Fast Image Clustering Based on Camera Fingerprint Ordering

FAST IMAGE CLUSTERING BASED ON CAMERA FINGERPRINT ORDERING

Sahib Khan and Tiziano Bianchi

Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Torino, Italy

ABSTRACT

This work presents a new camera fingerprint-based image clustering algorithm. The proposed algorithm is based on sorting the camera fingerprints according to information that is inherently present in images. A ranking index is constructed for each image, taking into account the combined effect of gray-level, saturation and texture on camera fingerprint estimation. Then, camera fingerprints are ordered according to this ranking index and clusters are iteratively constructed using as reference fingerprint the top-ranked fingerprint among the currently un-clustered fingerprints. The algorithm can be optionally implemented with an additional attraction stage to refine clustering. The results confirm that the proposed method achieves a performance comparable to state of the art approaches, with a significantly lower computational complexity. The method can also handle cases in which the number of clusters is much larger than the average size of the clusters.

***Index Terms*—** Image clustering, photo response non-uniformity, computational complexity

1. INTRODUCTION

Knowing the source of an image is very important for forensic experts since it can help finding non obvious clues and solving criminal cases. Sometimes, such information can be obtained from the metadata, e.g., the Exif header. However, this information is not always available, and if available it can be easily modified. It is important to use some stable, non-removable and unique features that are intrinsically present in the image and give information about the acquisition device. For this purpose, it has been observed that each image acquisition device, due to sensor imperfection, adds Sensor Pattern Noise (SPN), mostly contributed by photo response non-uniformity (PRNU)[1, 2]. The PRNU is a sort of camera fingerprint, that is unique for every device [3, 4] and can be used to group images from the same source. In the following, we name this process clustering based on camera fingerprints. This can be useful to find the number of cameras belonging to a suspect and link images from different crime scenes with the suspect's cameras [5].

To properly estimate the camera fingerprint, a sufficient number of flat, unsaturated and uniformly bright images from the same camera are required [2]. But, in a realistic scenario this is not usually possible because, the clustering algorithm is faced with images having very different subjects and exposures, without any prior knowledge about how to group images form the same camera. The camera fingerprint is just estimated from the noise residual of a single image, by subtracting the de-noised image from the original image [6, 7].

Existing camera fingerprint-based clustering algorithms uses the normalized correlation among fingerprints. The normalized correlation is used as a similarity measure, and pairs of fingerprints with normalized correlation above a threshold are considered from the same source.

One of the first clustering algorithms was proposed by Bloy in [8] using enhanced fingerprints. Bloy's algorithm is a multi round process composed of three steps, i.e., finding a pair of matched camera fingerprints and merging them, constructing a cluster on the basis of merged fingerprint, and refining the cluster. The algorithm used the pairwise nearest neighbor (PNN) algorithm, with predefined threshold [9]. Enhanced camera fingerprints were used in [10] by Li. The fingerprints were treated as random variables and Markov random field (MRF) is used to iteratively cluster these fingerprints. Liu et al. presented a graph partitioning strategy using K-nearest neighbor graphs to cluster images [11]. A multi-class spectral clustering algorithm is presented in [12] to partition the vertices of the constructed K-nearest graph.

A hierarchical clustering algorithm using silhouette coefficient as grouping criteria was presented in [13]. To speedup the clustering process , in [14] compressed fingerprints were used, to reduce computational cost. The use of Hus moment vector in [15] improved the results further. This methodology was adopted in [16] for smart-phone clustering. In [17], Lin and Li presented a large-scale clustering (LSC) algorithm. The clustering is done in four different stages, i.e. coarse clustering, fine clustering, attraction and post-processing. The same authors proposed a fast source-oriented image clustering technique using Markov Random Fields (MRF) in [18]. In [5], Phan et al., presented a sparse subspace clustering (SSC) based technique [19]. The technique used the sparse representation of camera fingerprints to cluster images.

There are some serious limitations of high computational cost, I/O cost, large memory requirements, sensitivity to outliers and the need of prior information, due to which most of the classical clustering algorithms [20, 21] are not used for

this problem. Along with this, most of the existing image clustering algorithms [10, 11, 12, 13] can be computationally expensive, since they compute the full cross-correlation matrix among n fingerprints requiring $(n(n-1))/2$ correlations. The situations get worse in the case of large data sets. Another problem occurs when the number of cameras (NC) is much larger than the average number of images captured by a single camera (SC). Several existing algorithms [10, 13, 16, 21] have low performance in this case.
The main objective of our proposed algorithm is to reduced the computational complexity by avoiding n^2 correlation for clustering a data set of n images. The proposed algorithm sorts camera fingerprints according to their quality, using the gray-scale, saturation and texture information of respective images to predict the estimation noise on the fingerprints. Then, images are clustered using these fingerprints as initial attractors, with a significantly reduced computational cost.

2. PROPOSED ALGORITHM

Clustering images on the basis of camera fingerprints would be much easier if we knew the centroids of clusters and used these centroids as attractors to group images from the same camera together. In real scenarios, we do not know such centroids and, usually, we select them from the fingerprints estimated from available images. However, we can assume that fingerprints with lower estimation error would be closer to their centroid. The proposed algorithm uses the same assumption and tries to sort camera fingerprints using inherent information of the respective images to predict fingerprint estimation quality. According to Cramer-Rao Lower Bound on the variance of the estimated fingerprint [2], dark or textured images are inappropriate for fingerprint estimation. For a better estimation of PRNU, the images should be as bright as possible but not saturated and the brightness should be uniformly distributed. Therefore, it can be inferred that fingerprints estimated from dark, saturated or textured image will have high estimation error and are not good centroid candidates.
The detailed implementation of the proposed fast image clustering based on camera fingerprint ordering (FICFO) algorithm is presented in the following subsections.

2.1. Ranking Index Computation

Before starting the clustering process, we propose to compute a ranking index RI using the average gray level, saturation and texture level of each image X_i present in image data set I. Our assumption is that images with high value of RI will result in fingerprints with lower estimation error.
The average and normalized gray-level G and saturation S for an image X_i are calculated as in the following equations, respectively.

$$G_i = \frac{\sum_{j=1}^{d} X_i(j)}{255 \times d} \quad (1)$$

$$S_i = \frac{\sum_{j=1}^{d} (X_i(j) == 255)}{d} \quad (2)$$

where d is the dimension of image X_i.
To calculate the texture T of an image X_i, we use a Laplacian filter that highlights the regions of rapid gray-level change. The Laplacian L_i of an image X_i is given by

$$L_i = imfilter(X_i, A) \quad (3)$$

where, $imfilter(.)$ denotes 2D filtering and A is a kernel that approximates the second order derivative. The texture level T_i is calculated as

$$T_i = \frac{\sum |L_i|^2}{\sum |X_i|^2} \quad (4)$$

Finally, the RI_i for image X_i is obtained by combining the values of G_i, S_i, and T_i according to the following equation

$$RI_i = G_i^{\frac{1}{\alpha}} \times (1 - S_i)^{\frac{1}{\beta}} \times (1 - T_i)^{\frac{1}{\gamma}} \quad (5)$$

where, α, β and γ are factors defining the contribution of G_i, S_i and T_i in RI_i, respectively.
Eq. 5 shows that unsaturated and flat images with average gray scale values will results in high values of RI whereas saturated, highly textured or dark images will result in lower values of RI. So, our assumption is that images with high values of RI will yield fingerprints characterized by a lower estimation error.
Then, the images are processed for fingerprint estimation. A set of camera fingerprints M, standardized to zero mean and unit variance, is obtained from the images in data set I as follows

$$M = \{F_i \mid F_i = \Phi(X_i - D(X_i)) \wedge 1 \leqslant i \leqslant n, X_i \in I\} \quad (6)$$

where $D(.)$ is the de-noising function, $\Phi(.)$ is the standardization function, n is the number of images in the data set, X_i is the i^{th} image and F_i is the camera fingerprint estimated from X_i.
The estimated fingerprints are arranged in decreasing order of RI, to get a set of sorted fingerprints M_S. These fingerprints are then used for clustering.

2.2. Fingerprint Clustering

At a generic clustering step denoted by index K, a cluster $C_K = \emptyset$ and a set of un-clustered fingerprints UC_K are considered. When we start clustering, i.e., $K = 1$, all un-clustered fingerprints are assigned to UC_K i.e. $UC_1 = M_S$. To construct C_K, the K^{th} cluster, the proposed algorithm always selects as reference fingerprint RF_K the first fingerprint from the set of sorted and un-clustered fingerprints UC_K

and assigns it to cluster C_K. If the ranking index is consistent, RF_K will be the best estimated fingerprint among all the un-clustered fingerprints UC_K and the best representative of the respective cluster C_K. The normalized cross-correlation (NCC) ρ between all other fingerprints F_i and reference fingerprint RF_K is calculated as follows

$$\rho(i) = \frac{1}{d} \sum_{x=1}^{d} RF_K[x] F_i[x] \tag{7}$$

where d is the dimension of the fingerprint F_i.

If the NCC ρ between fingerprint F_i and reference fingerprint RF_K has a value greater than or equal to a threshold value Th, F_i is assigned to the cluster C_K, otherwise the fingerprint F_i is assigned to the set of un-clustered fingerprints UC_{K+1}. The threshold value is computed as follows

$$Th = \sqrt{2 \times \frac{1}{d}}\, erfc^{-1}(2 \times PFA) \tag{8}$$

where $erfc^{-1}(.)$ is the inverse of the complementary error function and PFA is the desired probability of false alarm. According to the Central Limit Theorem (CLT), the NCC ρ between two d-dimensional normalized fingerprints, X and Y, from different cameras approximately follows a normal distribution with zero mean and $1/d$ variance, i.e., $\rho(X,Y) \sim N(0, 1/d)$ [22]. Hence, the probability of assigning to cluster C_K a fingerprint from a different camera is bounded by PFA. While constructing the cluster C_K, a total of $|UC_K| - 1$ correlation operations are performed and a total of $|UC_{K+1}| = |UC_K| - |C_K|$ fingerprints are left un-clustered.

To cluster the remaining fingerprints, if any, the cluster index K is incremented by 1, i.e. $K = K + 1$ and the un-clustered UC_K fingerprints, are processed to construct a new cluster C_K by repeating the same procedure. The process continues till all fingerprints are assigned to a cluster and UC_{K+1} remains empty.

The total complexity t_c, of the proposed technique is given by

$$t_c = \sum_{K=1}^{nc} (|UC_K| - 1). \tag{9}$$

The clusters can be further refined using an attraction process. For attraction, all fingerprints in each cluster C_K are averaged and standardized to zero mean and unit variance to get an average reference fingerprint ARF_K for each cluster C_K. All merged reference fingerprints ARF_K are processed by using the previous technique. The clusters whose merged reference fingerprints have NCC ρ greater than threshold Th are combined together, otherwise the clusters are left unaffected. After attraction, refined clusters are obtained. The proposed technique can be implemented with attraction as well as without attraction. Using attraction process increases the computation cost. The total complexity t_c, of the proposed technique with attraction can be estimated as

$$t_c = \sum_{K=1}^{nc} (|UC_K| - 1) + cost_{att} \tag{10}$$

where, $cost_{att}$ is the complexity added by the attraction process and is evaluated experimentally.

Here it is worth noting that, differently from [5, 23], the proposed algorithm does not exclude any image from the clustering process on the basis of darkness, saturation and texture level.

3. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed clustering algorithm has been evaluated on the Dresden image database [24, 25]. The data set is composed of 10960 images from 53 cameras of 18 different models and 10 different brands. To have fingerprints of uniform sizes for processing, the fingerprints are center cropped to 1023×1023 pixels. Camera fingerprints are extracted from the images using the technique mentioned in [1, 2]. To fix the threshold criteria, PFA is set to 10^{-6} in all subsequent experiments.

The following metrics related to clustering accuracy, i.e. precision, recall and F-measure, are calculated to judge the performance of FICFO. Let's denote the ground truth as

$$\Omega = \{\omega_1, \omega_2, \omega_3, ...\omega_{NC}\} \tag{11}$$

where each ω denotes a set of fingerprints coming from the same camera. C is the set of clusters generated by clustering algorithm and is given as

$$C = \{c_1, c_2, c_3, ...c_y\} \tag{12}$$

where each c denotes set of fingerprints assigned to a cluster. The precision P and recall R are calculated from the classes and clusters as given in the following equations

$$P = \frac{\sum_k (max_j |c_k \cap \omega_j|)}{\sum_k |c_k|} \tag{13}$$

$$R = \frac{\sum_j (max_k |c_k \cap \omega_j|)}{\sum_j |\omega_j|} \tag{14}$$

where $|c_k|$ and $|\omega_j|$ are cardinalities of cluster c_k and ground truth class ω_j, respectively, $max_j |c_k \cap \omega_j|$ is used to find the largest number of fingerprints in cluster c_k that comes from a ground truth class and $max_k |c_k \cap \omega_j|$ return the largest number of fingerprints in ground truth class ω_j that are also in a recovered cluster.

The F-measure F is calculated using P and R as

$$F = 2 \times \frac{(P \times R)}{(P + R)}. \tag{15}$$

Complexity reduction cr measures the complexity of FICFO relative to the upper bound complexity $(n(n-1))/2$, and is given by

$$cr = \frac{n \times (n-1)}{2 \times t_c}. \tag{16}$$

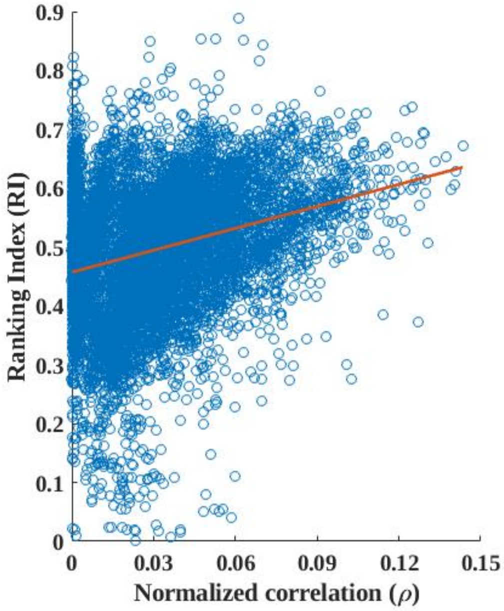


Fig. 1. Analysis of ranking index RI vs normalized correlation ρ.

3.1. Ranking index analysis

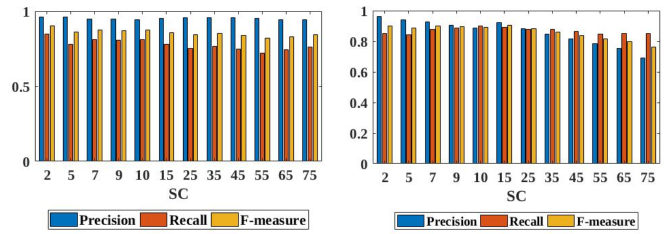
To validate the role of ranking index RI, natural and flat field images of different camera models are used. The flat field images are used to estimate an average reference fingerprint for each camera model. Then, natural images are processed to obtain camera fingerprints and calculate RI for each image using Eq. 5. The NCC ρ between each fingerprint of every camera model and average reference fingerprint of the corresponding camera model is calculated. The RI of all images is plotted versus the values of ρ, as shown in Figure 1. The linear regression is applied to get a linear model of RI in term of ρ and the results are presented in Figure 1. From experiments, it has been observed that slope of the linear model varies with α, β and γ. In the following, we will use RI obtained when $\alpha = 2$, $\beta = 0.5$ and $\gamma = 2$.

The results show that the fingerprints with high value of RI results is high ρ, and RI can be used to predict the correlation between a fingerprint and the corresponding reference fingerprint. From this it can be concluded that fingerprints with high value of RI are the best choice to be used as reference fingerprints during clustering. In all subsequent experiments the fingerprints are sorted using RI with the same values of parameters.

3.2. Analysis of FICFO when $NC \gg SC$

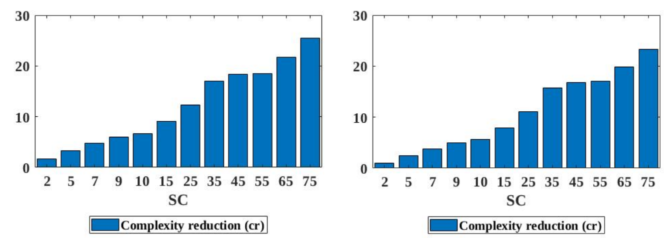
The proposed algorithm without attraction is applied to different sub sets of images selected from Dresden [24, 25] using the same number of cameras, i.e., $NC = 53$, and varying the average number of images from each camera SC. Figure 2 shows the performance evaluation metrics, i.e. precision P, recall R, F-measure F obtained in the different cases. The results show that the proposed technique without attraction perform well for different sizes of data sets and different number of images per camera. Figure 2(a), shows that the evaluation matric P, R and F remain stable with varying SC. The experimental results obtained with attraction, shown in Figure. 2-(b) show that R and F remains almost stable with increase in SC, but P decreases, when SC increases. This can be due to attraction of some wrong clusters. The results also show that FICFO with and without attraction does not suffer from $NC \gg SC$ problem. However, the performance of the FICFO with attraction degrades for $SC \geq NC$, which is due to the attraction of fingerprints with high estimated error.

Figure 3 shows the complexity reduction obtained in the different cases. It has been observed that the complexity of both versions of the FICFO decreases with respect to the upper bound of complexity i.e. $n(n-1)/2$ and hence the complexity reduction cr factor increases. The results also show that when $NC \gg SC$ the cr decreases, hence the complexity increases.



(a) without attraction (b) with attraction

Fig. 2. Evaluation matric of FICFO for different values of SC and fixed $NC = 53$.



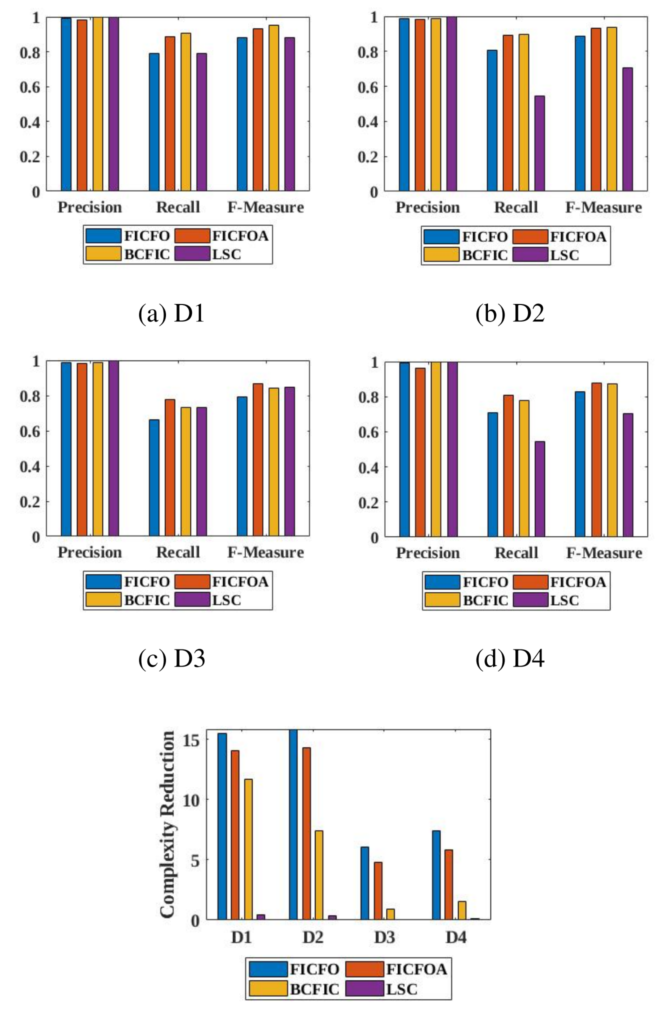
(a) without attraction (b) with attraction

Fig. 3. Complexity reduction cr of FICFO for different values of SC and fixed $NC = 53$.

3.3. Comparisons

Both versions of FICFO i.e. without attraction (FICFO) and with attraction (FICFOA), are further compared with state of the art blind camera fingerprinting and image clustering (BCFIC) [8], large scale clustering (LSC) [17], algorithms using four different data sets. The data set are termed as symmetric, easy asymmetric, hard symmetric and hard asymmetric and labeled as $D1$, $D2$, $D3$ and $D4$, respectively [17]. The $D1$ is composed of images from 25 cameras and 40 images from each camera. The $D2$ also have images from 25 cameras, contributing 20, 30, 40, 50 and 60 images, alternatively. The $D3$ and $D4$ have images from 50 camera, each contributing 20 images in case of $D3$, while alternatively contributing 10, 15, 20, 25 and 30 images in case of $D4$.
The experimental results are shown in Figure 4. The results shows that FICFO and FICFOA perform slightly worse than BCFIC and close to LSC algorithm in case of $D1$, while in $D3$ FICFOA performs slightly better than the other algorithms. In case of $D2$, both FICFO and FICFOA perform better than LSC and slightly worse than BCFIC algorithm. The results obtained on $D4$ show that FICFOA performs slightly better than BCFIC and much better than LSC algorithms.

While comparing FICFO with BCFIC and LSC, the total complexity of all algorithms is calculated. The comparison in term of complexity reduction cr is shown in Figure 4(e). The total complexity t_c of BCFIC is computed in the same way as FICFO because both use the same size of fingerprints. However, as LSC uses two fingerprints of different sizes, called reduced and full fingerprints, the number of correlation operations performed on reduced and full fingerprints are weighed differently. In case of LSC, the total complexity t_c is calculated as $t_c = ncf + (r/d) \times ncr$, where, ncf and ncr are the number of correlation among full and reduced fingerprints respectively, while r and d are the sizes of reduced and full fingerprints, respectively. Here it is important to mention that FICFO performs some computation while calculating G, S, T and RI and also in sorting fingerprints. However, the cost of calculating G, S, T and RI can be assumed negligible with respect to estimation of fingerprints and also complexity of sorting fingerprints is far less than computing correlation of very long vectors, hence these computations are not considered in computing the total complexity t_c.
The results shows that fingerprint ordering reduces the complexity and it can be seen that the complexity of FICFO and FICFOA is quite less than the complexity of both BCFIC and LSC algorithm. BCFIC algorithm has high computation cost because it performs three rounds to construct a single cluster. These rounds are repeated for each cluster. Conversely, FICFO algorithm selects the best estimated fingerprint among all un-clustered, as a reference, to construct a cluster. Due to attraction, FICFOA algorithm has a slightly larger complexity, but since only the cluster centroids are considered, the overall complexity does not increases significantly. The LSC complexity is quite obvious due to coarse clustering, fine clustering and attraction.



(e) Complexity reduction cr

Fig. 4. Comparison of FICFO and FICFOA with state of the art algorithms

4. CONCLUSIONS

We proposed a fast and efficient image clustering algorithm based on camera fingerprints. The algorithm computes a ranking index RI indicating the quality of each estimated fingerprint, and uses fingerprints having higher RI as attractors to form clusters. The results obtained on different subsets of the Dresden dataset show that the proposed clustering algorithm performs comparably better than prior related work, with significantly lower computational complexity. The proposed algorithm is suitable for large data sets, due to the fact that computational complexity per image decreases as the size of the image data set increases. At the same time, the proposed algorithm is also robust when the size of clusters is small compared to the number of cameras, which is a typical problem in this kind of applications.

5. REFERENCES

[1] J. Luks, J. Fridrich, , and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Trans. Inf. Forensics Security.*, vol. 1, no. 2, pp. 205–214, 2006.

[2] M. Chen, J. Fridrich, M. Goljan, and J. Luks, "Determining image origin and integrity using sensor noise," *IEEE Trans. Inf. Forensics Security.*, vol. 3, no. 1, pp. 74–90, 2008.

[3] C.T. Li and Y. Li, "Digital camera identification using colour-decoupled photo response non-uniformity noise pattern," in *Proc. IEEE Int. Symp. Circuits Syst.* IEEE, 2010, pp. 3052–3055.

[4] T. Filler, J. Fridrich, and M. Goljan, "Using sensor pattern noise for camera model identification," in *15th IEEE Int. Conf. Image Process.* IEEE, 2008, pp. 1296–1299.

[5] Q.T. Phan, G. Boato, and F.G. De Natale, "Image clustering by source camera via sparse representation," in *Proc. Int. Workshop on Multi. Forensics Secur.* ACM, 2017, pp. 1–5.

[6] S. Georgievska, R. Bakhshi, A. Gavai, A. Sclocco, and B. van Werkhoven, "Clustering image noise patterns by embedding and visualization for common source camera detection," *Digital Investigation.*, vol. 23, pp. 22–30, 2017.

[7] C.T. Li, "Source camera identification using enhanced sensor pattern noise," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 5, no. 2, pp. 280–287, 2010.

[8] G.J. Bloy, "Blind camera fingerprinting and image clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 532–534, 2008.

[9] W.H. Equitz, "A new vector quantization clustering algorithm," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 10, pp. 1568–1575, 1989.

[10] C.T. Li, "Unsupervised classification of digital images using enhanced sensor pattern noise," in *Proc. IEEE Int. Symp.Circuits Syst.* IEEE, 2010, pp. 3429–3432.

[11] B.B. Liu, H.K. Lee, Y. Hu, and C.H. Choi, "On classification of source cameras: A graph based approach," in *IEEE Int. Workshop Inf. Forensics Secur.* IEEE, 2013, pp. 1–5.

[12] S.X. Yu and J. Shi, "Multiclass spectral clustering," in *IEEE Int. Conf. Comput. Vis.* IEEE, 2003, pp. 313–319.

[13] R. Caldelli, I. Amerini, F. Picchioni, and M. Innocenti, "Fast image clustering of unknown source images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.* IEEE, 2010, pp. 1–5.

[14] F. Gisolf, P. Barens, E. Snel, A. Malgoezar, M. Vos, A. Mieremet, and Z. Geradts, "Common source identification of images in large databases," *Forensic Sci. Int.*, vol. 44, pp. 222–230, 2014.

[15] O.M. Fahmy, "An efficient clustering technique for cameras identification using sensor pattern noise," in *Proc. Int. Conf. Syst., Signals and Image Process.* IEEE, 2015, pp. 249–252.

[16] L.G. Villalba, A.S. Orozco, and J.R. Corripio, "Smartphone image clustering," *Expert Syst. Appl.*, vol. 42, pp. 1927–1940, 2015.

[17] X. Lin and C.T. Li, "Large-scale image clustering based on camera fingerprints," *IEEE Trans. Inf. Forensics Security.*, vol. 12, no. 4, pp. 793–808, 2017.

[18] C.T. Li and X. Lin, "A fast source-oriented image clustering method for digital forensics," *EURASIP J. Image Video Process.*, vol. 1, pp. 69, 2017.

[19] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* IEEE, 2009, pp. 2790–2797.

[20] R.T. Ng and J. Han, "Clarans: A method for clustering objects for spatial data mining," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 5, pp. 1003–1016, 2002.

[21] S. Guha, R. Rastogi, and K. Shim., "Cure: An efficient clustering algorithm for large databases," *ACM SIGMOD Rec.*, vol. 27, no. 2, pp. 73–84, 1998.

[22] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva., "Blind prnu-based image clustering for source identification," *IEEE Trans. Inf. Forensics Security.*, vol. 12, no. 9, pp. 2197–2211, 2017.

[23] Q.T. Phan, G. Boato, and F.G. De Natale., "Accurate and scalable image clustering based on sparse representation of camera fingerprint," *arXiv preprint arXiv:1810.07945.*, Oct 2018.

[24] T. Gloe and R. Böhme, "The dresden image database for benchmarking digital image forensics," *J. Digit. Forensic Pract.*, vol. 3, no. 2-4, pp. 1584–1590, 2010.

[25] T. Gloe, S. Pfennig, and M. Kirchner, "Unexpected artefacts in prnu based camera identification: A dresden image database case-study," in *Proc. ACM Workshop Multi. Secur.* ACM, 2012, pp. 109–114.