NEAR-DUPLICATE VIDEO DETECTION EXPLOITING NOISE RESIDUAL TRACES

Silvia Lameri, Luca Bondi, Paolo Bestagini, Stefano Tubaro

Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

ABSTRACT

Video phylogeny research about joint analysis of correlated video sequences has shown the possibility of developing interesting forensic applications. As an example, it is possible to study the provenance of near-duplicate (ND) video sequences, i.e., videos generated from the same original one through content preserving transformations. To perform this kind of analysis, accurate detection of ND videos is paramount. In this paper, we propose an algorithm for ND video detection and clustering in a challenging setup. Specifically, we analyze a scenario in which many videos, depicting the same event, are recorded by different users. This situation is critical as non-ND videos acquired from very close viewpoints run the risk of being incorrectly detected as ND. The proposed approach leverages on robust hashing properties and the concept of sensor noise traces.

***Index Terms*—** near-duplicate videos, semantically similar videos, video forensics, video phylogeny

1. INTRODUCTION

The forensic community has developed a wide set of algorithms to blindly reconstruct the history of a video sequence based on single video analysis [1, 2, 3, 4]. Additionally, in the last few years, researchers have also shown the possibility of performing interesting forensic analysis by jointly studying and processing multiple near-duplicate (ND) video objects, i.e., sequences obtained applying a set of content preserving transformations to the same original content [5, 6, 7]. As an example, when multiple ND copies of the same video of sensitive nature are diffused online, it is possible to help investigators pointing out which user first posted the original content. This can be done by reconstructing the video phylogeny tree, i.e., a directed graph summarizing the parental relationships among ND videos [8, 9]. Additionally, exploiting ND analysis, it is possible to help a forensic investigator to increase the media coverage about an important event of interest (e.g., to gain more information about a crime scene [10]).

A fundamental step in this kind of algorithms is the accurate collection of ND video sequences to analyze. To this purpose, many algorithms robust against video transformations have been proposed in the video search and retrieval literature [11, 12]. However, in a forensic scenario, it is not uncommon that analysts investigate public events of interest (e.g., public speeches, criminal attacks, etc.).

This material is based on research sponsored by DARPA and Air Force Research Laboratory (AFRL) under agreement number FA8750-16-2-0173. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and Air Force Research Laboratory (AFRL) or the U.S. Government.

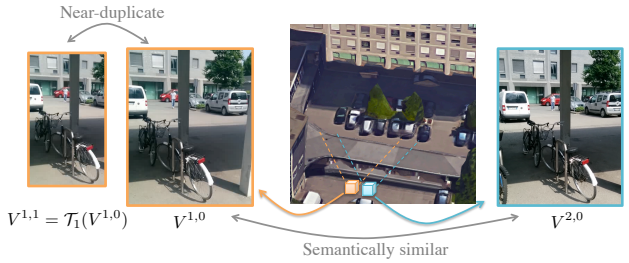


Fig. 1: Example of ND and SSI videos. A scene is captured simultaneously with two devices, originating $V^{1,0}$ and $V^{2,0}$. Video $V^{1,0}$ is further edited to obtain a ND video $V^{1,1}$. In this scenario it is difficult to distinguish videos coming from different devices.

These are often simultaneously documented by many people using their own capturing devices (e.g., smartphones) from different viewpoints, thus giving rise to semantically similar (SSI) videos in addition to NDs (see Fig. 1). Algorithms tailored to ND video detection that do not take this possibility into account may incorrectly recognize as NDs also SSI videos acquired by different users from particularly close points of view. As an example, the authors of [13] showed that the hashing-based algorithm for ND detection discussed in [10] can be used to detect also SSI video sequences. This is undesirable for video phylogeny algorithms that strictly process ND videos and do not deal with SSI ones [7, 8, 9].

In order to solve the aforementioned problem, in this paper we propose a ND video detection algorithm, which is robust against SSI sequences even if coming from very close viewpoints. Given a pool of videos under analysis, the goal is to blindly cluster groups of NDs, not containing spurious SSI sequences. The proposed method is based on two founding concepts: (i) robust video hashing [9, 12] to perform a first rough and fast screening; (ii) sensor noise traces left on video sequences by the capturing device [14, 15], for detection refinement.

The rationale behind the proposed method is that ND videos are all generated from the same original sequence, thus they come from the same acquisition device. Conversely, SSI videos recorded simultaneously are by definition acquired from different cameras. Therefore, after we group SSI videos based on semantical content using robust hashing techniques, we can resort to traces characterizing different devices to recognize ND families. Specifically, as it is well known that each device leaves on captured videos peculiar footprints that can be exposed by denoising operations [16, 14, 15], we rely on this concept for the refinement step.

The validation campaign is performed on a set of more than 12 000 videos among ND and SSI ones coming from seven different devices. Experiments show that the proposed approach is capable of reaching more than 98% of accuracy in ND detection, thus greatly

improving over the baseline ND detection method [10] based only on robust hashing. We also tested the proposed algorithm on a set of videos gathered online.

The rest of the paper is structured as follows. Section 2 presents ND detection problem formulation. Section 3 reports a detailed description of the proposed algorithm. Section 4 presents the experimental results. Finally, Section 5 concludes the paper providing some additional remarks.

2. PROBLEM FORMULATION

In everyday scenarios, events of interest (e.g., public speeches, criminal attacks, but also concerts and sport events) are often documented by different users that simultaneously capture the scene with their own devices from multiple viewpoints. In this context, let us define $\{V^{p,0}\}_{p\in[1,P]}$ a set of P original videos, each one acquired by the p-th device (i.e., from the p-th point of view).

If these videos are shared, other users can edit and re-distribute their own copies, thus creating other J_p versions $V^{p,j} = \mathcal{T}_j(V^{p,0})$, $j \in [1, J_p]$ of each original content $V^{p,0}$, where $\mathcal{T}_j$ is any content preserving transformation (e.g., colour enhancement, logo insertion, resizing, frame cropping, or combinations of them). As an example, this happens every time newscasts broadcast the same interview at different resolutions with different overlay text and superimposed logos.

In this scenario, all videos $V^{p,j},\ p \in [1, P],\ j \in [0, J_p]$ are defined *semantically similar* (SSI) sequences [13], since they all capture the same semantic information about a scene or an event. In particular, videos $V^{p,j},\ j \in [0, J_p]$ for a fixed p value are denoted as strictly *near-duplicate* (ND) sequences [7, 8, 10]. Fig. 1 shows an example of the ND and SSI videos generative process.

Given a generic set of video sequences, ND video detection refers to the problem of correctly clustering separate groups of strictly ND videos. Solving this problem is paramount for video phylogeny tools designed to jointly analyze multiple versions of the same video object [7, 8], as they work under the hypothesis that only NDs are present within the analysis pool. For this reason, some effective ND detection solutions have been proposed in the literature [11, 12]. However, when the set of considered video sequences contains SSI videos shot from very close viewpoints (as those shown in Fig. 1), ND detection problem turns out to be more challenging. Indeed, solutions as the one proposed in [10] tend to cluster SSI videos as if they were NDs [13].

In this work we propose an algorithm for ND video detection in this challenging scenario. Specifically, our goal is to separate actual ND sequences from SSI sequences captured from different viewpoints, even if very close to each other. Formally, given a set of videos $\{V^{p,j}\}_{p\in[1,P],\, j\in[0,J_p]}$, we aim at individuating: (i) the number P of ND clusters; (ii) the composition of each cluster $C_p = \{V^{p,j}\}_{j\in[0,J_p]},\ p \in [1, P]$, each one containing only ND videos generated from the original sequence $V^{p,0}$.

3. NEAR DUPLICATE VIDEO DETECTION

In this section we present the proposed algorithm for ND video detection starting from a generic set of SSI videos relative to the same event $\mathcal{V} = \{V^{p,j}\}_{p\in[1,P],\, j\in[0,J_p]}$. For notation simplicity, from now on, a generic video in $\mathcal{V}$ will be denoted as V_i, univocally mapping the pair of indexes (p, j) into a single index i.

The proposed algorithm works by comparing pairs of videos (V_{i_1}, V_{i_2}) in $\mathcal{V}$ separately in three stages, as shown in Fig. 2. When

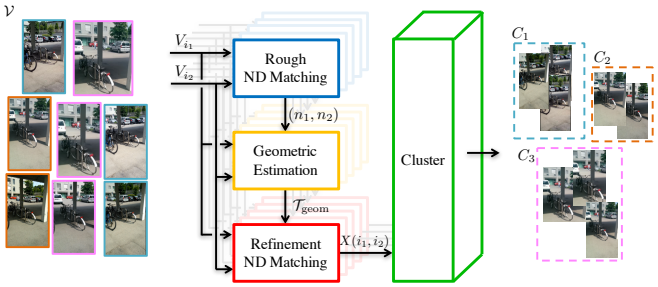


Fig. 2: Pipeline of the proposed method. A pool of SSI videos is analyzed, and separate clusters of NDs are identified.

all video pairs in $\mathcal{V}$ have been compared to estimate how likely they are ND, a clustering algorithm is applied to group separate sets of ND video sequences. In the following a detailed description of each step.

Rough ND matching. The first step of the algorithm consists in performing a rough and fast ND detection based on the robust hashing method proposed in [9] to determine whether sequences V_{i_1} and V_{i_2} are ND candidate. Indeed, if the two videos happen to be ND or SSI depicting the same time instant from closed viewpoints, the algorithm in [9] returns the sets of corresponding frames of V_{i_1} and V_{i_2} that are temporally synchronized. Depending on the output of [9], we fill a logical near-duplicate relationship matrix defined as

$$M(i_1, i_2) = \begin{cases} 1, & \text{if synchronized frames are found,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

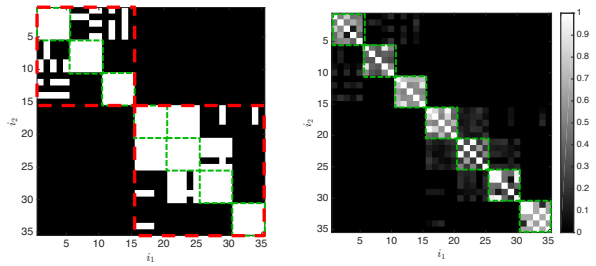
If $M(i_1, i_2) = 1$, we detect videos V_{i_1} and V_{i_2} as candidate ND and apply the following refinement steps of the algorithm. If $M(i_1, i_2) = 0$, we proceed analyzing the next video pair.

Note that M can be interpreted as a binary adjacency matrix representing a undirected graph, whose nodes represent videos, and video pairs (V_{i_1}, V_{i_2}) for which $M(i_1, i_2) = 1$ are linked by an edge. As a matter of fact, the ND detection algorithm proposed in [10] is equivalent to running a Depth-First Search (DFS) algorithm on M to find the connected components of the graph. However, in presence of SSI videos, this simple strategy is not sufficient. An example of M computed on a set of 35 videos split into 7 SSI groups of 5 NDs each, is shown in Fig. 3(a). Correct clusters of NDs are highlighted in green, whereas connected components identified by DFS are reported in red. This motivate our further analysis.

Geometric estimation. If V_{i_1} and V_{i_2} are detected as ND candidate (i.e., $M(i_1, i_2) = 1$), we need to estimate the geometric transformations $\mathcal{T}_{\text{geom}}$ that maps V_{i_1} into V_{i_2} in terms of resize and crop, before further proceeding with noise analysis. To do so, we select a pair of matching frames $(V_{i_1}(s_1), V_{i_2}(s_2))$ returned by [9], and estimate the homography and crop transformation $\mathcal{T}_{\text{geom}}$ between them by matching points of interest using Speeded-Up Robust Features (SURF) [17] and applying RANSAC [18] (see Fig. 4).

Note that we estimate $\mathcal{T}_{\text{geom}}$ only once for each video pair, assuming that the geometric transformation mapping V_{i_1} into V_{i_2} is exactly the same for all the sequences' frames, which is typical for ND videos. However, in order to increase algorithm's robustness, it is possible to compute a different transformation $\mathcal{T}_{\text{geom}}$ for each pair of frames returned by [9].

Refinement ND matching. Given the video pair V_{i_1} and V_{i_2} and the geometrical transformation $\mathcal{T}_{\text{geom}}$, the rationale of the refinement step is to leverage camera fingerprint to assess whether the two



(a) $M(i_1, i_2)$ (b) $X(i_1, i_2)$

Fig. 3: Matrices M and X computed on $\mathcal{V}$ composed by 35 videos split into 7 SSI clusters of 5 ND sequences each depicting the scene represented in Fig. 1. Only clustering on X leads to the correct results (in green), whereas considering M, sequences are under-clustered (in red).

potential ND videos have been actually acquired by the same device (i.e., they really are NDs). To this purpose, we estimate sequences' fingerprints $\hat{K}_{i_1}$ and $\hat{K}_{i_2}$ by aggregating noises extracted from a set of S frames as typically done in camera fingerprinting works [14, 16]. This can be done also in the ND scenario as camera traces are not deleted by cropping or resizing [15, 19].

Formally, the noise residual of each frame $V(s)$ of a video is defined as $W(s) = V(s) - \mathcal{D}(V(s))$, where $\mathcal{D}$ is a denoising operator. Given a set of frames $V(s),\ s \in [1, S]$ from the same camera, the fingerprint can be computed as

$$\hat{K} = \frac{\sum_{s=1}^{S} W(s)V(s)}{\sum_{s=1}^{S} V(s)^2}. \tag{2}$$

Note that, as we are considering ND videos, they may have different resolutions due to the processing operations they underwent, e.g, resize and cropping. Therefore, before comparing two different fingerprints $\hat{K}_{i_1}$ and $\hat{K}_{i_2}$ it is necessary to geometrically register them. To this purpose, we apply to $\hat{K}_{i_1}$ the estimated transformation $\mathcal{T}_{\text{geom}}$, thus obtaining $\hat{K}_{i_1 \to i_2} = \mathcal{T}_{\text{geom}}(\hat{K}_{i_1})$, i.e, the warped version of $\hat{K}_{i_1}$ into $\hat{K}_{i_2}$. We compare the registered noise fingerprints by means of the normalized cross-correlation (NCC) $\rho(\hat{K}_{i_1 \to i_2}, \hat{K}_{i_2})$, and store this value in a cross-correlation matrix X defined as

$$X(i_1, i_2) = \rho(\hat{K}_{i_1 \to i_2}, \hat{K}_{i_2}). \tag{3}$$

The higher the $X(i_1, i_2)$ value, the higher the probability of V_{i_1} and V_{i_2} coming from the same device (i.e., are NDs). Conversely, low $X(i_1, i_2)$ values are expected for sequences coming from different devices (i.e., SSI videos). An example is reported in Fig. 3(b).

Note that this registration step is paramount for two reasons. First, it compensates for ND geometrical transformations allowing us to correctly match NDs as explained. Second, if we compare two videos from the same camera that are not NDs (i.e., they are not obtained from the same original video) it allows us to correctly detect them as non NDs. Indeed, in this scenario, the estimated $\mathcal{T}_{\text{geom}}$ would be a meaningless transformation leading to fingerprint desynchronization. Therefore, noise traces would not match, and we would not incorrectly detect as NDs videos that actually are not.

Finally, it is also important to notice that we do not really need to estimate a clean camera fingerprint for our goal. Indeed, if some scene content leaks into the estimated fingerprint, it helps us in matching ND videos through correlation. Therefore, even if in principle many video frames are needed to correctly estimate the camera noise for attribution problems, in our scenario we can simply exploit a reduced set of frames.

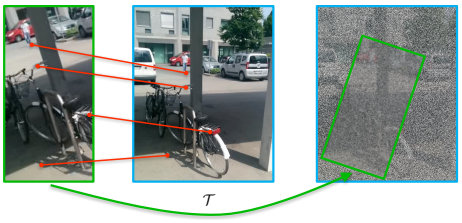


Fig. 4: Keypoint detection and matching for $\mathcal{T}_{\text{geom}}$ estimation between two ND frames.

Clustering. Once the comparison between all pairs of candidate ND videos are carried out, we run a clustering algorithm on rows of matrix X. This step returns the separate sets of ND videos $C_p,\ p \in [1, P]$.

4. RESULTS

In this section we first describe the datasets used for the experimental campaign, then we report the achieved results.
Supplemental results on some videos gathered online can be found at `https://tinyurl.com/y9gacunr`.

Datasets. We acquired a set of SSI sequences simulating a scenario in which multiple users simultaneously take videos of the same scene. Then, NDs were generated from each SSI video through editing transformations. We opted for this strategy instead of using any common multiview video dataset available in the literature, as these datasets are usually acquired with high-end devices and cameras movements are constrained, thus leading to less realistic results.

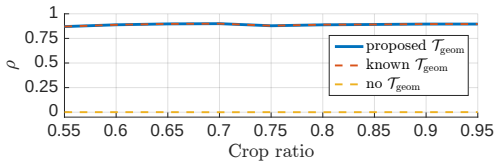
For the SSI generation process, 7 users with 7 handheld devices acquired 9 scenes (between 15 s and 40 s each) of different nature (e.g., indoor, outdoor, moving objects, buildings, people, repeating patterns, etc.) from very close viewpoints. To be more realistic, users were free to pan or slightly rotate, given that each camera was centered on the scene of interest. Videos were not temporally synchronized. This process led to 9 families of 7 original SSI videos each, which were resized to a common 640×360 resolution. Starting from these 63 original sequences, we built different datasets to test the proposed algorithm (see Table 1).

To evaluate the effect of $\mathcal{T}_{\text{geom}}$ estimation and its application to PRNUs, we built $\mathcal{V}_{\text{res}}$ and $\mathcal{V}_{\text{crop}}$ composed by 1 386 videos. Each one contains 63 sets (i.e., 9 scenes for 7 devices) of 11 sequences: an original video plus 10 ND copies obtained through resizing ($\mathcal{V}_{\text{res}}$) or cropping ($\mathcal{V}_{\text{crop}}$) to a resolution that ranges from 95% to 55% of the original one.

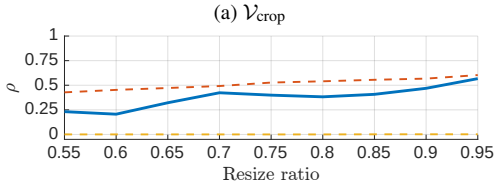
To evaluate the overall proposed pipeline, we generated six additional datasets $\mathcal{V}^p_{\text{ND}},\ p \in [2, 7]$ for a total amount of 12 150 videos. Each $\mathcal{V}^p_{\text{ND}}$ contains 90 sets (i.e., 9 scenes for 10 random ND realizations) of p clusters of 5 ND videos. For ND generation, we

Table 1: Each dataset is composed by different video sets, characterized by different ND clusters and transformations.

Dataset	Sets (scene x realiz.)	Videos	Clusters	Transf	Tot
$\mathcal{V}_{\text{res}}$	63 (9x7)	11	1	resize	693
$\mathcal{V}_{\text{crop}}$	63 (9x7)	11	1	crop	693
$\mathcal{V}^2_{\text{ND}}$	90 (9x10)	10	2	any	900
$\mathcal{V}^3_{\text{ND}}$	90 (9x10)	15	3	any	1350
$\mathcal{V}^4_{\text{ND}}$	90 (9x10)	20	4	any	1800
$\mathcal{V}^5_{\text{ND}}$	90 (9x10)	25	5	any	2250
$\mathcal{V}^6_{\text{ND}}$	90 (9x10)	30	6	any	2700
$\mathcal{V}^7_{\text{ND}}$	90 (9x10)	35	7	any	3150



(a) $\mathcal{V}_{\text{crop}}$



(b) $\mathcal{V}_{\text{res}}$

Fig. 5: NCC obtained on ND videos in $\mathcal{V}_{\text{crop}}$ and $\mathcal{V}_{\text{res}}$ at different resolutions applying the proposed pipeline. If geometric transformations are not compensated (yellow line), NCC tends to zero.

considered the following transformations [9]: contrast enhancement, brightness adjustment, spatial cropping and resizing, in any combination. Each transformation was followed by compression with a random codec (MPEG-2, MPEG-4 Part 4, H.264/AVC), group of picture (1 to 15) and quality parameter (1 to 10).

Geometric transformation. First, we evaluated the performance of fingerprint registration. We know from [15, 19] that fingerprints survives ND editing steps, but we need to validate the reliability of using $\mathcal{T}_{\text{geom}}$ to compensate for fingerprints geometric transformations.

To this purpose, for each set of videos in $\mathcal{V}_{\text{res}}$ and $\mathcal{V}_{\text{crop}}$, we analyzed the 10 video pairs $(V_i,\ V_0)$, $i \in [1, 10]$, where V_0 is the original SSI of the set and V_i is a resized or cropped version. For each pair, we estimated the fingerprints $\hat{K}_i,\ \hat{K}_0$ aggregating the residuals extracted from the first $S = 20$ frames of each video. We computed both $\hat{K}_{i\to 0}$ and $\hat{K}_{0\to i}$ by means of $\mathcal{T}_{\text{geom}}$ estimated from V_i and V_0. Evaluation is carried out by computing normalized cross-correlations (NCCs) between registered fingerprints. The higher the NCC, the better the geometrical registration.

Figs. 5a and 5b show NCC values at different resolutions obtained on $\mathcal{V}_{\text{crop}}$ and $\mathcal{V}_{\text{res}}$, respectively. Solid blue lines represent the average between NCC values $\rho(\hat{K}_{i\to 0}, \hat{K}_0)$ and $\rho(\hat{K}_{0\to i}, \hat{K}_i)$ (i.e. the proposed approach). Yellow dashed lines represent NCC values $\rho(\hat{K}_{i_1}, \hat{K}_{i_2})$ obtained without spatially synchronizing fingerprints. Dashed orange lines represent the NCC upper bounds obtained when $\mathcal{T}_{\text{geom}}$ is known a priori. These results confirm the importance of estimating $\mathcal{T}_{\text{geom}}$. Indeed, if $\mathcal{T}_{\text{geom}}$ is not applied (yellow line), NCCs tend to zero (i.e., we cannot recognize videos from the same device). Conversely, the proposed approach (blue line) enables to achieve high NCC values that allow to detect ND sequences.

Overall pipeline. We applied the proposed pipeline to every set of videos in datasets $\mathcal{V}^p_{\text{ND}},\ p \in [2, 7]$. Each set is composed by p clusters of ND videos. The goal is to detect: (i) the number of clusters; (ii) which videos belong to each cluster (i.e., ND videos). For both tasks, we tested two different clustering approaches on X: (i) DFS(X, Γ) denotes the use of Depth First Search algorithm on matrix X, binarized according to a threshold $\Gamma = 0.06$ (learned on a small training set); (ii) Hier(X, Γ) denotes hierarchical clustering on X using correlation distance and euclidean linkage, where $\Gamma = 1$ represents the cutoff threshold (learned on a small training set). The first approach is based on the knowledge that X is composed by NCC values of fingerprints. Therefore it links in the same cluster videos whose NCC is grater than Γ. The second approach makes use

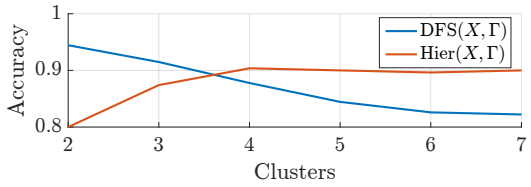
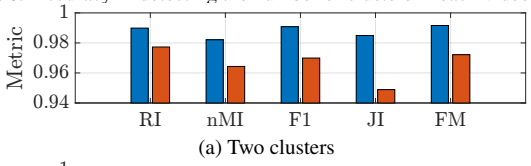
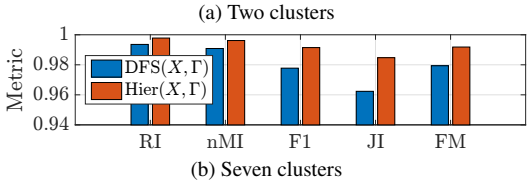


Fig. 6: Accuracy in detecting the number of clusters in each video set.



(a) Two clusters



(b) Seven clusters

Fig. 7: Clustering metrics obtained with the proposed techniques on datasets that contain two (a) and seven (b) ND clusters.

of rows of X as generic features, applying the hard Γ thresholding only at the very end.

Fig. 6 shows the accuracy in correctly detecting the number of ND clusters within each video set. Accuracy is always greater than 80%. If the number of clusters to be detected is small (i.e., 2) the DFS(X, Γ) strategy tends to better results. Conversely, the Hier(X, Γ) approach has a better accuracy (90%) when the number of ND clusters is greater than 4. DFS applied on M [10] aggregated more ND clusters together, thus it was never able to correctly detect the number of clusters.

Evaluation of clustering approaches is carried out by measuring how well pairs of ND are assigned to the same cluster. Specifically, we used the *Rand index* (RI) [20] (the percentage of data pairs that are correctly clustered), the *Jaccard index* [21] (JI) (similar to RI, not considering pairs of elements that are in different clusters), the *F-measure* (F1) [22] (the harmonic mean of precision and recall), the *Fowlkes-Mallows index* (FM) [23] (the geometric mean of precision and recall), and the *normalized mutual information* measure (nMI) [24] (residual entropy within clusters). All clustering measures yields values between 0 (worst result) and 1 (best result).

Fig. 7 shows the aforementioned metrics on $\mathcal{V}^2_{\text{ND}}$ (a) and $\mathcal{V}^7_{\text{ND}}$ (b). In the first case, only two ND clusters are present, and DFS(X, Γ) shows more promising results. Conversely, the second scenario validates the use of Hier(X, Γ) when more clusters of NDs are present within the analysis pool.

5. CONCLUSIONS

We proposed a pipeline based on hash and camera fingerprint tailored to separate SSI videos while clustering together ND ones. We validated the proposed algorithm on a set of more than 12 000 video sequences specifically designed in a challenging scenario, achieving promising results. From the computational point ov view, it is important to notice that a lot of data can be precomputed (e.g., hashes for rough detection, fingerprints for refinement, etc.) and only videos that pass the rough detection step are further processed, thus decreasing the computational burden. Future work will be devoted to enrich the pipeline exploiting additional information coming from this multi-view scenario, as done in [25] for the case of still images.

6. REFERENCES

[1] S. Milani, M. Fontani, P. Bestagini, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro, "An overview on video forensics," *APSIPA Transactions on Signal and Information Processing*, vol. 1, pp. e2, 2012.

[2] M. Stamm, W. Lin, and K. Liu, "Temporal forensics and anti-forensics for motion compensated video," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 7, pp. 1315 –1329, 2012.

[3] L. D'Amiano, D. Cozzolino, G. Poggi, and L. Verdoliva, "Video forgery detection and localization based on 3D patchmatch," in *IEEE International Conference on Multimedia Expo Workshops*, 2015.

[4] P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro, "Codec and GOP identification in double compressed videos," *IEEE Transactions on Image Processing (TIP)*, vol. 25, pp. 2298–2310, 2016.

[5] L. Kennedy and S.-F. Chang, "Internet image archaeology: automatically tracing the manipulation history of photographs on the web," in *ACM International Conference on Multimedia (ACM-MM)*, 2008.

[6] J. R. Kender, M. L. Hill, A. P. Natsev, J. R. Smith, and L. Xie, "Video genetics: A case study from youtube," in *ACM International Conference on Multimedia (ACM-MM)*, 2010.

[7] Z. Dias, A. Rocha, and S. Goldenstein, "Video phylogeny: Recovering near-duplicate video relationships," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2011.

[8] F. O. Costa, S. Lameri, P. Bestagini, Z. Dias, A. Rocha, M. Tagliasacchi, and S. Tubaro, "Phylogeny reconstruction for misaligned and compressed video sequences," in *IEEE International Conference on Image Processing (ICIP)*, 2015.

[9] F. O. Costa, S. Lameri, P. Bestagini, Z. Dias, S. Tubaro, and A. Rocha, "Hash-based frame selection for video phylogeny," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2016.

[10] S. Lameri, P. Bestagini, A. Melloni, S. Milani, A. Rocha, M. Tagliasacchi, and S. Tubaro, "Who is my parent? reconstructing video sequences from partially matching shots," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014.

[11] A. Jaimes, S.-F. Chang, and A. C. Loui, "Duplicate detection in consumer photography and news video," in *ACM Workshop on Multimedia and Security*, 2002.

[12] B. Coskun, B. Sankur, and N. Memon, "Spatio-temporal transform based video hashing," *IEEE Transactions on Multimedia (TMM)*, vol. 8, pp. 1190–1208, 2006.

[13] A. Melloni, S. Lameri, P. Bestagini, M. Tagliasacchi, and S. Tubaro, "Near-duplicate detection and alignment for multi-view videos," in *IEEE International Conference on Image Processing (ICIP)*, 2015.

[14] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Source digital camcorder identification using sensor photo-response nonuniformity," in *SPIE Electronic Imaging (EI)*, 2007.

[15] S. Bayram, H. T. Sencar, and N. Memon, "Video copy detection based on source device characteristics: A complementary approach to content-based methods," in *ACM International Conference on Multimedia Information Retrieval*, 2008.

[16] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 1, pp. 205–214, 2006.

[17] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding (CVIU)*, vol. 110, pp. 346–359, 2008.

[18] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commununications of the ACM*, vol. 24, pp. 381–395, 1981.

[19] M. Goljan and J. Fridrich, "Camera identification from cropped and scaled images," in *SPIE Electronic Imaging (EI)*, 2008.

[20] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, pp. 846–850, 1971.

[21] P. Jaccard, "The distribution of the flora in the alpine zone.," *New phytologist*, vol. 11, no. 2, pp. 37–50, 1912.

[22] C. Van Rijsbergen, *Information Retrieval*, Butterworth-Heinemann, 1979.

[23] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American Statistical Association*, vol. 78, pp. 553–569, 1983.

[24] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of machine learning research*, vol. 3, pp. 583–617, 2002.

[25] S. Milani, P. Bestagini, and S. Tubaro, "Phylogenetic analysis of near-duplicate and semantically-similar images using viewpoint localization," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2016.