Reduced Complexity Image Clustering Based on Camera Fingerprints

(Article begins on next page)

REDUCED COMPLEXITY IMAGE CLUSTERING BASED ON CAMERA FINGERPRINTS

Sahib Khan and Tiziano Bianchi

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

ABSTRACT

This work presents a reduced complexity image clustering (RCIC) algorithm that blindly groups images based on their camera fingerprint. The algorithm does not need any prior information and can be implemented without and with attraction, to refine clusters. After a camera fingerprint is estimated for each image in the data set, a fingerprint is randomly selected as reference fingerprint and a cluster is constructed using this fingerprint as centroid. The clustered fingerprints are removed from the data set and the remaining fingerprints are clustered repeating the same process. A further attraction stage can be included, in which a similar algorithm is performed using the centroids of the clusters found after the first stage. Despite its simplicity, results show that RCIC algorithm has lower computational cost than existing algorithms, while maintaining similar or even better performance. Moreover, the performance of the proposed algorithm is not affected significantly when the number of cameras in the data set is much larger than the average number of images from each camera.

***Index Terms*—** Camera fingerprints, large scale clustering, complexity reduction, image forensics

1. INTRODUCTION

The digital representation of real scenes has brought significant advantages in human life. Along with entertainment, saving memories in pictures, sharing life moments with family and friends on social media, digital images can be used as evidences of important events in a court of law. At the same time, new serious challenges arise, including detection of image modification, identification of image origin, attribution of images to a common source [1, 2]. Therefore, finding the source of images and authenticating their content are very important problems in image forensics. It has been found that each image acquisition device leaves unique intrinsic traces, called camera fingerprints [3, 4]. The traces come from shot noise [5, 6] and pattern noise, with dominant contribution of photo response non uniformity (PRNU). The PRNU is unique, stable and multiplicative in nature and is used as unique camera fingerprint [4, 7].

The camera fingerprint uniquely identifies the acquisition device and is used in multiple forensic applications, like source identification, device linking, camera brand and model identification etc. [7, 8]. The fingerprint extraction and estimation is very easy if the camera, or a significant number of candidate images, are available. However, this become a very challenging problem when only unclassified images are available to forensics investigators, without any information about their source. Nevertheless, it is still desired to cluster images, each cluster composed of images taken by the same camera to link different crime scenes and related evidence to the device owned by suspects.

In the literature, various blind clustering techniques have been presented. The first attempt was made in the work of Bloy [9] and used the pairwise nearest neighbor (PNN) algorithm, with predefined threshold [10] to cluster images on the basis of enhanced fingerprint. In [11] Li used enhanced fingerprints as random variables and Markov random field (MRF) is used to iteratively cluster these fingerprints. Liu et al. adopted a graph partitioning strategy and used K-nearest neighbor graphs to cluster images [12]. A multi-class spectral clustering algorithm is presented in [13] to partition the vertices of the constructed K-nearest graph.

In [14] a faster solution based on hierarchical clustering is proposed, together with a criterion based on a silhouette coefficient. In [15], Gisolf used compressed fingerprints to reduce computational cost. The results are further refined using Hus moment vector [16]. The same solution is adopted in [17] for smart-phone clustering. In [18], Lin and Li presented a large-scale clustering (LSC) algorithm. Lin and Li initially performed coarse clustering followed by fine clustering, attraction and postprocessing, to enhance the clustering results. Li and Lin, in [19], using Markov random field (MRF) proposed a fast source-oriented image clustering technique. In [20], Phan et al., presented a sparse subspace clustering (SSC) based technique [21].

There are many classical clustering algorithms [22, 23] but, due to serious limitations of high computation cost, I/O cost, large memory requirements, sensitivity to outliers and need of prior information, they are not used for clustering camera fingerprints.

Clustering digital images based on camera fingerprints has high computational and I/O cost, as well as large memory requirements. Most of the existing clustering algorithms face these problems. Many of the above clustering algorithms [11, 12, 13] compute the full cross-correlation matrix among the n fingerprints, which requires $(n(n-1))/2$ or more correlations and can be computationally intensive, especially in case of large data sets. Along with these problems,some algorithms such as [11, 14, 17, 23] suffer when the number of cameras (NC) is much larger than the average number of images captured by a single camera (SC). The aim of this work is to efficiently cluster images on the basis of camera fingerprints with reduced computational cost. Namely, we look for solutions where the computational complexity grows linearly in the number of images. Moreover, the proposed solutions should work also when $NC \gg SC$.

2. PROPOSED TECHNIQUE

The proposed technique explores the image data set in a blind way, without any prior information about source camera, number of cameras, number of images captured by a camera and candidate images. As a preliminary step, camera fingerprints, i.e. PRNU patterns, are extracted from all images in the data set. Each image is de-noised using Mihcak filter operation [4], and subtracted from the original image to get the camera fingerprint [2, 6]. A set of, initially un-clustered, camera fingerprints M is obtained from data set of images I and are standardized to zero mean and unit variance using Eq. 1.

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

Where $D(.)$ is the de-noising function, $\Phi(.)$ is standardization function, n is the number of images in data set, X_i is the i^{th} image in data set and F_i is the camera fingerprint obtained from X_i.

The clustering algorithm is applied to the set of un-clustered fingerprints M_k, to construct k^{th} cluster C_k. Initially M_k is equal to M. All extracted fingerprints in the data set are in random order. The total number of fingerprints in the data set is equal to $|M|$. To start clustering, an empty cluster C_k is initiated and a fingerprint F_i is randomly selected as reference fingerprint RF_k and assigned to cluster C_k. The clustering is done by calculating the normalized cross-correlation (NCC) ρ between all other fingerprints F_i and reference fingerprint RF_k, as given by Eq. 2.

$$\rho(i) = \frac{1}{d}\sum_{j=1}^{d} RF_k[j]F_i[j] \quad (2)$$

where d is the dimension of the fingerprints.

If the fingerprint F_i has a NCC ρ value greater than or equal to a threshold value T, it is assigned to the cluster C_k, otherwise the fingerprint F_i is left un-clustered. The threshold T is calculated as given in Eq. 3.

$$T = \sqrt{2 \times \frac{1}{d}}\, erfc^{-1}(2 \times PFA) \quad (3)$$

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 follows a normal distribution with zero mean and $1/d$ variance, i.e., $\rho(X,Y) \sim N(0, 1/d)$ [24]. Hence, the probability of assigning to cluster C_k a fingerprint from a different camera is bounded by PFA.

After processing all fingerprints a total of $|M_k| - 1$ correlation operations are performed to construct cluster C_k. The fingerprints grouped in cluster C_k are removed from the data set M_k and we are left with $|M_k| - |C_k|$ un-clustered fingerprints.

To cluster the remaining fingerprints, new cluster C_{k+1} is initiated and a fingerprint F_i is randomly selected from M_{k+1} as reference fingerprint RF_{k+1}. The un-clustered fingerprints are processed by repeating the same procedure used for constructing the first cluster C_k. The algorithm stops when all fingerprints are assigned to a cluster.

The constructed clusters are further refined by using an attraction stage. In attraction, an average reference fingerprint ARF_k is calculated by averaging all fingerprints in cluster C_k and standardizing it to zero mean and unit variance. Each of the average reference fingerprint is treated as a single fingerprint and the previously adopted procedure is repeated. The clusters whose average reference fingerprints have a NCC ρ greater than threshold T are merged, otherwise the clusters are left unaffected. The attraction is an optional stage and the proposed technique can be implemented with attraction as well as without attraction. The total complexity t_c, of the proposed technique is given by Eq. 4.

$$t_c = \sum_{k=1}^{nc}(|M_k| - 1) + ac \quad (4)$$

Where, $|M_k| = |M_{k-1}| - |C_{k-1}|, \forall\, k \geq 2$ and $M_1 = M$, nc is the number of clusters constructed by the algorithm before attraction and ac is the complexity of attraction stage and is evaluated experimentally.

The performance of the algorithm is assessed by calculating different metrics related to clustering accuracy, i.e. precision, recall and F-measure. Let's denote the ground truth as

$$\Omega = \{\omega_1, \omega_2, \omega_3, ...\omega_{NC}\} \quad (5)$$

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 in Eq. 6.

$$C = \{c_1, c_2, c_3, ...c_y\} \quad (6)$$

where each c denote set of fingerprints assigned to a cluster. The precision P and recall R are calculated from the classes and clusters as given by Eq. 7 and Eq. 8, respectively.

$$P = \frac{\sum_k (max_j |c_k \cap \omega_j|)}{\sum_k |c_k|} \quad (7)$$

$$R = \frac{\sum_j (max_k |c_k \cap \omega_j|)}{\sum_j |\omega_j|} \tag{8}$$

Where, $|c_k|$ is the size of cluster c_k and $|\omega_j|$, is the size of ground truth class ω_j, $max_j |c_k \cap \omega_j|$ 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 given by Eq. 9.

$$F = 2 \times \frac{(P \times R)}{(P + R)} \tag{9}$$

Complexity reduction cr measures the complexity of the proposed algorithm, relative to the upper bound complexity $(n(n-1))/2$, and is given in Eq. 10.

$$cr = \frac{n \times (n-1)}{2 \times t_c} \tag{10}$$

Where, t_c is the total complexity of the proposed technique.

3. EXPERIMENTAL RESULTS

The proposed clustering algorithm has been evaluated on the Dresden image database [25, 26]. The data set is composed of 10960 images from 53 cameras of 18 different models and 10 different brands. To avoid fingerprints with different sizes, the images used for clustering are center cropped to 1023×1023 pixels. Camera fingerprints are extracted from the images using the technique mentioned in [2, 4]. The PFA is set to 10^{-6} in all the following experiments. The same setup is used throughout experimentation.
As the clustering algorithm randomly selects reference fingerprints to construct clusters, each experiment is repeated different number of times to obtain an average performance metric. The proposed algorithm without attraction is applied to a set of 387 images from 18 different cameras. The clustering algorithm is repeated 25, 20, 15 and 10 times. The results obtained in term of variance of the evaluation parameters are listed in Table 1. The experimental results show that the variance of the evaluation parameters is very small and the algorithm is very stable. Hereafter, clustering is repeated 15 times for each experiment and the average values of evaluation metrics are reported.

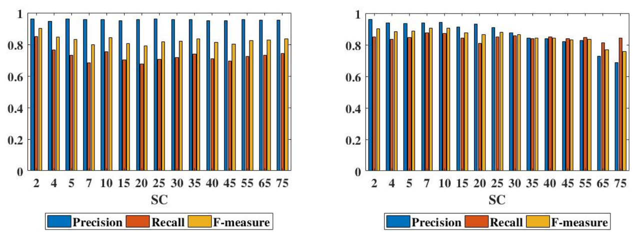
The proposed algorithm with and without attraction, is applied to different sub sets of images selected from Dresden [25, 26] using the same number of cameras, i.e., $NC = 53$, and varying the average number of images from each camera SC. The experimental results are shown in Figure 1, and demonstrate that the proposed technique with and without attraction perform well for different sizes of data sets and different number of images per camera. The results shows that as the average number of images per camera SC changes with respect to number of cameras NC, P is almost constant and

Table 1. Variance of evaluation measures for different No. of experiments

No. of Exp.	σ^2**(P)**	σ^2**(R)**	σ^2**(F)**
25	1.288×10^{-6}	1.504×10^{-4}	5.985×10^{-5}
20	1.994×10^{-5}	1.779×10^{-4}	7.510×10^{-5}
15	1.748×10^{-6}	1.360×10^{-4}	5.112×10^{-5}
10	4.479×10^{-6}	1.120×10^{-4}	4.150×10^{-5}

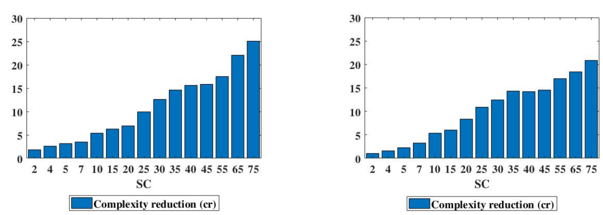
R and F-measure also do not change significantly, without attraction. While, with attraction R and F-measure are stable, but P decreases, due to attraction of some wrong clusters, when SC grows. This shows that the proposed algorithm does not suffer from $NC \gg SC$ problem.

It can be observed that as the size of data set increases the



(a) (b)

Fig. 1. P, R and F-measure of the proposed algorithm vs increasing SC and fixed number of cameras $NC = 53$, (a) without attraction (b) with attraction



(a) (b)

Fig. 2. Complexity reduction of the proposed algorithm vs increasing SC and fixed number of cameras $NC = 53$, (a) without attraction (b) with attraction

evaluation measures decrease, however, this decrease is compensated by the significant reduction in complexity. The results shown in Figure 2, show that as the size of data set increases the complexity of both versions of the proposed algorithm decreases with respect to the upper bound of complexity i.e. $n(n-1)/2$ and hence the complexity reduction cr factor increases.

3.1. Comparison with prior work

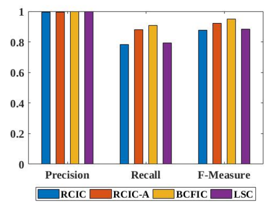
The proposed algorithm with and without attraction, hereafter called RCIC-A and RCIC, respectively, is further compared with state of the art Bloy [9] and Lin and Li [10] algorithms, hereafter called BCFIC and LSC respectively, using symmetric $D1$, easy asymmetric $D2$, hard symmetric $D3$ and hard asymmetric $D4$, image data sets [10]. The $D1$ and $D2$ have images from 25 cameras, each camera contributing 40 to $D1$, and contributing 20, 30, 40, 50 and 60 images, alternatively, to $D2$. 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 complexity t_c of BCFIC is computed in the same way as RCIC by counting the total number of correlations performed. LSC operates on reduced and full fingerprints, therefore number of correlation operations are counted by giving less weight to correlation on reduced fingerprints. In case of LSC the total complexity t_c is calculated as $t_c = ncf + (K/d) \times ncr$, where, ncf and ncr are the number of correlation among full and reduced fingerprints respectively. While, K and d are the sizes of reduced and full fingerprints. The cr for BCFIC and LSC are calculated using Eq. 10.

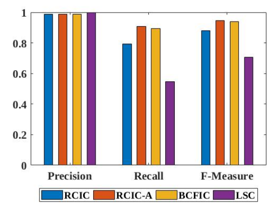
The experimental results are shown in Figure 3. The results shows that in $D1$ the RCIC and RCIC-A algorithm have a slightly lower performance, while in $D3$ the RCIC-A matches the performance of BCFIC and LSC algorithms. However, in the case of $D2$ and $D4$, both RCIC and RCIC-A algorithms perform much better than LSC and slightly better than BCFIC algorithm.

While comparing the complexity of the different algorithms, the RCIC algorithm has quite less complexity than both BCFIC and LSC algorithm. BCFIC algorithm has high computation cost because it perform three rounds to construct a single cluster. It first scan the data set for finding a pair of fingerprints that can be merged and used as centroid, then the centroid is used to scan the data set to construct a cluster. In the third round an average fingerprint is calculated from 50 or all fingerprints in the cluster, if the cluster contains less than 50 fingerprints, and again all un-clustered fingerprints are search for possible match. These rounds are repeated for each cluster. Conversely, in RCIC algorithm, we randomly select a reference fingerprint and scan the fingerprints data set to construct a cluster. The RCIC-A algorithm has slightly large complexity than RCIC due to additional complexity of attraction, but since this considers only cluster centroids, it is much less than the third round in BCFIC algorithm. The LSC complexity is quite obvious due to coarse clustering, fine clustering and attraction. The comparison in term of complexity reduction cr is shown in Figure 3(e).

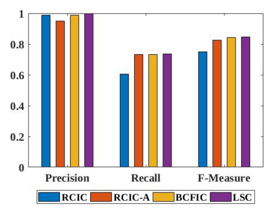
The experimental results show that proposed algorithm both with attraction and without attraction has significantly high cr and the computation cost is quite less than LSC and also BCFIC algorithm.



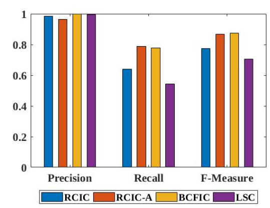
(a) Comparison on $D1$



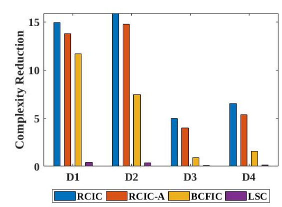
(b) Comparison on $D2$



(c) Comparison on $D3$



(d) Comparison on $D4$



(e) Complexity reduction

Fig. 3. Comparison of RCIC and RCIC-A algorithm with state of the art BCFIC [9] and LSC [18] algorithm

4. CONCLUSION

We proposed a very simple yet efficient algorithm for clustering images according to camera fingerprints. The experimental results obtained on different subsets of the Dresden image database illustrate that the proposed clustering algorithm has lower computational complexity than state of art algorithms, yet its performance is still comparable, or in some cases even better. The computational complexity per image decreases as the size of the image data set increases, which proves the effectiveness of the proposed algorithm for large scale clustering. Along with low complexity and effectiveness for large data sets, 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] H. Pearson, "Image manipulation: Csi: Cell biology," *Nature*, vol. 434, pp. 952–953, 2005.

[2] M. Chen, J. Fridrich, M. Goljan, and J. Luks, "Determining image origin and integrity using sensor noise,"

IEEE Transactions on Information Forensics and Security, vol. 3, no. 1, pp. 74–90, 2008.

[3] 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.

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

[5] G.C. Holst, "Ccd arrays, cameras, and displays," in *SPIE press*. Bellingham, Washington USA, 1998.

[6] J.R. Janesick, "Scientific charge-coupled devices," in *SPIE press*. Bellingham, Washington USA, 2001, vol. 83.

[7] C.T. Li and Y. Li, "Digital camera identification using colour-decoupled photo response non-uniformity noise pattern," in *Proceedings of International Symposium on Circuits and Systems*. IEEE, 2010, pp. 3052–3055.

[8] T. Filler, J. Fridrich, and M. Goljan, "Using sensor pattern noise for camera model identification," in *15th International Conference on Image Processing*. IEEE, 2008, pp. 1296–1299.

[9] G.J. Bloy, "Blind camera fingerprinting and image clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 532–534, 2008.

[10] W.H. Equitz, "A new vector quantization clustering algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 10, pp. 1568–1575, 1989.

[11] C.T. Li, "Unsupervised classification of digital images using enhanced sensor pattern noise," in *Proceedings of International Symposium on Circuits and Systems*. IEEE, 2010, pp. 3429–3432.

[12] B.B. Liu, H.K. Lee, Y. Hu, and C.H. Choi, "On classification of source cameras: A graph based approach," in *Proceedings of the International Workshop on Information Forensics and Security*. IEEE, 2013, pp. 1–5.

[13] S.X. Yu and J. Shi, "Multiclass spectral clustering," in *International Conference on Computer Vision*. IEEE, 2003, pp. 313–319.

[14] R. Caldelli, I. Amerini, F. Picchioni, and M. Innocenti, "Fast image clustering of unknown source images," in *Proceeding of the International Workshop on Information Forensics and Security*. IEEE, 2010, pp. 1–5.

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

[16] O.M. Fahmy, "An efficient clustering technique for cameras identification using sensor pattern noise," in *Proceedings of the International Conference on Systems, Signals and Image Processing*. IEEE, 2015, pp. 249–252.

[17] L.G. Villalba, A.S. Orozco, and J.R. Corripio, "Smartphone image clustering," *Expert Systems with Applications*, vol. 42, pp. 1927–1940, 2015.

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

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

[20] Q.T. Phan, G. Boato, and F.G. De Natale, "Image clustering by source camera via sparse representation," in *Proceedings of the 2nd International Workshop on Multimedia Forensics and Security*. ACM, 2017, pp. 1–5.

[21] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 2790–2797.

[22] R.T. Ng and J. Han, "Clarans: A method for clustering objects for spatial data mining," *IEEE transactions on Knowledge and Data Engineering*, vol. 14, no. 5, pp. 1003–1016, 2002.

[23] S. Guha, R. Rastogi, and K. Shim., "Cure: An efficient clustering algorithm for large databases," *Information Systems*, vol. 26, no. 1, pp. 35–58, 2001.

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

[25] T. Gloe and R. Bohme, "The dresden image database for benchmarking digital image forensics," *Journal of Digital Forensic Practice*, vol. 3, no. 3-4, pp. 1584–1590, 2010.

[26] T. Gloe, S. Pfennig, and M. Kirchner, "Unexpected artefacts in prnu based camera identification: A dresden image database case-study," in *Proceedings of ACM Workshop on Multimedia Security*. ACM, 2012, pp. 109–114.