Image Clustering by Source Camera via Sparse Representation

Quoc-Tin Phan
University of Trento
Italy
quoctin.phan@unitn.it

Giulia Boato
University of Trento
Italy
giulia.boato@unitn.it

Francesco G. B. De Natale
University of Trento
Italy
francesco.denatale@unitn.it

ABSTRACT

The discovery of clusters of images sharing the same origin based on camera fingerprints, such as Sensor Pattern Noise (SPN), plays an important role in the realm of multimedia forensics. In this work, we present a new approach for grouping images having the same origin based on Sparse Subspace Clustering (SSC). Due to noisy nature of data, we propose to find sparse representation of each camera fingerprint by solving ℓ_1−regularized least squares, and to estimate appropriate parameter in a data-driven fashion. These sparse representations characterize underlying data segmentation. Experimental results confirm the effectiveness of our approach in comparison with existing works.

CCS CONCEPTS

• **Theory of computation** → *Unsupervised learning and clustering*; • **Computing methodologies** → *Computational photography*; *Image processing*;

KEYWORDS

Image clustering, sensor pattern noise, multimedia forensics, sparse subspace clustering

ACM Reference format:
Quoc-Tin Phan, Giulia Boato, and Francesco G. B. De Natale. 2017. Image Clustering by Source Camera via Sparse Representation. In *Proceedings of MFSec'17, Bucharest, Romania, June 06, 2017,* 5 pages.
https://doi.org/10.1145/3078897.3080532

1 INTRODUCTION

Each digital image contains a unique intrinsic trace left by the camera through the image acquisition process, the so-called *camera fingerprint* [14]. Such camera fingerprint uniquely identifies the acquisition camera and can be exploited for multiple forensic purposes. Up-to-date technologies enable camera fingerprint estimation if an analyst has the suspected device in hand, or a set of original images proved to be taken by that camera. In practical applications, it is difficult to fulfill those requirements since often only a set of unsourced images is available to forensic investigators. In order to deal with those circumstances, grouping images with respect to their acquisition cameras is a preliminary step before performing other forensic steps, i.e., estimating the number of cameras, estimating reliable camera fingerprints, linking a suspected image to one cluster with a certain level of confidence. Image clustering by acquisition camera can be solved using noise residuals on images as camera fingerprints, the so-called Sensor Pattern Noise (SPN) [12]. The problem is challenging as SPN is a weak signal, and easily contaminated by lossy compression or geometrical transformations.

In the literature, unsupervised learning techniques have been proposed in order to group images taken by the same camera together. Most of those techniques use normalized correlation of two SPNs as similarity measurement. Bloy in [3] initially proposed a clustering algorithm to group images taken by the same camera. An image is assigned to a group if the correlation between its noise residual and the centroid is greater than a threshold which is adjusted after the number of group members increases. This approach is simple but sensitive to weakly separated groups. That is, if an image is incorrectly assigned into a group at one step, the centroid is falsely updated resulting in error propagation in later steps . Markov Random Field was proposed in [12] to iteratively assign a class label to an image based on the consensus of a small set of SPNs, called *membership committee*. This raises another problem on how to choose a good committee, especially on asymmetric datasets where the number of samples on each class is unbalanced. In [5, 10], hierarchical partitions (a binary tree containing singleton clusters as leaf nodes and whose root node is a cluster containing all data points) are obtained based on hierarchical clustering. Existing hierarchical approaches are sensitive to noise and outliers, since after incorrectly assigning a sample to one class, the algorithms do not consider it again, and this results in the propagation of errors. Multiclass spectral clustering was applied in [2] to partition a graph of unsourced images, and this requires the number of clusters as input. The algorithm starts with two clusters and stops when there exists a cluster containing only one member. This stopping condition is heuristic, and an improved method using normalized-cuts criterion was introduced in [1]. The normalized-cuts based algorithm first partitions the data into two groups, and decides if the current group should be sub-divided by using a pre-defined threshold. A solution dealing with large-scale datasets has been recently proposed in [13]. The idea is to split the dataset into small batches which can be efficiently loaded on RAM, and a simple clustering step is performed to obtain a coarse segmentation. After that, clustering results are improved by a fining step.

In this paper, we propose a novel approach based on sparse subspace clustering to group images with respect to the acquisition camera. Although the number of pixels in a SPN is huge, the number

MFSec'17, June 06, 2017, Bucharest, Romania
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5034-1/17/06…$15.00
https://doi.org/10.1145/3078897.3080532

of correlated pixels between two SPNs of the same camera is often much smaller. Variety of pixels represent redundancies. SPNs of the same camera can be interpreted as lying in a subspace whose dimension is much smaller than the dimension of the ambient space. We show that solving ℓ_1−regularized least squares can recover sparse representations of data which characterize the data segmentation. We also propose a data-driven method to estimate a good regularization parameter. Experimental results confirm the advantages of our method, in terms of quality of clusters, robustness to noisy data, and computational efficiency.

2 THE PROPOSED METHOD

In the creation of a digital image, the image Y output of a camera originates from a noise-free image overlaid partly by Sensor Pattern Noise (SPN), and partly by shot noise [14]. While shot noise is a random component which can be eliminated by averaging, SPN is present in every image taken by that camera and can be used as camera fingerprint. The image content is suppressed by using a denoising filter F in order to obtain noise residual $W = Y - F(Y)$. Various denoising filters are investigated in [6, 7]. We adopt wavelet-based filter as in [14] since this is commonly used in this research area. Since the extracted noise includes non-unique artifacts introduced by color interpolation or JPEG compression, we normalize columns and rows to zero-mean to suppress those artifacts. The similarity between two fingerprints a and b of dimension d can be calculated by normalized correlation (equation (1)).

$$corr(a,b) = \frac{\sum_{i=1}^{d}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{d}(a_i - \bar{a})^2}\sqrt{\sum_{i=1}^{d}(b_i - \bar{b})^2}}, \quad (1)$$

where $\bar{a}, \bar{b}$ denote arithmetic mean of a and b, respectively. Fundamental dimensionality reduction techniques, e.g., PCA, aim at projecting data into a *single* low-dimensional space. However, data might not lie in a single subspace but the union of multiple subspaces. In such cases, finding the subspaces data points lie in reveals hidden clusters. In our problem, finding subspaces that best fit the data is challenging since the data segmentation is unknown. With the assumption that data points lie in the union of multiple subspaces, the segmentation of data can be obtained thanks to Sparse Subspace Clustering (SSC) [9]. SSC relies on the fact that a data point can be written by the *linear* combination of other points in the same subspace.

After extracting camera fingerprints, they are then flattened and stored as columns of $X \in \mathbb{R}^{d\times n}$ In Figure 1, data points lie in

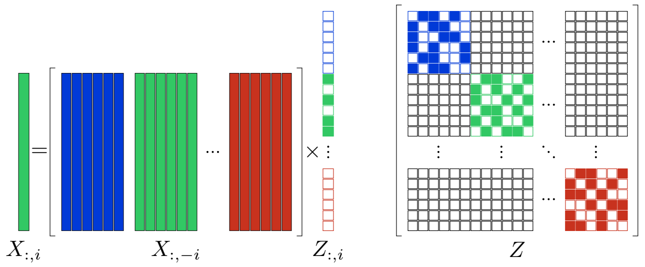

Figure 1: The presentation of each column of X (left), and the sparse representation of entire X (right).

columns of $X_{:,-i}$ which stands for the matrix X with column i^{th} removed . The data point $x = X_{:,i}$ can be written by the linear combinations of other data points with coefficient vector $Z_{:,i} (1 \le i \le n)$. There are probably many such combinations. If a *sparse* solution containing combination coefficients can be efficiently found, it reveals data points in the same subspace. Since non-zero entries of Z characterize the data segmentation, we expect to recover $Z_{:,i}$ as a sparse vector which has non-zero entries corresponding to data points in the same cluster of $X_{:,i}$, and zero entries corresponding to data points in other clusters. Sparse representation of $x = X_{:,i}$ can be theoretically found by solving the following convex optimization problem:

$$Z_{:,i} = \arg\min_{z} \quad \|z\|_1 \quad \text{s.t.} \quad x = X_{:,-i}z. \quad (2)$$

In the problem of camera fingerprint clustering, the data are corrupted by multiple noise sources. Therefore, it is unlikely to recover x exactly. Suppose the error of reconstruction is at most ξ, i.e., $\|X_{:,-i}z - x\|_2 \le \xi$, we can find the sparse solution $Z_{:,i}$ by solving:

$$Z_{:,i} = \arg\min_{z} \quad \|z\|_1 \quad \text{s.t.} \quad \|X_{:,-i}z - x\|_2 \le \xi. \quad (3)$$

On real data, however, the noise level ξ is also unknown. Instead of solving (3), we propose to solve ℓ_1−regularized least squares (also known as Lasso):

$$Z_{:,i} = \arg\min_{z} \quad \gamma\|z\|_1 + \frac{1}{2}\|X_{:,-i}z - x\|_2^2, \quad (4)$$

where $\gamma > 0$ is the parameter controlling the balance between the sparseness and the reconstruction error. There exist a lot of algorithms to solve (4) but the Alternating Direction Method of Multipliers (ADMM) [4] can deal with high dimensionality and large-scale datasets. This is the reason we adopt it to solve our Lasso problem. We first convert (4) to the following equivalent problem:

$$Z_{:,i} = \arg\min_{z} \quad \gamma\|v\|_1 + \frac{1}{2}\|X_{:,-i}z - x\|_2^2 \quad \text{s.t.} \quad z = v, \quad (5)$$

where $\gamma > 0$ is the parameter controlling the balance between the sparseness and the reconstruction error. The following augmented Lagrangian form of (5) is used to eliminate the *equality constraint*.

$$\mathcal{L}_\mu = \gamma\|v\|_1 + \frac{1}{2}\|X_{:,-i}z - x\|_2^2 + y^T(z - v) + \frac{\mu}{2}\|z - v\|_2^2, \quad (6)$$

where y is Lagrangian multiplier and $\mu > 0$ is the augmented Larangian parameter. The scaled form of augmented Lagrangian can be written as follows:

$$\mathcal{L}_\mu = \gamma\|v\|_1 + \frac{1}{2}\|X_{:,-i}z - x\|_2^2 + \frac{\mu}{2}\|z - v + u\|_2^2 - \frac{\mu}{2}\|u\|_2^2, \quad (7)$$

with $u = \frac{y}{\mu}$.

In the iteration $(k+1)$, ADMM optimizes z and v in an *alternating* fashion by keeping one variable fixed and updating the other. The algorithm can converge efficiently to an inexact yet acceptable solution when $\|z - v\|_2 \to 0$. The procedure of solving (5) is outlined in Algorithm 1. Note that Algorithm 1 solves for entire Z instead of each columns of Z. This proposed vectorized implementation significantly reduces the running time.

It is worth mentioning that the notation $diag(Z)$ is a matrix containing only the diagonal of Z, other elements are all zeros. ℓ_1

norm, Frobenius norm of the matrix M are respectively defined as: $\|M\|_1 = \sum_{i,j} |M_{i,j}|$, $\|M\|_F = \sqrt{\sum_{i,j} |M_{i,j}|^2}$.

Algorithm 1 Lasso solving by ADMM

```
1:  procedure $lasso(X, \gamma, \mu)$
2:      Initialize: $V = 0, U = 0, Z = 0, \epsilon = 10^{-4}$
3:      while not converged do
4:          Fix the others and update $z$ by
                $Z = (X^T X + \mu I)^{-1}(X^T X + \mu(V - U))$
                $Z = Z - diag(Z)$
5:          Fix the others and update $v$ by
                $V = \arg\min_V \quad \gamma \|V\|_1 + \frac{\mu}{2}\|V - (Z + U)\|_F^2$
                $V = V - diag(V)$
6:          Fix the others and update $U$: $U = U + Z - V$
7:          Check convergence condition: $\max_{i,j} |(Z - V)_{i,j}| < \epsilon$
8:      end while
9:      return $Z$
10: end procedure
```

The matrix Z is asymmetric, it therefore cannot be used as a proper similarity matrix. However, an affinity graph W can be constructed from Z where $W_{i,j} = |Z_{i,j}| + |Z_{j,i}|$. Here we denote W as *similarity matrix* even if $W_{i,j}$ do not explicitly represent spatial distance. The diagonal of Z consists of all zeros which mean that a data point cannot be expressed by itself, thus resulting in useless solution.

Clustering algorithms usually require a similarity matrix and the number of clusters K as parameters. There exist algorithms able to estimate K by searching over possible segmentations and selecting the best segmentation that maximizes or minimizes a criterion. In this work, we adopt a simpler yet efficient approach based on *eigengap heuristic* [16] to find an estimate of K. The whole procedure of clustering is summarized in Algorithm 2.

Algorithm 2 Camera fingerprint clustering

```
1:  procedure $SSC(X, \gamma, \mu)$
2:      $Z = lasso(X, \gamma, \mu)$
3:      for $j = 1, \cdots, n$ do
4:          Normalize $Z_{:,j}$ as $Z_{:,j} = \frac{Z_{:,j}}{\|Z_{:,j}\|_\infty}$
5:      end for
6:      Compute the similarity matrix $W = |Z| + |Z|^T$
7:      Compute normalized Laplacian matrix $L$ from $W$
            $D_{i,i} = \sum_{j=1}^{n} W_{i,j}$ and $L = I - D^{-1/2} W D^{-1/2}$
8:      Estimate $K$ using eigengap heuristic
9:      Apply spectral clustering [15] to partition data
10: end procedure
```

The effectiveness of eigengap heuristic closely relies on how well clusters are separated. Observing the similarity matrix W, the node i is connected with the node j by the edge whose weight is $|Z|_{i,j} + |Z|_{j,i}$. The regularization parameter γ therefore has a strong impact on clustering results. In the next section, we describe a data-driven method to find a good γ.

3 EXPERIMENTAL RESULTS

In order to assess the effectiveness of the proposed method, we define one development set and four testing datasets (details in Table 1) containing images from two benchmarking datasets RAISE [8] and DRESDEN [11]. The development set is used only for parameter estimation. The symmetry of a dataset refers to the balance between number of images on each camera. This property directly influences the effectiveness of clustering algorithms. The symmetric dataset $\mathcal{D}_1, \mathcal{D}_2$ are designed so that each camera has equal number of images, while this number is not fixed on asymmetric dataset $\mathcal{D}_3, \mathcal{D}_4$.

Table 1: Details of datasets used in our experiments

Name	From	#models	#cameras	#images / camera	Total
$\mathcal{D}_1$	RAISE	3	3	50	150
$\mathcal{D}_2$	DRESDEN	10	10	100	1000
$\mathcal{D}_3$	RAISE	3	3	50, 75, 100	225
$\mathcal{D}_4$	DRESDEN	10	10	40, 60, 80, 100	660
$\mathcal{D}_{ev}$	DRESDEN	5	5	100	500

The top-left regions of all testing images are cropped in order to have images of the same size 512 × 512. We select the F1-measure ($\mathcal{F}$) as the main evaluation metric, which is computed based on Precision ($\mathcal{P}$) and Recall ($\mathcal{R}$) measures:

$$\mathcal{F} = 2 \cdot \frac{\mathcal{P} \cdot \mathcal{R}}{\mathcal{P} + \mathcal{R}}, \quad \mathcal{P} = \frac{TP}{TP + FP}, \quad \mathcal{R} = \frac{TP}{TP + FN},$$

where

- True Positive (TP): number of image pairs from the same cluster which are assigned to the same cluster.
- False Positive (FP): number of image pairs from different clusters which are assigned to the same cluster.
- True Negative (TN): number of image pairs from different clusters which are assigned to different clusters.
- False Negative (FN): number of image pairs from the same cluster which are assigned to different clusters.

Beside $\mathcal{F}$, we also report comparison on the ratio of the number of discovered clusters K_p to the number of ground-truth clusters K_g. The closer to one this score is, the better quality of clusters the algorithm produces.

3.1 Parameter settings

Parameter estimation is conducted on D_{ev} to find the good configuration in our system. The regularization parameter γ controls the natural trade off between false discovery and true discovery rate. The larger γ results in sparser solutions making few false discoveries. At the same time, we also expect to have many true discoveries by making the solution less sparse. We define a criterion based on *normalized cuts* ($Ncuts$) and *eigengap*.

Ncuts is a measure of the proportion of connections going out from one cluster, and is defined as in [17]:

$$Ncuts_K = \frac{1}{K} \sum_{c=1}^{K} \frac{W(A_c, \bar{A}_c)}{W(A_c, \bar{A}_c) + W(A_c, A_c)}, \tag{8}$$

where K refers to the number of clusters, and the similarity matrix W and the partition $A_1, \cdots, A_K$ are known. Note that $\bar{A}$ is the complement of A and $W(A,B) = \sum_{i \in A, j \in B} W_{i,j}$.

We apply eigengap heuristic in our method to select K such that sorted eigenvalues $\lambda_2, \cdots, \lambda_K$ of the normalized Laplacian matrix are relatively small but λ_{K+1} is large. The parameter γ should be selected so that it maximizes the eigengap $\Gamma_K = \lambda_{K+1} - \lambda_K$ and minimizes neighbor eigengaps $\Gamma_{K-1}, \Gamma_{K+1}$. Composing two criterions, normalized cuts and eigengap, we seek for γ minimizing the following cost function:

$$J(K) = Ncuts_K + \frac{1}{\Gamma_K} + \Gamma_{K+1} + \Gamma_{K-1}. \tag{9}$$

On the development set $\mathcal{D}_{ev}$, we vary γ from 0.0001 to 0.02 and select the best value $\gamma = 0.0018$.

3.2 Results and comparison

Performance is compared with existing works described in Section 1. We implement Markov Random Field (MRF) [12] following the configuration suggested by the paper in which 1/5 of the entire dataset is selected as committee members. MRF applies K–means to find the *reference similarity* for each fingerprint, thus its output is nondeterministic. We therefore run the algorithm 10 times and average all observed results. For Multiclass Spectral Clustering (MSC) [2], the input of the algorithm is a star graph which is constructed from a similarity matrix. We build the graph with 5 nearest neighbors since this configuration results in best performance as reported in [2]. Performance variation is also observed in MSC due to random initialization, we also run the algorithm 10 times and averaging all results in the same manner as with [12]. Hierarchical clustering (HC) is applied by [5, 10], but we select [10] for comparison since it is faster than [5]. In Spectral Clustering with Normalized Cut criterion (SCNCuts) [1], the threshold τ is estimated again due to its lack of robustness using different datasets. Instead of selecting $\tau = 0.037$ as reported, we estimate τ again on D_{ev} and select $\tau = 0.032$. Most recently, large-scale method (LS) [13] has been proposed to deal with large-scale datasets by using divide-and-conquer strategy. We implement and test LS using all best parameters reported in [13].

As can be seen from Table 2 and 3, all methods result in high $\mathcal{F}$ on $\mathcal{D}_1$ and $\mathcal{D}_3$ as these datasets comprise raw images from RAISE. SCNCuts performs slightly better than other methods on $\mathcal{D}_1, \mathcal{D}_3$. However, the goodness of SCNCuts completely relies on a fixed threshold deciding whether a cluster should be partitioned into two smaller clusters. This issue makes SCNCuts unable to discover correctly the number of clusters on $\mathcal{D}_2, \mathcal{D}_4$, resulting in low performance. On $\mathcal{D}_2, \mathcal{D}_4$, the number of images is larger and all images undergo JPEG compression. Results show that the proposed method is more robust, both on symmetric and asymmetric datasets. In all cases, our method can discover exactly or almost exactly the number of clusters.

The running time of the proposed method is compared with other methods on all four datasets. All measurements were done on the computer running Linux Ubuntu 14.04 LTS 64 bits, 2.4 GHz Intel(R) Xeon(R) CPU E5-2630 v3, 64 GB RAM.

The key difference between the proposed method and the others is that instead of calculating $\frac{n(n-1)}{2}$ pairwise correlations, we solve the ℓ_1–regularized least squares which can converge after few iterations. As depicted in Figure 2 (a), the proposed optimization is more efficient than normal correlation calculation and LS correlation calculation (divide-and-conquer strategy as in [13]). Total running time is shown for all techniques in Figure 2 (b). It is worth mentioning that LS is specifically designed for large-scale datasets. LS applies fingerprint compression on the coarse step, which requires additional computation. As shown in Figure. 2 (b), LS is slightly slower on small datasets $\mathcal{D}_1, \mathcal{D}_3$, but is faster than methods running normal correlation calculation on $\mathcal{D}_4$. Note that running time is normalized by the number of images.

Table 2: Comparison on symmetric datasets $\mathcal{D}_1$, $\mathcal{D}_2$.

Methods	$\mathcal{D}_1$		$\mathcal{D}_2$	
	$\mathcal{F}$	K_p/K_g	$\mathcal{F}$	K_p/K_g
HC	0.75	28/3	0.62	48/10
MSC	0.98	5/3	0.80	21.7/10
SCNCuts	**0.99**	**3/3**	0.32	2/10
MRF	0.82	9.6/3	0.81	24/10
LS	0.93	5/3	**0.91**	12/10
Proposed	0.97	**3/3**	**0.91**	**11/10**

Table 3: Comparison on asymmetric datasets $\mathcal{D}_3$, $\mathcal{D}_4$.

Methods	$\mathcal{D}_3$		$\mathcal{D}_4$	
	$\mathcal{F}$	K_p/K_g	$\mathcal{F}$	K_p/K_g
HC	0.82	39/3	0.64	25/10
MSC	0.92	5.5/3	0.74	12.7/10
SCNCuts	**0.98**	**3/3**	0.44	4/10
MRF	0.88	10/3	0.50	21.10
LS	0.97	**3/3**	0.89	13/10
Proposed	0.97	**3/3**	**0.90**	**11/10**

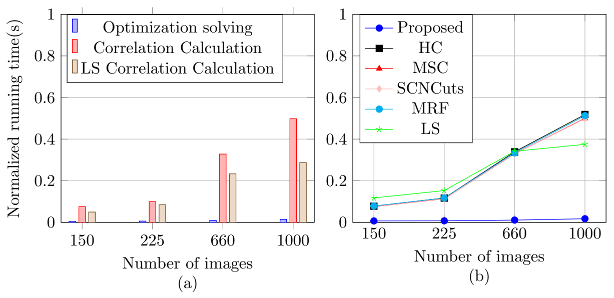

Figure 2: Running time comparison.

4 CONCLUSION

We have proposed a framework to cluster camera fingerprints via sparse representations of data. In our framework, we solve ℓ_1-regularized least squares in order to find sparse combination coefficients which characterize underlying data segmentation. Experimental results confirms the effectiveness of our approach compared to existing works. Our future study will be devoted to deal with low quality images and large-scale datasets.

REFERENCES

[1] I. Amerini, R. Caldelli, P. Crescenzi, A. Del Mastio, and A. Marino. 2014. Blind Image Clustering based on The Normalized Cuts Criterion for Camera Identification. *Signal Processing: Image Communication* 29, 8 (2014), 831–843. https://doi.org/10.1016/j.image.2014.07.003

[2] B. b. Liu, H. K. Lee, Y. Hu, and C. H. Choi. 2010. On Classification of Source Cameras: A Graph based Approach. In *Proc. of IEEE International Workshop on Information Forensics and Security*. 1–5. https://doi.org/10.1109/WIFS.2010.5711446

[3] G. J. Bloy. 2008. Blind Camera Fingerprinting and Image Clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 30, 3 (2008), 532–534. https://doi.org/10.1109/TPAMI.2007.1183

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. 2011. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning* 3, 1 (2011), 1–122. https://doi.org/10.1561/2200000016

[5] R. Caldelli, I. Amerini, F. Picchioni, and M. Innocenti. 2010. Fast Image Clustering of Unknown Source Images. In *Proc. of IEEE International Workshop on Information Forensics and Security*. 1–5. https://doi.org/10.1109/WIFS.2010.5711454

[6] V. Conotter and G. Boato. 2011. Analysis of Sensor Fingerprint for Source Camera Identification. *Electronics Letters* 47, 25 (2011), 1366–1367.

[7] A. Cortiana, V. Conotter, G. Boato, and F. G.B. De Natale. 2011. Performance Comparison of Denoising Filters for Source Camera Identification. In *Proc. of SPIE 7880, Media Watermarking, Security, and Forensics III*, Vol. 7880. 788007–788007–6. https://doi.org/10.1117/12.872489

[8] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato. 2015. RAISE: A Raw Images Dataset for Digital Image Forensics. In *Proc. of the 6th ACM Multimedia Systems Conference*. 219–224. https://doi.org/10.1145/2713168.2713194

[9] E. Elhamifar and R. Vidal. 2009. Sparse Subspace Clustering. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 2790–2797. https://doi.org/10.1109/CVPR.2009.5206547

[10] L. J. García Villalba, A. L. Sandoval Orozco, and J. R. Corripio. 2015. Smartphone Image Clustering. *Expert Systems with Applications* 42, 4 (2015), 1927–1940. https://doi.org/10.1016/j.eswa.2014.10.018

[11] T. Gloe and R. Böhme. 2010. The 'Dresden Image Database' for Benchmarking Digital Image Forensics. In *Proc. of the 2010 ACM Symposium on Applied Computing*. 1584–1590. https://doi.org/10.1145/1774088.1774427

[12] Chang Tsun Li. 2010. Unsupervised Classification of Digital Images using Enhanced Sensor Pattern Noise. In *Proc. of IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*. 3429–3432. https://doi.org/10.1109/ISCAS.2010.5537850

[13] X. Lin and C. T. Li. 2017. Large-Scale Image Clustering Based on Camera Fingerprints. *IEEE Trans. on Information Forensics and Security* 12, 4 (2017), 793–808. https://doi.org/10.1109/TIFS.2016.2636086

[14] J. Lukas, J. Fridrich, and M. Goljan. 2006. Digital Camera Identification from Sensor Pattern Noise. *IEEE Trans. on Information Forensics and Security* 1, 2 (2006), 205–214. https://doi.org/10.1109/TIFS.2006.873602

[15] Andrew Y. Ng, M. I. Jordan, and Y. Weiss. 2001. On Spectral Clustering: Analysis and An Algorithm. In *Proc. of Advances in Neural Information Processing Systems*. 849–856.

[16] Ulrike von Luxburg. 2007. A Tutorial on Spectral Clustering. *Statistics and Computing* 17, 4 (2007), 395–416. https://doi.org/10.1007/s11222-007-9033-z

[17] S.X. Yu and J. Shi. 2003. Multiclass Spectral Clustering. *Proc. of Ninth IEEE International Conference on Computer Vision* (2003), 313–319. https://doi.org/10.1109/ICCV.2003.1238361