Signal Processing: *Image Communication* 29 (2014) 831–843

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

Blind image clustering based on the Normalized Cuts criterion for camera identification

I. Amerini [a], R. Caldelli [a,c,*], P. Crescenzi [b], A. Del Mastio [a], A. Marino [d]

[a] *MICC - Media Integration and Communication Center, University of Florence, Florence, Italy*
[b] *DINFO - Department of Information Engineering, University of Florence, Florence, Italy*
[c] *CNIT - National Interuniversity Consortium for Telecommunications, Parma, Italy*
[d] *DI - Department of Computer Science, University of Milan, Milan, Italy*

ARTICLE INFO

Article history:
Received 18 December 2013
Received in revised form
18 July 2014
Accepted 18 July 2014
Available online 30 July 2014

Keywords:
Image forensics
Source identification
Clustering
Normalized Cuts

ABSTRACT

Camera identification is a well known problem in image forensics, addressing the issue to identify the camera a digital image has been shot by. In this paper, we pose our attention to the task of clustering images, belonging to a heterogenous set, in groups coming from the same camera and of doing this in a blind manner; this means that side information neither about the sources nor, above all, about the number of expected clusters is requested. A novel methodology based on Normalized Cuts (NC) criterion is presented and evaluated in comparison with other state-of-the-art techniques, such as Multi-Class Spectral Clustering (MCSC) and Hierarchical Agglomerative Clustering (HAC). The proposed method well fits the problem of blind image clustering because it does not a priori require the knowledge of the amount of classes in which the dataset has to be divided but it needs only a stop threshold; such a threshold has been properly defined by means of a ROC curves approach by relying on the goodness of cluster aggregation. Several experimental tests have been carried out in different operative conditions and the proposed methodology globally presents superior performances in terms of clustering accuracy and robustness as well as a reduced computational burden.

1. Introduction

Digital images can be easily manipulated by common users for disparate purposes so that origin and authenticity of the digital content we are looking at is often very difficult to be assessed without uncertainty. Technological instruments which allow to give answers to basic questions regarding image origin and image authenticity are needed [1]. Both these issues are anyway connected and sometimes are investigated together. However, by focusing on the task of assessing image origin, the two main aspects are to be studied: the first one is to understand which kind of device has generated that digital image (e.g. a scanner, a digital camera) [2–5] and the second one is to succeed in determining which is the specific sensor that has acquired such a content (i.e. the specific brand and/or model of a camera) [6–8]. The main idea behind this kind of approaches is that each sensor leaves a sort of unique fingerprint on the digital content it acquires due to some intrinsic imperfections of the acquisition process. Usually this kind of fingerprint is computed by means of the extraction of PRNU (Photo Response Non-Uniformity) noise [9] from an image through

* Corresponding author at: MICC - Media Integration and Communication Center, University of Florence, Florence, Italy.
E-mail addresses: irene.amerini@unifi.it (I. Amerini), roberto.caldelli@unifi.it (R. Caldelli), pierluigi.crescenzi@unifi.it (P. Crescenzi), andrea.delmastio@unifi.it (A. Del Mastio), andrea.marino.it@gmail.com (A. Marino).

http://dx.doi.org/10.1016/j.image.2014.07.003
0923-5965/

a digital filtering operation. After that, the PRNU of the to-be-checked image is compared with the pre-computed PRNU fingerprints, belonging to a reference set, and then it is assigned to a certain digital camera. Nowadays most of the image source attribution approaches operate in a closed set scenario, where an image is generated by one of n known cameras available during training [9–12].

However, in a realistic situation, images could have been generated by an unknown device not available in the set of cameras under investigation; so it is important to consider the source camera attribution problem in a open set scenario. In particular, it could be the case, for instance, in which the forensic analyst has a set of photographs in hand and he wants to know if those images were taken by the same camera or not. To find an automatic method to solve this kind of problem could have important implication in the case of inspection of big amount of images like photo repository on Internet (e.g. Picasa, Flickr) and on social networks. Usually, in these circumstances, when the number of cameras and images scales up, methods which resort at the adoption of digest-based descriptors are taken into account [13,14] to reduce computational burden but maintaining performances in terms of classification accuracy.

In [15] the image source attribution problem in an open set scenario is faced attesting if a set of images were taken by a specific camera by comparing each of these images to a reference image. The constraint is that the digital camera of the reference image is known even if the analyst does not have physical access to it. Li in [16] proposed a classification system to distinguish among images taken by unknown digital cameras. First of all, PRNU is extracted and enhanced from each image, which is used as the fingerprint of the camera that has taken the image. Secondly, an unsupervised classifier is applied to a training set of PRNUs to cluster them into classes; centroids of previously identified classes are used as the trained classifier to test a new dataset.

Starting from the idea in [16], the paper in [17] presented an improved approach working in a completely open set scenario. The authors proposed to employ a blind classification to group images taken by digital camera by implementing a different enhancer function with respect to [16] to improve PRNU quality and then a HAC clustering procedure is presented. Another blind-classification method to group digital camera images is presented in [18], where the authors formulate the classification task as a graph partitioning problem by using a multiclass spectral clustering. A drawback of this method is the stop criterion, so in [19] the usage of a Silhouette coefficient is proposed to overcome this limitation. Nonetheless, the use of Silhouette coefficient is not able to completely solve the randomness of the multiclass spectral clustering, in fact, the random selection of the starting point in the clustering procedure implies multiple and sometimes very different results in the classification of images.

In this paper, the problem of classifying images without the use of a trained set is faced by overcoming the randomness problem generated by the multi-class spectral clustering. Such an improvement is mainly achieved by resorting at a new and effective clustering procedure, based on *Normalized Cuts criterion* [20], and by introducing a novel and simple method to determine an automatic stop criterion; such a criterion relies on the goodness of cluster aggregation and the estimation of the cut-off threshold is obtained by means of ROC curves [21]. Experimental results are provided to confirm that the proposed technique permits to achieve higher performances both in image grouping (in terms of true/false positive rate – TPR/FPR) and of computational burden with respect to the state-of-the-art methods.

The paper is organized as follows: Section 2 describes the state of the art regarding multi-class spectral clustering method, while Section 3 presents the new proposed one; in Section 4 experimental results are presented and Section 5 concludes the paper. In Appendix A, a detailed description of the evaluation metrics used for the experimental tests is provided.

2. An analysis on multi-class spectral clustering

This section is dedicated to the analysis of the multi-class spectral clustering (MCSC) method, as presented in [18,22] (see Section 2.1). By following this approach, each image is considered as a node in a weighted undirected graph, thus making the clustering task to converge to a graph partitioning problem, where images (nodes) belonging to the same partition are assigned as acquired by the same digital camera. However, such a technique has two main open issues that will be discussed hereafter: firstly, it provides results that strictly depend on the random starting point and, secondly, it needs a stop criterion. In Section 2.2, an in-depth analysis on such issues is presented.

2.1. The MCSC algorithm

Given an image set I of N images (each one indicated as I_i, $i=[1,N]$), a weighted undirected graph G is defined on I such as $G=(V,\varepsilon,W)$, where $V=\{V_i\}$, $i=[1,N]$ is the set of all nodes/images (V_i corresponds to I_i), and ε is the edge set, whose elements are represented by the entries of the affinity matrix $W=\{w_{ij}\}$. The complete structure of the graph can thus be characterized by means of its affinity matrix W. Note that each image is described by its PRNU noise, extracted as previously described in Section 1, and it will be used for the whole clustering process; thus, in the following, when the terms *image* or *noise image* will be adopted, they will state for *PRNU noise extracted from the image*, if not otherwise defined.

Partitioning the graph is accomplished by finding the optimal relaxed solution of the eigensolution matrix of W; the basic steps are presented hereafter:

1. Given the weight matrix W and the number of classes K.
2. Find the optimal eigensolution Z^* and normalize it to $\tilde{X}^*$.
3. Define a new working matrix R^*, having N rows and K columns.
4. Select a random row of the matrix $\tilde{X}^*$ and assign it to the first column of the matrix R^*.

5. For $k=2{:}K$, compute the product $abs(\tilde{X}^* \cdot R^*_{k-1})$, and iteratively increment, by this value, a vector c (dimensions N by 1); then, assign to the k-th column of R^* the i-th row of $\tilde{X}^*$, where $i=\arg\min c$;
6. Repeat:
   - compute $\tilde{X}=\tilde{X}^* \cdot R^*$;
   - build the matrix X^* (N – number of classified images – by K – number of classes) assigning, for each n-th row (i.e. for each image), 1 to the k-th column corresponding to the class the n-th image belongs to, and 0 elsewhere;
   - compute the *Singular Value Decomposition* of $X^{*T}\tilde{X}^*$ and let it be $U\Omega U^T$;
   - re-compute R^* as $\tilde{U} \cdot U^T$;

   until the deviation in the trace of Ω is lower than the machine precision.
7. Output X^*, i.e. the optimal discrete solution.

Refer to [22] for a more detailed description of the procedure. It is worth noting that the MCSC algorithm is based on the initial knowledge of the number of expected classes K, but, in our application scenario, it is not possible to a priori know how many digital cameras are involved in the clustering task; therefore, in [18] the MCSC procedure is repeated iteratively, starting from $K=2$ and increasing, at each iteration, the number of expected groups. This is done until a cluster composed of only one node (image) is found. The cycle then stops and the classification achieved at the previous iteration ($K=K_{STOP}-1$) is assumed to be the optimal one. When implementing and testing the algorithm proposed in [18], the so-called Basic MCSC, we had to deal with two major problems: the first one is the random selection of the row of $\tilde{X}^*$ in the initialization of R^* (step 4 in the previously mentioned procedure) and the second is the stop criterion for the definition of the optimal number of classes. The random selection of a row of $\tilde{X}^*$ impacts onto the repeatability of the experiments; this means that a different final clustering can be achieved on the same image dataset and, consequently, diverse performances in image classification are obtained. It is interesting to notice that such an initialization-dependant difference is quite relevant and, on the other side, the stop criterion which requires to find a single image/node in a cluster does not actually take into account of the inner cluster homogeneity and of the inhomogeneity among clusters. Thus, we studied the behavior of Basic-MCSC by means of *TPR* (*True Positive Ratio*), whose definition is introduced in Appendix A; TPR basically permits to understand how many images are correctly grouped together. In particular, we analyzed the performances on a number of 500 tries for each of 4 selected sets (see hereafter). The images composing all the sets come from 8 different cameras (see Table 1).

Set 1: 300 images, coming from cameras 1–6, 50 images per camera.
Set 2: 400 images, coming from all the 8 cameras in Table 1, 50 images per camera.

Table 1
Digital cameras used in the experiments.

Digital camera	Brand and model
1	Canon EOS 400D DIGITAL
2	Canon DIGITAL IXUS II
3	Canon DIGITAL IXUS i zoom
4	Panasonic DMC-FX12
5	Panasonic DMC-LZ5
6	FujiFilm FinePix J20
7	Olympus FE-120,X-700
8	Samsung VP-MS11

Set 3: 300 images, from cameras 1–6, distributed as 70, 70, 70, 30, 30, 30 images per camera.
Set 4: 440 images, from cameras 1–6, non-uniformly distributed as 30, 150, 80, 20, 60, 100 images per camera.

In Fig. 1, TPR over the 4 different sets is presented. To improve the readability, only 100 of the 500 tries have been shown; the behavior outside this range is qualitatively similar to this anyway. It is worthy to notice that the method's performances are not stable at all because of the random initialization and, furthermore, a high variance with respect to the average (the dotted lines in Fig. 1) is evidenced for all the tested data-sets.

2.2. Some variations

After that, we have tried to modify the algorithm in order to better understand which is the real impact of randomization issue. We have calculated the random index i, in the above algorithm, outside of the K-cycle (step 5), thus having a single value for all the k values within each of 500 tests. In Fig. 2, TPR over Set 1 and Set 2 with such a variation is presented (results achieved for Set 3 and Set 4 were similar and so they are omitted).

Experiments demonstrate that a very limited improvement is anyway obtained when applying the above explained modification to the basic algorithm which however maintains its performance variability due to its intrinsic randomness. Successively, we have tried to redefine the stop criterion, trying to introduce a measure which was not only based on the cardinality of a cluster (one image in a cluster as in [18]) but fundamentally on the internal homogeneity and the external inhomogeneity of the clusters. To do so, we introduced the usage of the *Silhouette coefficient*, which has been successfully used in [17,19]. Silhouette coefficient s_i combines both the measures of cohesion (inside clusters) and separation (among clusters). For each noise image, the coefficient s_i is simply computed as

$$s_i = b_i - a_i \tag{1}$$

where

- a_i (cohesion): the average correlation of the noise image I_i with all other noise images in the same cluster;
- b_i (separation): the average correlation of the noise image I_i with all other noise images in each one of the

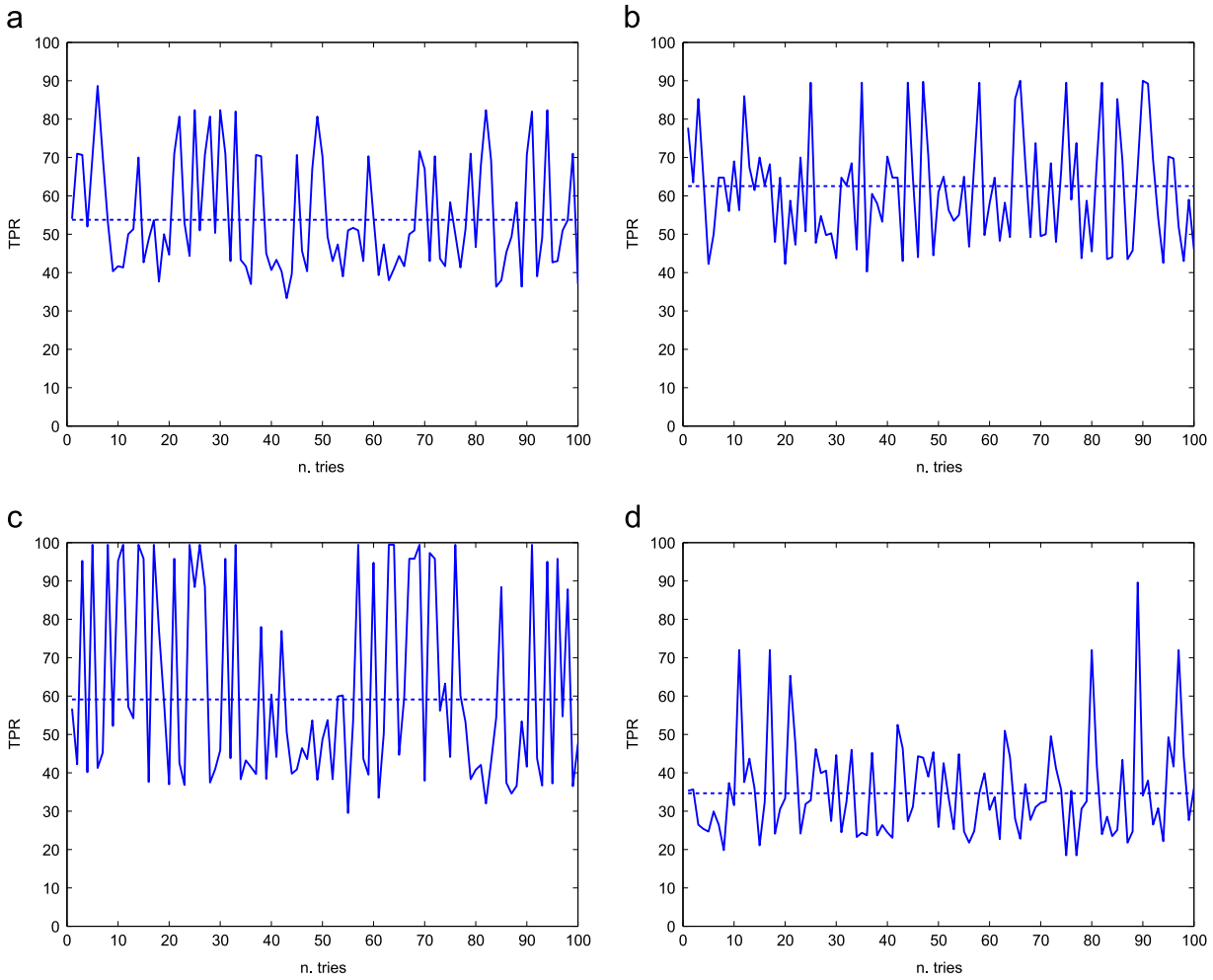

Fig. 1. MCSC algorithm: TPR computed on 500 tries (for the sake of readability only 100 tries are reported) over 4 different sets: (a) Set 1, (b) Set 2, (c) Set 3, and (d) Set 4.

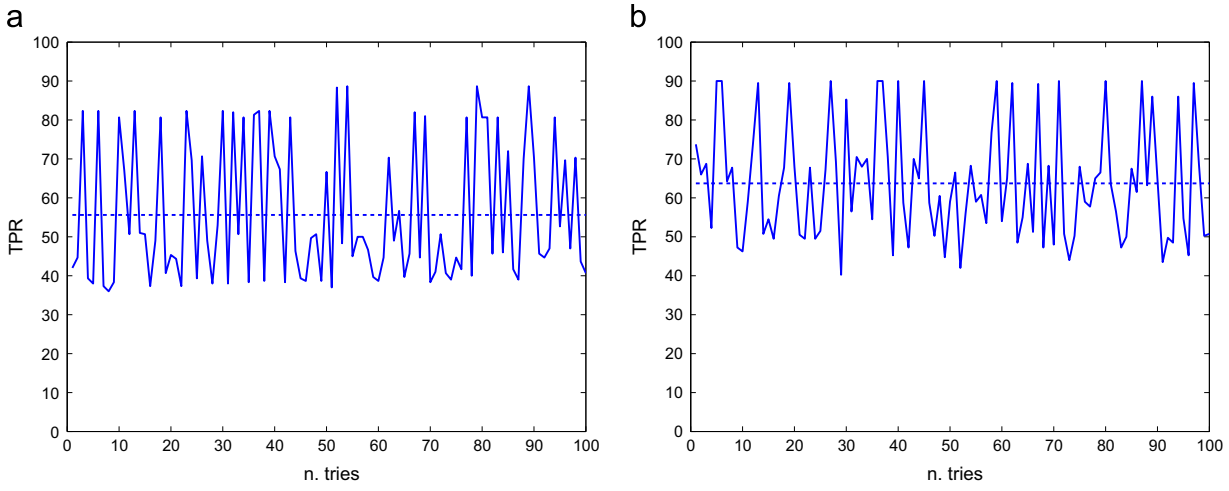

Fig. 2. MCSC algorithm with the random index *i* fixed within each try: TPR computed on 500 tries (for the sake of readability only 100 tries are reported) over 2 different sets: (a) Set 1 and (b) Set 2.

other clusters, taking the minimum value with respect to all clusters.

It is possible to note that a very negative value of s_i comes from a separation value b_i which is highly negative and a cohesion a_i which is very positive: this is the desired situation, indicating that clusters are highly homogeneous inside each one and highly separated among each other. The procedure aims therefore to find the smallest s_i value to succeed in a correct classification. To introduce the Silhouette coefficient as stop criterion within the algorithm (now renamed Silhouette MCSC), we have made the following modifications. Let K be the number of clusters and N the number of to-be-clustered images, by starting from $K=1$ (i.e. all the images in only one cluster), we iterate the clustering and the Silhouette coefficient computation up to $K=N$ (i.e. only one image in each cluster). The chosen clustering configuration is the one corresponding to the lowest Silhouette coefficient value. In Fig. 3, TPR for 100 tries on Set 1 and Set 2 is presented again (results achieved for Set 3 and Set 4 were similar).

As it is possible to note, the mean values of TPR are improved by introducing the Silhouette coefficient stop criterion: TPR becomes significantly higher (almost 30%). Furthermore, there is also a marked difference with respect to the trend of the original stop criterion, resulting in a reduced variability of values; this is also true for all the sets used in this analysis stage. In Table 2 the mean values of TPR and, for the sake of completeness, FPR for Sets 1–4 are reported. It is possible to note that introducing the Silhouette coefficient improves the performances in terms of TPR while FPR remains similar.

3. The proposed method

In recent years, spectral clustering has become one of the most popular modern clustering algorithms as it outperforms traditional clustering algorithms [23–25] in terms of computational efficiency and speed. These techniques make use of the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions. The previously presented Multi-Class Spectral Clustering technique (see Section 2.1), though it shows such welcomed features, evidences some drawbacks: firstly, it depends upon performance variability due to its random initialization and, secondly, it needs a criterion to select the best number of K clusters. To overcome such inconveniences, in this section we introduce and discuss another spectral clustering method, named *Normalized Cuts* method [20], which does not require to initially know the number of expected clusters (though it asks for a termination criterion itself) and does not present any randomness in performance.

3.1. The algorithm

Given a graph $G=(V,E)$, the edges E connecting each pair of nodes/images are weighted by means of a chosen similarity function $w(i,j)$, with i and j being two nodes of the graph. The graph G is partitioned into two disjoint graphs A and B ($A \bigcup B = V$ and $A \bigcap B = \varnothing$) by simply removing edges connecting the two parts. The total weight of the edges removed in the partition activity gives a computation of the degree of dissimilarity between these two parts: this is called the *cut* and is computed as

$$cut(A,B) = \sum_{u \in A, v \in B} w(u,v) \tag{2}$$

Table 2
The Basic MCSC algorithm and its variation with Silhouette coefficient: mean values of TPR and FPR over the 4 different sets.

Datasets	**Basic MCSC** (%)		**Silhouette MCSC** (%)	
	mean TPR	*mean FPR*	*mean TPR*	*mean FPR*
Set 1	53.76	0.1481	86.70	0.1367
Set 2	62.54	0.8093	86.25	0.2364
Set 3	59.12	0.0873	65.79	0.1717
Set 4	34.68	0.4881	73.67	1.5296

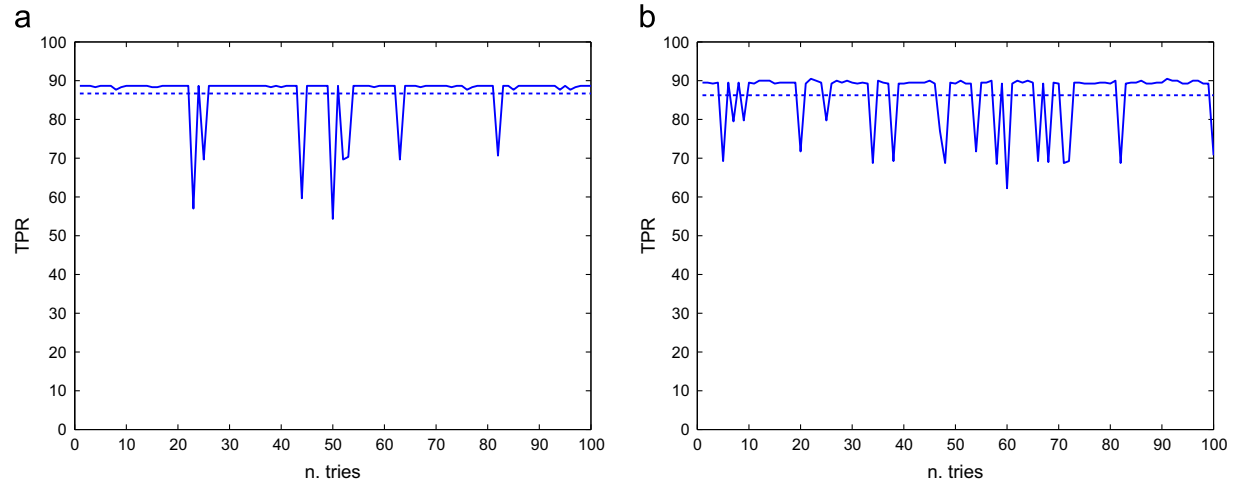

Fig. 3. MCSC algorithm with the Silhouette coefficient: TPR computed on 500 tries (for the sake of readability only 100 tries are reported) over 2 different sets: (a) Set 1 and (b) Set 2.

The optimal bipartition of a graph is obtained by means of the minimization of the *cut* value. However, it was noted that the minimum cut criteria favor cutting small sets of isolated nodes in the graph because the cut defined in Eq. (2) increases with the number of edges going across the two partitioned parts. To avoid such an unwanted behavior, in [20], the authors proposed a different disassociation measure, as a fraction of the total edge connections to all the nodes in the graph, called the *normalized cut* (*Ncut*) and defined as in the following:

$$Ncut(A,B)=\frac{cut(A,B)}{assoc(A,V)}+\frac{cut(A,B)}{assoc(B,V)} \tag{3}$$

where the *association* measure represents the total connections from nodes in A to all nodes in the graph (similarly for association of B) and it is defined as

$$assoc(A,V)=\sum_{u\in A,\ t\in V} w(u,t) \tag{4}$$

Consider the graph $G=(V,E)$ shown in Fig. 4, which is composed of two complete graphs L and R, with 6 nodes each, linked by two edges (a,c) and (b,d), where $a,b\in L$ and $c,d\in R$, and by a node x that does not belong to L and R and is linked to a and c. If we consider the partition of V into $A=L\cup R$ and $B=\{x\}$, then we have that $cut(A,B)=2$: moreover, any other partition would produce a cut value greater than 2. Hence, with respect to the cut value, this partition is optimal. If we consider, instead, the normalized cut value, since $assoc(A,V)=|E|$ and $assoc(B,V)=2$, we have that $Ncut(A,B)=2/|E|+\frac{2}{2}>1$. On the other hand, if we consider the partition of V into $A=L\cup\{x\}$ and $B=R$, then we have $cut(A,B)=3$, $assoc(A,V)=((|V|-1)/2)((|V|-1)/2-1)+4=34$ and $assoc(B,V)=((|V|-1)/2)(((|V|-12)/-1)+3=33$. Hence, in this case we have that $Ncut=\frac{3}{34}+\frac{3}{33}<1$, and this partition would be preferred. The example can be generalized by considering two complete graphs with n nodes: once again the normalized cut value of the first partition would be greater than 1, while the normalized cut value of the second partition would be $O(1/n^2)$, so that this latter partition would be preferred.

Though minimizing normalized cut exactly is NP-complete, an approximate discrete solution can be found efficiently in a real value domain. $G=(V,E)$ being a graph to be partitioned into two sets A and B; let $\mathbf{x}$ be an indicator vector of dimension $N=|V|$, where $x_i=1$ if the node i belongs to A and -1 otherwise, and let also $d(i)=\sum_j w(i,j)$ be the total connection from node i to all other nodes, and $\mathbf{D}$ an $N\times N$ diagonal matrix with $\mathbf{d}$ on its diagonal. On the basis of such assumptions, the problem of minimizing the normalized cut can be rewritten as (see [20] for more details)

$$\min_{\mathbf{x}} Ncut(\mathbf{x})=\min_{\mathbf{y}}\frac{\mathbf{y}^T(\mathbf{D}-\mathbf{W})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}} \tag{5}$$

where

$$\mathbf{y}=(\mathbf{1}+\mathbf{x})-b(\mathbf{1}-\mathbf{x})$$
$$b=\frac{\sum_{x_i>0}\mathbf{d}_i}{\sum_{x_i<0}\mathbf{d}_i}$$

If $\mathbf{y}$ is relaxed to take on real values, we can minimize Eq. (3) by solving the generalized eigenvalue system in the following equation:

$$(\mathbf{D}-\mathbf{W})\mathbf{y}=\lambda\mathbf{D}\mathbf{y} \tag{6}$$

In the end, the partitioning procedure can be summarized as follows:

1. Given a set of features, set up a weighted graph $G=(V,E)$, compute the weight on each edge which measures the similarity between two nodes.
2. Solve Eq. (6) for eigenvectors with the smallest eigenvalues.
3. Use the eigenvector with the second smallest eigenvalue to bipartition the graph by finding the splitting point such that *Ncut* is minimized.
4. Decide if the current partition should be subdivided recursively by checking the stability of the cut (see Section 3.2).

3.2. The threshold matter

The Normalized Cuts procedure is based on an iteration step (step 4) which depends on a stability check. In [20], such a check was made by comparing the *Ncut* value to a pre-specified threshold. In our implementation we have defined an *aggregation coefficient* (*AC*) (see Eq. (7)) which is computed for each one of the obtained clusters; the group corresponding to the lower *AC* value is split if such coefficient is under a pre-defined threshold T_h. The iteration stops when the aggregation coefficients of all the clusters are greater than the threshold T_h. We used, as aggregation coefficient, simply the mean value of the weights among nodes (N_k) belonging to that cluster. This kind of coefficient requires a low computational burden and has demonstrated a good effectiveness with respect to other more sophisticated measures:

$$AC(k)=\frac{1}{N_k}\sum_{i,j} w(i,j) \tag{7}$$

The definition of the threshold T_h is not a trivial task; we have used an approach based on ROC curves by taking as parameters of the correctness of a clustering the values of TPR and FPR with respect to a ground truth. To do that we have considered five diverse sets of images whose acquisition cameras were known. Four of these five sets were those already used for analyzing the Multi-Class Spectral Clustering algorithm (see Section 2.1 for their detailed composition), while the fifth set (Set 5), constituted by photos downloaded by *Flickr-photo sharing*, has been taken

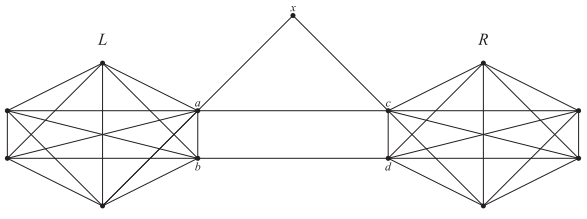

Fig. 4. An example of a graph in which a cluster with an isolated node is preferred.

as additional. In particular, such a fifth set was composed of images coming from five different devices of the same camera model, a Nikon Coolpix S210, and the set composition was the following:

Set 5: 110 images, 5 cameras, distribution 23, 20, 25, 24, 18 images per camera.

The variation range of the threshold T_h was set between 0.02 and 0.07, with a step of 0.001. We thus obtained, for each set, a ROC curve, as depicted in Fig. 5 (left).

By means of these ROC curves, for each set it is possible to find the optimal threshold value, as the one producing the (*TPR*,*FPR*) pair as close (in terms of minimum euclidean distance) as possible to the ideal pair $TPR=100\%$ and $FPR=0\%$. Table 3 (on the left) summarizes the obtained results. To determine the threshold T_h to be used in our experiments, we have averaged the values of TPR and FPR over the five sets for each threshold within the variation range, obtaining the ROC curve depicted in Fig. 5 (right). At this point, we have again computed which point achieves the minimum euclidean distance of the (*TPR*, *FPR*) from the ideal point at the top-left of the graph: the threshold value obtained is finally $T_h=0.037$, such a value will be adopted in the following experiments of the paper. According to this threshold, performances in terms of TPR and FPR for the five sets have been recalculated and listed in Table 3 (right) with respect to each best case.

4. Experimental results

In order to compare the proposed algorithm with other state-of-the-art methods, some experiments have been performed.

First of all, three new image sets, different from those (Set 1–Set 5) adopted so far, have been prepared. They have been named Set A, Set B and Set C, and are composed of some images randomly chosen from the previous sets used to tune-up the algorithms, and some other new ones. Test sets are described in detail in Table 4.

After that, the proposed method (*Normalized Cuts* with threshold $T_h=0.037$) has been compared, over these three test sets, with the basic MCSC technique [18] and with its variation the Silhouette-MCSC, as presented in Section 2.2; furthermore, another methodology, the *Hierarchical Agglomerative Clustering* (HAC), has been tested. The HAC is a clustering algorithm which has been proposed for blind image clustering in [17]; it resorts at a Silhouette coefficient as stop criterion achieving good performances. Before starting with experimental results, we would like to provide a hint on computational time of the various tested algorithms. With reference to Test Set A, for example, we can assess that the proposed method presents an execution time of around 0.18 s against 1.48 s of the Basic MCSC, while, when the Silhouette coefficient is introduced, MCSC takes about 82 s and the HAC around 41. The computations have been derived on DELL PRECISION T1500 desktop PC, equipped with an Intel Core i7 64 bit CPU @2.80 GHz, RAM 4 GB, in a Matlab R2010a environment.

This section, dedicated to the experimental results, is organized as follows: firstly, in Section 4.1, PRNU extraction and enhancement is explained and, consequently, the construction of the similarity matrix is debated, while a quantitative analysis of the performances, in terms of TPR and FPR, is given in Section 4.2 and in Section 4.3, some considerations, concerning the actual distribution of the images within each cluster, are provided.

Table 3
Values of (TPR,FPR) for each best case for each set (on the left) and for the selected threshold $T_h=0.037$ (on the right).

Datasets	T_h (best of)	**TPR** (%)	**FPR** (%)	T_h	**TPR** (%)	**FPR** (%)
Set 1	0.047	89.33	0.33	0.037	86.67	2.33
Set 2	0.046	95.75	0.18	0.037	85.25	1.96
Set 3	0.022	83.33	1.85	0.037	70.48	1.85
Set 4	0.029	89.92	1.04	0.037	73.08	1.04
Set 5	0.037	97.09	0.44	0.037	97.09	0.44

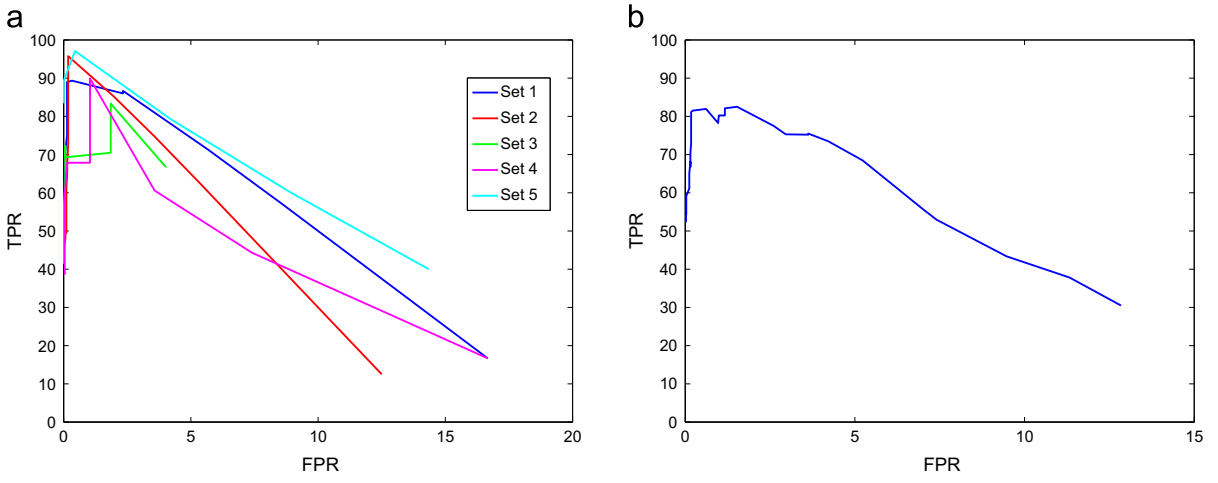

Fig. 5. ROC curves for the five considered sets (left) and ROC curve for mean values of TPR and FPR over the five sets (right).

4.1. PRNU extraction/enhancement and the similarity matrix

Photo Response Non-Uniformity (PRNU) noise is a sort of fingerprint which is typical of each kind of sensors; its presence is induced by intrinsic disconformities in the manufacturing process of silicon CCD/CMOSs and it can be extracted by means of digital filtering operations [17,9–12]. In our implementation, we have considered a wavelet-based denoising filter as in [17]. After that a suppression artifact procedure was applied to the obtained PRNU in order to remove some weak artifacts due to color interpolation in accordance with [9] and an enhancer is then adopted as defined in [17]. Such a procedure for the extraction of the noise image is applied to the image of the dataset (JPEG format at the maximum resolution of the camera). Once all the noise images have been generated, the similarity matrix has been created, computing the *normalized cross-correlation* between each noise image and all the others. The normalized cross-correlation is computed as

$$corr(n_i, n_j) = \frac{(n_i - \overline{n}_i) \cdot (n_j - \overline{n}_j)}{\| n_i - \overline{n}_i \| \cdot \| n_j - \overline{n}_j \|} \quad (8)$$

where n is the enhanced PRNU image, and the operator $\| \cdot \|$ stands for the L^2-norm operation, with i and j being the indexes of noise images belonging to each evaluated set ($i \neq j$), for all sets it is possible to build a *similarity matrix*, featuring all the distances (or similarities) between each pair of PRNU images.

Table 4
The three different test sets: the images are spread in 6, 5, and 4 sets.

Test set	No images	Brand and model	No images per camera
A	266	Canon EOS 400D DIGITAL	30
		Canon DIGITAL IXUS i zoom	50
		Panasonic DMC-FX12	30
		Panasonic DMC-LZ5	50
		Nikon COOLPIX S210	24
		Olympus Mju 1050SW	82
B	184	Canon EOS 400D DIGITAL	30
		Canon DIGITAL IXUS i zoom	50
		Panasonic DMC-FX12	30
		Panasonic DMC-LZ5	50
		Nikon COOLPIX S210	24
C	204	Canon DIGITAL IXUS 50	37
		FujiFilm FinePix F10	44
		NIKON D80	60
		Sony DSC-P200	63

4.2. Performance comparison

The proposed method is hereafter compared with the other three presented methods; experimental results are presented in terms of TPR and FPR. The proposed method (*Normalized Cuts* with threshold $T_h = 0.037$) has been compared with the Basic MCSC technique, with the Silhouette MCSC and the Hierarchical Agglomerative Clustering (HAC). In Table 5, the values of TPR and FPR for each test set (A, B and C) and for each method have been reported. It is to be noted that when dealing with the Multi-Class Spectral Clustering algorithm (Basic and Silhouette), the problem of randomization still persists (see Section 2.1); thus, several tries have been performed (500 for the Basic MCSC algorithm, 100 for Silhouette MCSC) and the averaged values over the tries (indicated with the subscript *av*) are the ones reported in Table 5 where the minimum and maximum values are reported as well. On the other side the HAC method and the proposed one do not present any variability at all, not depending on any random initialization. Looking at Table 5 and by referring to the averaged values, it is possible to observe that the proposed method outperforms the MCSC techniques (Basic and Silhouette), both in terms of TPR and FPR, though FPR of the proposed method is slightly higher for Set B and Set C. By considering the min/max values achieved by the MCSC (Basic and Silhouette), it can be appreciated that, also in a lucky try, the maximum TPR is always lower than the proposed one, except for Silhouette MCSC onto Set B. In comparison with Silhouette HAC, the proposed method presents a comparable TPR, though over Set C there is a significant improvement of about 22%, but a FPR that is basically smaller.

For a more complete analysis and a visual comparison, in Fig. 6 the trends of the TPR (left column) and FPR (right column) for each set are drawn; because of the randomness of two MCSC-based methods, results are reported over 100 tries and their average values, as presented in Table 5, have been drawn too. The blue line corresponds to the proposed method and the magenta represents the HAC (both lines are obviously constant onto the 100 tries), while the red one corresponds to the Basic MCSC and the green one to the Silhouette MCSC (both showing a high variability at the turn of the mean values). The considerations made when discussing Table 5 are visually confirmed by looking at Fig. 6. Fundamentally, the proposed method outperforms MCSC-based ones mainly in terms of TPR and provides a lower FPR than HAC while maintaining a comparable, slightly better, TPR.

Table 5
TPR and FPR values for each test set for the different methods.

(%)	**Proposed** $T_h = 0.037$		**Basic MCSC**		**Silhouette MCSC**		**Silhouette HAC**	
	TPR	*FPR*	TPR_{av} (min/max)	FPR_{av} (min/max)	TPR_{av} (min/max)	FPR_{av} (min/max)	*TPR*	*FPR*
Set A	79.45	0.22	45.42 (32.38/74.61)	0.24 (0.00/2.15)	63.58 (53.98/73.08)	1.00 (0.23/2.79)	79.85	2.68
Set B	81.50	3.99	38.66 (30.13/57.93)	0.11 (0.00/0.45)	70.57 (68.70/84.03)	0.27 (0.15/0.91)	79.20	4.72
Set C	92.32	1.56	37.27 (17.50/60.58)	0.14 (0.00/2.91)	55.04 (41.89/63.11)	0.97 (0.00/3.13)	70.07	9.52

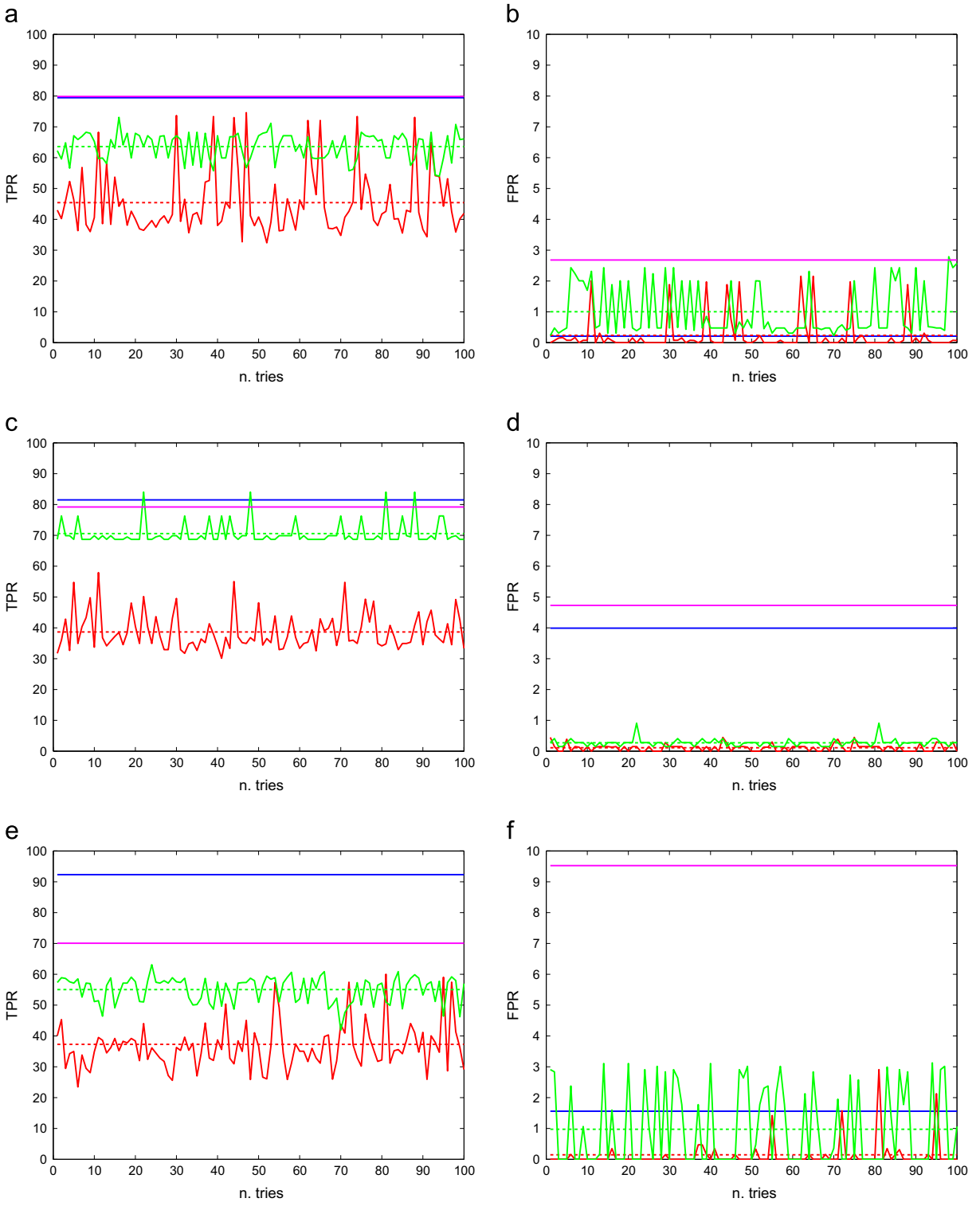

Fig. 6. TPR True Positive Rate (left column) and FPR False Positive Rate (right column): comparison among clustering methods for each test set. The proposed method in blue, basic MCSC in red, Silhouette MCSC in green and HAC in magenta: (a) TPR, Set A; (b) FPR, Set A; (c) TPR, Set B; (d) FPR, Set B; (e) TPR, Set C; and (f) FPR, Set C.

4.3. Qualitative analysis of clusters composition

For a better comprehension of the real behavior of the tested methods, and beyond the abrupt evaluation of TPR and FPR, a qualitative analysis concerning how the images are actually distributed in the obtained clusters is presented in this subsection. In Figs. 7, 8 and 9, the classification of images for Set A, Set B and Set C has been reported, respectively. For each method, on the x-axis the found clusters are shown by means of columns, whose height represents the cardinality of each cluster. According to the ground truth, images originally belonging to a certain camera (cluster) are of the same color, so different colors appear in the columns where false positives are occurred. For the sake of clarity, the ideal case ($TPR = 100\%$ and $FPR = 0\%$) would be represented by all single-colored columns whose number equals the number of cameras in the ground truth.

Looking at Fig. 7, it is immediate to appreciate that the proposed method (Fig. 7 (a)) and the HAC (Fig. 7 (d)) grant a reduced dispersion with respect to the MCSC-based techniques (Fig. 7 (b) and (c)). In this case (Set A, see composition in Table 4), the proposed method obtains 9 clusters (numbers 8 and 9 are quite small), instead of 6 as expected, and with a high degree of compactness; in fact, clusters colored in blue, dark blue and orange (Canon IXUS, Canon 400D and Nikon COOLPIX S210, respectively) are completely compact and do not present any of their images wrongly located in other clusters. Similarly, clusters in yellow and light blue (Panasonic DMC-LZ5 and Panasonic DMC-FX12, respectively) show a high level of compactness, in fact only few images have been dispersed; some images, originally belonging to the yellow cluster, have been grouped in a spurious group (number 9), while for the light blue one, some images have been associated with the yellow cluster (number 4). It is interesting to point out that such a cluster (number 4) is representative of a camera of the same brand but of a different model. On the contrary, the brown cluster (Olympus Mju 1050sw) is split into three sub-parts: the main one in the group number 1 and

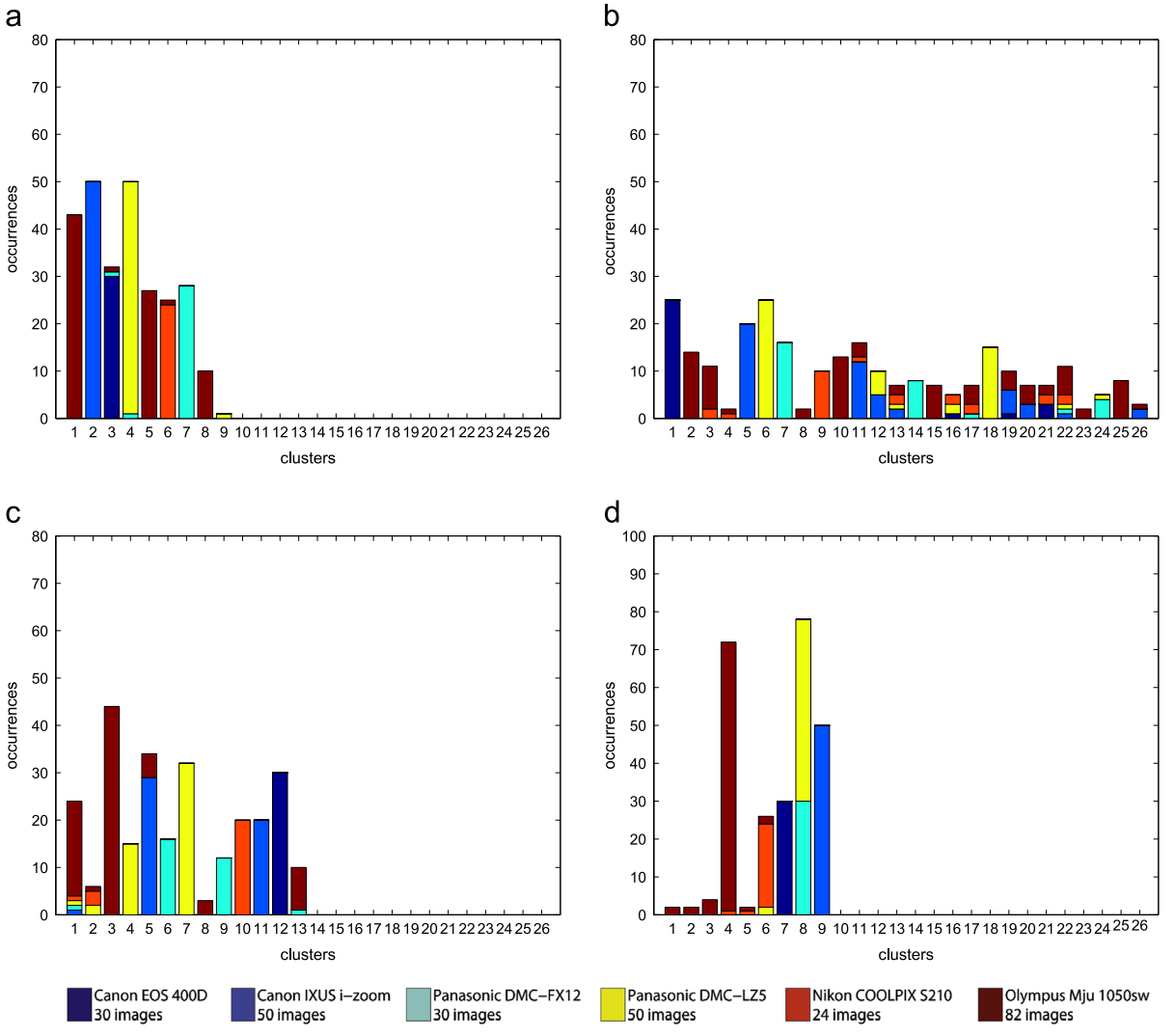

Fig. 7. Set A, comparison among the tested methods. Note that for MCSC-based methods, it has been reported the configuration showing the pair *TPR–FPR* nearest, in the euclidean sense, to the mean values pair $\overline{TPR}$–$\overline{FPR}$ (try n. 42 for Basic MCSC, [TPR, FPR]=[45.62, 0] and n. 49, [TPR, FPR]=[63.90, 0.75] for Silhouette MCSC): (a) proposed method, (b) basic MCSC, (c) Silhouette MCSC, and (d) Silhouette HAC.

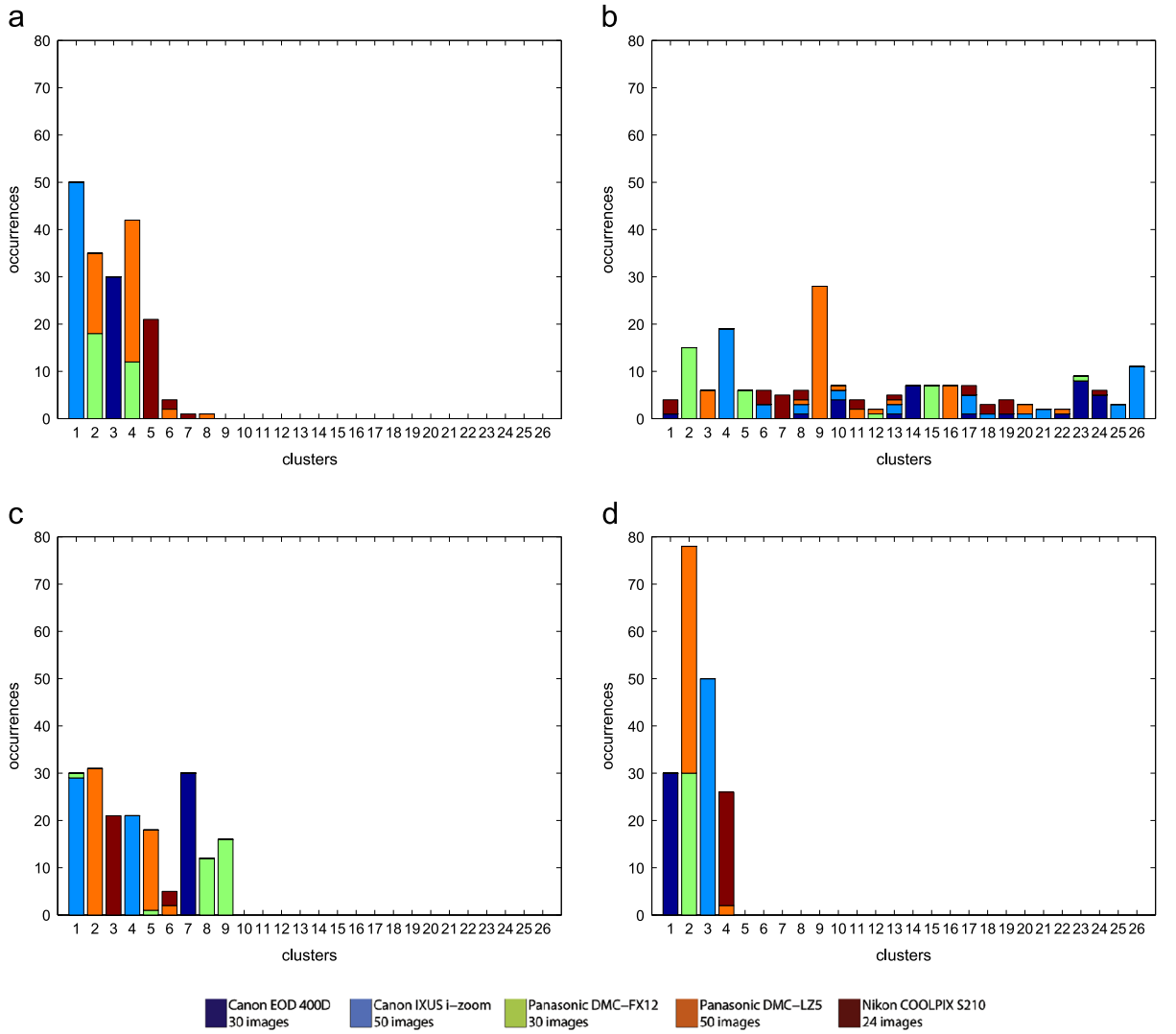

Fig. 8. Set B, comparison among the presented methods. Note that for MCSC-based methods, it has been reported the configuration showing the pair *TPR–FPR* nearest, in the euclidean sense, to the mean values pair $\overline{TPR}$–$\overline{FPR}$ (try n. 16 for Basic MCSC, [TPR, FPR]=[38.53, 0.13] and n. 3, [TPR, FPR]=[69.90, 0.15] for Silhouette MCSC): (a) proposed method, (b) basic MCSC, (c) Silhouette MCSC, and (d) Silhouette HAC.

two others (numbers 6 and 8) which, though separated, are not however inserted within other clusters. The brown clusters present, also for the other methods, a large level of dispersion which could be due to the intrinsic specific characteristics of the images.

It is also interesting to highlight how the introduction of the Silhouette coefficient in MCSC method reduces the spreading of images (see Fig. 7(b) and (c)) both in terms of creating new clusters (clusters are halved from 26 to 13) and of assigning some images to different cameras (e.g. the brown cluster is distributed over 16 in Fig. 7(b) and then only over 6 in Fig. 7(c)). On the contrary, the Hierarchical Agglomerative Clustering tends to produce a lower number of classes, thus aggregating sets of images coming from different cameras. It is possible to say that HAC prefers the aggregation of images, instead of the separation. This is also confirmed by the values of FPR, which are higher, on average, for HAC method than for the others.

Results presented in Figs. 8 and 9 globally confirm, for each one of the methods, the behavior shown in Fig. 7 for the other two test sets B and C. From a careful observation of Figs. 7–9, it can be deduced that the proposed method seems to offer an effective trade-off between the tendency of the MCSC-based methods to over-split images into too many clusters and the trend of HAC to join images actually belonging to different classes.

5. Conclusion and future works

In this paper a novel methodology based on Normalized Cuts criterion to perform blind image clustering has been introduced and evaluated in comparison with other state-of-the-art techniques, achieving results that are better in terms of accuracy and computational complexity. Such a methodology resorts at a stop threshold to freeze the clustering procedure but does not need any additional information

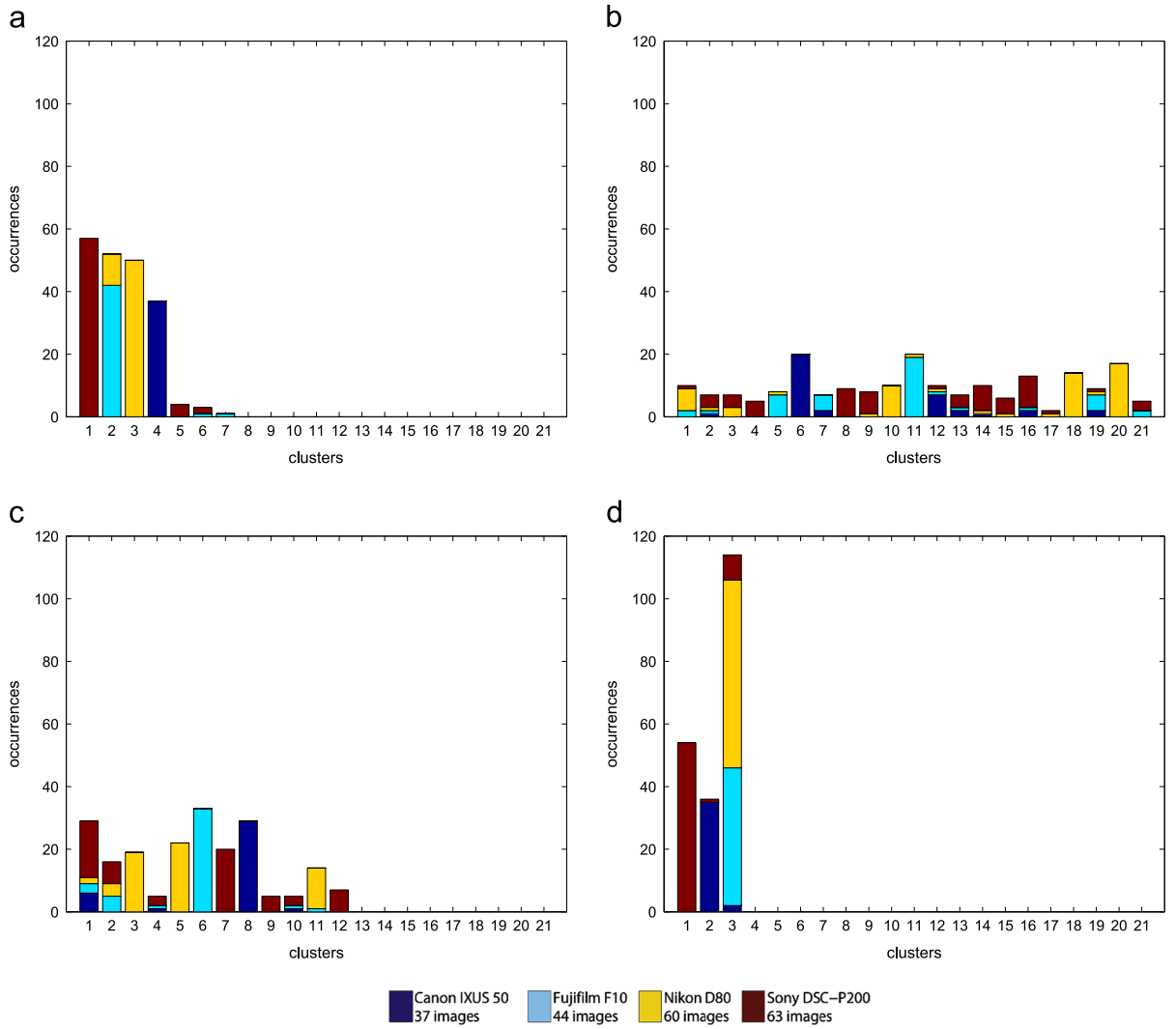

Fig. 9. Set C, comparison among the presented methods. Note that for MCSC-based methods, it has been reported the configuration showing the pair *TPR*–*FPR* nearest, in the euclidean sense, to the mean values pair $\overline{TPR}$–$\overline{FPR}$ (try n. 344 for Basic MCSC, [TPR, FPR]=[37.23, 0.16] and n. 64, [TPR, FPR]=[55.44, 0] for Silhouette MCSC): (a) proposed method, (b) basic MCSC, (c) Silhouette MCSC, and (d) Silhouette HAC.

concerning the number of expected clusters, so it offers a good instrument to work in this kind of application scenario. Future works will be dedicated to individuate an adaptive stop criterion which takes into account the actual characteristics of the clustering stage. Another possible future development is related to the analysis of the performances of the proposed technique when fingerprint digests are adopted to deal with a significant number of cameras and images. In this case, it could be interesting to also take into account the adoption of techniques for parallel spectral clustering [26].

Acknowledgments

This work was partially supported by the SECURE! Project, funded by the POR CreO FESR 2007–2013 programme of the Tuscany Region (Italy) and by the AMANDA (Algorithmics for MAssive and Networked DAta) Project under PRIN 2012C4E3KT national research programme of the Italian Ministry of Education, University, and Research (MIUR).

Appendix A. TPR and FPR definition

Clustering procedures have been evaluated and compared by means of quantitative measures such as the *True Positive Ratio* (*TPR*) and the *False Positive Ratio* (*FPR*). These ratios can be computed with respect to a known *ground-truth classification* (images acquired by each camera); both the TPR and the FPR are calculated separately for each obtained cluster and then averaged on the number of ground-truth clusters, that is by considering only the number of expected clusters. More in detail, TPR is a measure stating how many images are allocated in the

"correct" cluster, with respect to the expected cardinality of the cluster; similarly, FPR states for how many images are erroneously assigned to a cluster with respect to the number of images actually not belonging to it. In formulae, TPR and FPR can be computed as

$$TPR = \frac{TP}{TP+FN}$$
$$FPR = \frac{FP}{FP+TN} \quad \text{(A.1)}$$

In previous equation (A.1), the used notations have the following meaning:

- TP: True Positives, images assigned to a cluster that actually belong to that cluster.
- FN: False Negatives, images belonging to a cluster, that have not been assigned to that cluster.
- FP: False Positives, images assigned to a cluster that actually belong to a different one.
- TN: True Negatives, images not assigned to a cluster that actually belong to different ones.
- $TP+FN$: The sum of True Positives and False Negatives represents the measure of how many images actually belong to a cluster, that is the cardinality of the cluster itself, known by the ground-truth.
- $FP+TN$: The sum of False Positives and True Negatives represents the measure of how many images actually belong to all the other clusters except the evaluated one.

TPR and FPR are assessed in percentage; the higher the TPR and the lower the FPR, the better the clustering. The ideal case is the one showing $TPR = 100\%$ and $FPR = 0\%$, asserting that all and only the images actually belonging to a cluster have been assigned to that cluster.

References

[1] J.A. Redi, W. Taktak, J.-L. Dugelay, Digital image forensics: a booklet for beginners, Multimed. Tools Appl. 51 (1) (2011) 133–162, http://dx.doi.org/10.1007/s11042-010-0620-1.

[2] S. Lyu, H. Farid, How realistic is photorealistic? IEEE Trans. Signal Process. 53 (2) (2005) 845–850.

[3] N. Khanna, G.T.-C. Chiu, J.P. Allebach, E.J. Delp, Forensic techniques for classifying scanner, computer generated and digital camera images, in: IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP 2008, 2008, pp. 1653–1656. http://dx.doi.org/10.1109/ICASSP.2008.4517944.

[4] C. McKay, A. Swaminathan, G. Hongmei, M. Wu, Image acquisition forensics: Forensic analysis to identify imaging source, in: IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP 2008, 2008, pp. 1657–1660. http://dx.doi.org/10.1109/ICASSP.2008.4517945.

[5] R. Caldelli, I. Amerini, F. Picchioni, A DFT-based analysis to discern between camera and scanned images, Int. J. Digit. Crime Forensics (IJDCF), e-Forensics 2009 Special Edition 2 (1) (2010).

[6] N. Khanna, A. Mikkilineni, G.T.-C. Chiu, J.P. Allebach, E.J. Delp, Scanner identification using sensor pattern noise, in: Proceedings of SPIE—Storage and Retrieval for Image and Video Databases, vol. 6505, 65051K, 2007.

[7] H. Gou, A. Swaminathan, M. Wu, Robust scanner identification based on noise features, in: Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX, Vol. 6505, 65050S, 2007.

[8] J. Lukás, J. Fridrich, M. Goljan, Digital camera identification from sensor pattern noise, IEEE Trans. Inf. Forensics Secur. 1 (2) (2006) 205–214.

[9] M. Chen, J. Fridrich, M. Goljan, J. Lukas, Determining image origin and integrity using sensor noise, IEEE Trans. Inf. Forensics Secur. 3 (1) (2008) 74–90.

[10] J. Fridrich, Digital image forensic using sensor noise, IEEE Signal Process. Mag. 26 (2) (2009) 26–37.

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

[12] N. Mondaini, R. Caldelli, A. Piva, V. Cappellini, M. Barni, Detection of malevolent changes in digital video for forensic applications, in: Proceedings of SPIE—Security, Steganography, and Watermarking of Multimedia Contents IX, 2007, vol. 6505, 2007, p. 65050T.

[13] M. Goljan, J. Fridrich, T. Filler, Managing a large database of camera fingerprints, in: Proceedings of SPIE, vol. 7541, 2010, pp. 754108-1–754108-12. http://dx.doi.org/10.1117/12.838378.

[14] M. Goljan, J. Fridrich, Sensor fingerprint digests for fast camera identification from geometrically distorted images, in: Proceedings of SPIE, vol. 8665, 2013, pp. 86650B-1–86650B-10. http://dx.doi.org/10.1117/12.2003234.

[15] F. Costa, M. Eckmann, W. Scheirer, A. Rocha, Open set source camera attribution, in: 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2012, pp. 71–78. http://dx.doi.org/10.1109/SIBGRAPI.2012.19.

[16] C.-T. Li, Unsupervised classification of digital images using enhanced sensor pattern noise, in: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS'10), 2010, pp. 3429–3432.

[17] R. Caldelli, I. Amerini, F. Picchioni, M. Innocenti, Fast image clustering of unknown source images, in: IEEE International Workshop on Information Forensics and Security (WIFS), 2010, pp. 1–5. http://dx.doi.org/10.1109/WIFS.2010.5711454.

[18] B.-B. Liu, H.-K. Lee, Y. Hu, C.-H. Choi, On classification of source cameras: a graph based approach, in: IEEE International Workshop on Information Forensics and Security (WIFS), 2010, pp. 1–5. http://dx.doi.org/10.1109/WIFS.2010.5711446.

[19] S. Luan, X. Kong, B. Wang, Y. Guo, X. You, Silhouette coefficient based approach on cell-phone classification for unknown source images, in: IEEE International Conference on Communications (ICC), 2012, pp. 6744–6747. http://dx.doi.org/10.1109/ICC.2012.6364928.

[20] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.

[21] A.P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognit. 30 (7) (1997) 1145–1159. http://dx.doi.org/10.1016/S0031-3203(96)00142-2.

[22] S.X. Yu, J. Shi, Multiclass spectral clustering, in: Proceedings of Ninth IEEE International Conference on Computer Vision, 2003 1 (2003) pp. 313–319.

[23] R. Kannan, S. Vempala, A. Vetta, On clusterings: good, bad and spectral, J. ACM 51 (3) (2004) 497–515.

[24] U. von Luxburg, M. Belkin, O. Bousquet, Consistency of spectral clustering, Ann. Stat. 36 (2) (2008) 555–586.

[25] U. Scaiella, P. Ferragina, A. Marino, M. Ciaramita, Topical clustering of search results, in: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, WSDM '12, ACM, New York, NY, USA, 2012, pp. 223–232. http://dx.doi.org/10.1145/2124295.2124324.

[26] Y. Song, W. Chen, H. Bai, C.-J. Lin, E. Y. Chang, Parallel spectral clustering, in: ECML/PKDD (2), 2008, pp. 374–389.