," *Digital Investigation*, vol. 12, pp. 66–76, 2015.

[12] John Entrieri and Matthias Kirchner, "Patch-based desynchronization of digital camera sensor fingerprints," *Electronic Imaging*, vol. 2016, no. 8, pp. 1–9, 2016.

[13] Samet Taspinar, Manoranjan Mohanty, and Nasir Memon, "Prnu based source attribution with a collection of seam-carved images," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 156–160.

[14] Miroslav Goljan, Jessica J. Fridrich, and Mo Chen, "Sensor noise camera identification: countering counter-forensics," In Memon et al. [22], p. 75410.

[15] Ahmet Karaküük, Ahmet E Dirik, Hüsrev T Sencar, and Nasir D Memon, "Recent advances in counter prnu based source attribution and beyond," in *SPIE/IS&T Electronic Imaging*. International Society for Optics and Photonics, 2015, pp. 94090N–94090N.

[16] Miroslav Goljan, Jessica Fridrich, and Tom Filler, "Large scale test of sensor fingerprint camera identification," in *Proceeding of IS&T SPIE Electronic Imaging 2010 Conference, San Jose, CA, USA - Media Forensics and Security 2009*, Feb 2009.

[17] Miroslav Goljan, Jessica Fridrich, and Tomás Filler, "Managing a large database of camera fingerprints," In Memon et al. [22], p. 754108.

[18] Sevinç Bayram, Husrev T. Sencar, and Nasir D. Memon, "Efficient sensor fingerprint matching through fingerprint binarization," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1404–1413, 2012.

[19] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, "Compressed fingerprint matching and camera identification via random projections," *IEEE Transactions of Information Forensics and Security*, vol. 10, no. 7, pp. 1472–1485, July 2015.

[20] Sevinç Bayram, Husrev T. Sencar, and Nasir Memon, "Efficient techniques for sensor fingerprint matching in large image and video databases," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2010, pp. 754109–754109.

[21] Sevinc Bayram, Husrev T. Sencar, and Nasir Memon, "Sensor fingerprint identification through composite fingerprints and group testing," *IEEE Transactions of Information Forensics and Security*, vol. 10, no. 3, pp. 597–612, March 2015.

[22] Nasir D. Memon, Jana Dittmann, Adnan M. Alattar, and Edward J. Delp, Eds., *Media Forensics and Security II, part of the IS&T-SPIE Electronic Imaging Symposium, San Jose, CA, USA, January 18-20, 2010, Proceedings*, vol. 7541 of *SPIE Proceedings*. SPIE, 2010.