Forensic Science International 231 (2013) 178–189

Contents lists available at SciVerse ScienceDirect

Forensic Science International

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

Toward image phylogeny forests: Automatically recovering semantically similar image relationships

Zanoni Dias, Siome Goldenstein, Anderson Rocha *

Institute of Computing, University of Campinas, Campinas, SP 13083-852, Brazil

ARTICLE INFO

Article history:
Received 17 October 2012
Received in revised form 2 May 2013
Accepted 7 May 2013
Available online

Keywords:
Image phylogeny
Phylogeny trees
Kinship analysis
Digital forensics

ABSTRACT

In the past few years, several near-duplicate detection methods appeared in the literature to identify the cohabiting versions of a given document online. Following this trend, there are some initial attempts to go beyond the detection task, and look into the structure of evolution within a set of related images overtime. In this paper, we aim at automatically identify the structure of relationships underlying the images, correctly reconstruct their past history and ancestry information, and group them in distinct trees of processing history. We introduce a new algorithm that automatically handles sets of images comprising different related images, and outputs the phylogeny trees (also known as a *forest*) associated with them. Image phylogeny algorithms have many applications such as finding the first image within a set posted online (useful for tracking copyright infringement perpetrators), hint at child pornography content creators, and narrowing down a list of suspects for online harassment using photographs.

1. Introduction

Most of us know the power of social media and its importance in making people more connected. Indeed, the new century's first decade has seen a vast rise of modern social media. What was formerly restricted to college campuses now easily achieve over millions of people with astonishing media uploading and sharing rates. For instance, Youtube claims that one hour of video is uploaded to their computers per second or more than 86 thousand hours of video per day (c.f., http://www.youtube.com/t/press_statistics). More importantly, Youtube has more than four billion video views per day which shows that content is not only being uploaded but also consumed.

Such shear amount of data has brought to us challenges never imagined before. For example, within such massive amount of data it is common that several documents are duplicates or near-duplicates of one another. While it is straightforward to find exact duplicates among available media, that does not hold true when media objects undergo small modifications. It is fairly common small changes to occur during the redistribution, usually without interfering on their semantic meaning. This is what we call *near-duplicate* media objects. These modifications can include, among others, A/D or D/A conversions, (de)-coding, transmission noise, and small editing/corrections such as brightness adjustments, or cropping.

The identification of near-duplicates has received particular attention during the past few years and [1–3] are just some examples in this area. However, a challenging task which has been vastly overlooked until recently, arises when we need to identify which document is the original within a set of digital related objects, and the structure of generation of each of them. In this case, we need to go beyond the identification of near-duplicate documents. We have to consider a population of multimedia objects as a whole to study the relationships among objects and their past history, as a result from the way they have been generated and manipulated overtime.

Although most of the changes related to near-duplicate multimedia objects are natural and not necessarily harmful, sometimes the distribution itself might cause copyright infringement or even represent a criminal action [4,5]. In some situations, the spreading pattern of an image or video can help companies to understand demographics and effectiveness of an ad campaign or a product. The identification of the original image posted online can help the analysis of a copyright infringement complaint. The original image is also the best candidate for a forensic authenticity analysis [6].

These scenarios motivated the dawning of a new research subfield called *Multimedia Phylogeny* [4], to investigate the history and evolutionary process of digital objects. In other words, we are looking for the structure of modifications of multimedia objects.

Solutions to problems in Multimedia Phylogeny have many applications:

* Corresponding author. Tel.: +55 19 3521 5854; fax: +55 19 3521 5838.
E-mail addresses: zanoni@ic.unicamp.br (Z. Dias), siome@ic.unicamp.br (S. Goldenstein), anderson@ic.unicamp.br, anderson.rocha@gmail.com (A. Rocha).

0379-0738/$ – see front matter © 2013 Elsevier Ireland Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.forsciint.2013.05.002

(a) security and law-enforcement (e.g., by narrowing down a list of suspects for online harassment);
(b) forensics (e.g., finding original documents within a set of related ones and allowing for more advanced document forensic analyses);
(c) copyright enforcement (e.g., traitor tracing without the requirement of active source control solutions such as watermarking or fingerprinting);
(d) news tracking services (e.g., document relationships can feed news tracking services with key elements for determining the opinion forming process across time and space [7,8]);
(e) content-based retrieval systems (e.g., showing similar photographs but of different photographers to a user without any metadata analysis).

Only recently there have been the first attempts to go beyond the near-duplicate identification problem to pinpoint the structure of relationships within a set of objects [9,4,10,8,7]. However, these early investigations are constrained to the case of image and video near-duplicates, which are related by a set of possible transformations – e.g., cropping, affine warping (considering as special cases resampling, rotation and translation), brightness/contrast adjustment and lossy compression. The main objective of such prior work was to identify the phylogeny tree associated with a set of near-duplicate images or videos.

In this paper, we go beyond prior work on Near Duplicate Images (NDI) and aim at finding the phylogeny trees within a set of Semantically Similar Images (SSI). Prior work in the literature have assumed the existence of relationships when analyzing a set of images. In this paper, without assuming the existence of relationships, we automatically find when images share a chain of processing history. For a better understanding, we formally define NDI and SSI documents in Section 2.

We expand upon state-of-the-art solutions [4,10,9] and present a new algorithm that automatically deals with sets of images from different sources, finding the different phylogeny trees.

We faced this problem for the first time while performing a real forensic analysis. On April 5th, 2009 [11], the Brazilian newspaper Folha de São Paulo, a major news player in Brazil, published an article about President Dilma Rousseff, back then the Brazilian Chief of Staff and a potential candidate for the 2010s presidential election (currently the president of that country). This article claimed that during her participation in the resistance to the Brazilian dictatorship, in the 60s, she engaged in violent or terrorist activities, such as armed robberies and kidnappings. To support this claim, the newspaper printed an alleged image of Secretary Rousseff's dossier from the internal files of the Repression Police (see Fig. 1), arguing it was obtained from the Public Archive of São Paulo, responsible for housing this collection of documents from that period of time.

Ms. Rousseff, who always declared herself as participant in a non-violent resistance movement, denied the allegations in the article and hired us to perform a forensic analysis of the image's authenticity. The newspaper never provided us the original printed image. Additionally, the image was virally widespread over the internet even before the newspaper chose to publish it – there were hundreds of copies in many different websites and blogs. Most of the copies were not exact but each one could have undergone additional image processing operations such as rescaling, cropping, and color adjustments.

That was the point where we identified that the literature lacked a robust approach for associating related images overtime, and the turning point for creating the multimedia phylogeny area. We needed a technique to answer questions

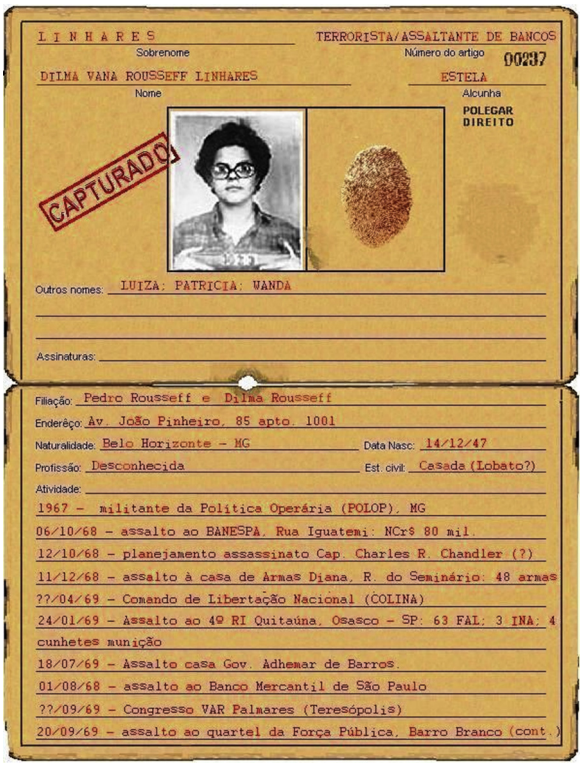


Fig. 1. The questioned object: the alleged image of Repression's Secret Police files on the Secretary of State Dilma Rousseff in 2009 as published by the Brazilian newspaper Folha de São Paulo.

such as: what image was the least modified one and, probably, the original released online (root of the tree)? What was the evolutionary associated tree? Could the tree's root (possibly the original image) be associated with auxiliary information of logs and website collected data to point out the perpetrator? In Section 6.4, we show the phylogeny tree associated with a few collected images related to this case and how we could put more effort analyzing the images on top of the tree instead of the leaves (which represent the least important modified versions for forensic purposes).

There are many forensic applications to image phylogeny solutions. Consider postings on the internet of private and/or abusive photographs of a regular person, such as in a bulling situation. Equally disturbing are online postings with fake and defamatory image content of celebrities or politicians (such as the case we discussed earlier). Similarly, our algorithm could help the fight against online child pornography (CP). Criminals use the internet to sell and disseminate CP images, and once these images go online, people usually redistribute them quickly using all sorts of hiding and changing methods.

Phylogeny algorithms can help us understand the evolutionary process among the set of replicated images. With the use of other metadata, and additional investigative work, it may be possible to track down the actual individuals who initially published the content online. Tools such as Microsoft's PhotoDNA [12,13] characterizes images using unique signatures looking for modified versions in an attempt to help law-enforcement to chase child pornographers. However, PhotoDNA normally looks only for exact or very similar copies, and it is not concerned about the evolutionary process among the images. Our solutions in image phylogeny look into the images' ancestry relationships, and is complementary to PhotoDNA.

This paper has seven sections. Section 2 formally defines what are near duplicate documents (NDI) and semantically similar documents (SSI). Section 3 briefly describes the related work in the literature while Section 4 presents the necessary background for the understanding of this paper. Section 5 introduces our novel image phylogeny forests algorithm, and Section 6 shows the experiments we perform to validate the proposed method. Finally, Section 7 concludes the paper and hints at possible future work.

2. Definitions

Two key definitions used in this paper are the concept of Near Duplicate Images (NDI) and Semantically Similar Images (SSI).

Definition 1. A set of Near Duplicate Images (NDI) refers to a set of images whose content is similar and that can all be obtained by a chain of transformations from a original source image. These transformations can be very complex, and include cropping, rotation, translation, scaling, brightness/contrast adjustments, gamma correction, lossy compression.

Definition 2. The concept of Semantically Similar Images (SSI) is a generalization of the NDI – each image in the set can be obtained by a chain of transformations from a group of original source images. For instance, we might have a set of SSI comprising two sets of near duplicates, each set rooted at a source that comes from a different camera, or that comes from the same camera but taken from a different point in space and time.

3. Related work

In the past decade, we have seen increasing progress on the development of efficient and effective systems to identify the cohabiting versions of a given document in the wild [1–3]. However, only recently there were the first attempts to go beyond the detection of near duplicates, with attempts to identify the structure of relationships within a set of near-duplicates.

During its lifetime, a multimedia object might undergo different processing stages whereby each processing operator might alter the underlying features of the object's content in a characteristic and detectable manner. Establishing relationships between pairs of digital objects through the analysis of their content is challenging due to the diversity of processing operators.

Kennedy and Chang [7] first addressed the problem of parent–child relationships between pairs of images. Their work proposed the detection of plausible parent–child relationships within a set of images using a visual migration map (VMM), representing an approximation of the history of the images. However, the authors did not discuss how to find possible parameters for the family of transformations that lead a parent image to its resulting offspring.

Different from the VMM approach proposed in [7], Rosa et al. [8] proposed to detect the image dependencies within a set of images by considering that the images' mutual information can be expressed as the sum of the mutual information between the content-based components and the content-independent components between them.

Dias et al. [9] introduced and defined the problem of Image Phylogeny Tree reconstruction finding the structure of transformations, and their parameters, that generated a given set of near-duplicate images. The authors presented an initial solution to the problem constrained to a tree, i.e., all images have a single source. This work has been expanded upon in [4] in which the authors presented an initial solution for dealing with more than one tree but requiring input from the user on how many trees to seek for. In [10], the authors showed the applicability of such phylogeny tree algorithms to the context of videos and introduced a first solution to reconstruct the tree of evolution of a set of near-duplicate videos.

Kender et al. [14] studied content-based relationships among video clips downloaded from YouTube and related to the same event. They illustrated the construction of a content-dependency graph, whose structure is justified from a "genetic" standpoint by representing how videos evolve in terms of mutations, crossover and other operators.

Although such approaches represent an important step toward the solution of multimedia phylogeny problems, they were constrained to the case of finding the tree of evolution of image/video near-duplicates. In addition, some of them required input from the user regarding the number of trees to seek for in the forest. In this paper, our objective is to go a step beyond and automatically determine the forest (set of trees) within a given collection of images.

According to [15], "Digital forensics can be defined as the collection of scientific techniques for the preservation, collection, validation, identification, analysis, interpretation, documentation, and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events, usually of a criminal nature." Since multimedia phylogeny algorithms help us to reconstruct events associated with a set of documents (e.g., images), it can be seen as a subfield of digital forensics.

Multimedia Phylogeny algorithms need to take advantage of important findings and methods developed on the last two decade in digital forensics related areas. To determine how two images relate to each other, we need to investigate several additional footprints associated with them.

Device identification is one of the most studied problems – when we want to identify if two images come from the same acquisition device. In this sense, multimedia phylogeny could benefit from techniques that rapidly decorrelate two near-duplicates as coming from different acquisition sensors. Some important works in acquisition footprint might be found in [16,5]. Digital lens reflex footprints have been explored in [17]. Democaising artifacts were explored in [18] while [19] explored the interpolation telltales for forensics purposes.

Image lossy compression, such as the one found in the JPEG image format, also leaves significant footprints. The image compression history associated with an image has been investigated by Mao et al. [20] and by Fan and Queiroz [21] whereby they investigated the temporal signatures associated with images. The first discussed an information theoretical approach for device temporal forensics while the latter discussed the identification compression history associated with a bitmap image. It is common to find images, originally stored in JPEG format, further edited and re-saved as new JPEGs. Double JPEG compression leaves characteristic telltales, which were vastly explored in the forensics literature [22]. Multiple JPEG compression has been analyzed in [23]. In the case of video coding, footprints might be related to the coding parameters (e.g., GOP structure) [24].

Editing-based footprints can also be helpful. Image editing operations can leave statistically identifiable artifacts that can be explored for further correlating the temporal evolution of a pair of images. Numerous works have investigated such footprints such as cloning detection [25,26], resampling [27], local tampering [28], chromatic aberration [29], camera response functions signatures [30]. Also, several researchers have explored illumination telltales for identifying traces of tampering [31,32]. The study of video editing footprints, as a parallel, is still limited. An example of such study is [24].

Information related to acquisition, coding-based, and editing-based footprints can be used together to devise a strong measure of similarity between a pair of images. Here, we use only simple geometry, pixel-based illumination adjustments (e.g., brightness, contrast, gamma correction), and compression. More information about image-related footprints can be found in [33,5,34,35] with discussions of the state of the art as well as connections among the currently available tools and solutions.

4. Concepts on multimedia phylogeny

According to [4], an *Image Phylogeny Tree* represents the structure of transformations and the evolution of a set of near-duplicate images. In general, we have a tree reconstruction algorithm that builds upon an initial set of near-duplicates and a dissimilarity function d. This function yields small values for ordered pairs that are likely father and son on the tree, and large values for ordered pairs that are unlikely so. As most of the concepts discussed in this section come from [4], the contents here might be similar.

Let $T_{\vec{\beta}}$ be an image transformation from a family $\mathcal{T}$. We can define a dissimilarity function between two images $\mathcal{I}_A$ and $\mathcal{I}_B$ as the minimum

$$d_{\mathcal{I}_A,\mathcal{I}_B} = \min_{\vec{\beta}} \left| \mathcal{I}_B - T_{\vec{\beta}}(\mathcal{I}_A) \right|_{\text{point-wisecomparison } \mathcal{L}}, \quad (1)$$

for all possible values of $\vec{\beta}$ that parameterizes $\mathcal{T}$. Eq. (1) measures the amount of residual between the best transformation of $\mathcal{I}_A$ to $\mathcal{I}_B$, according to the family of operations $\mathcal{T}$. We can use any point-wise comparison method $\mathcal{L}$ for the final residual analysis. Here, to calculate the dissimilarity between two images, we uncompress both of them, estimate the possible transformations to which they were subject with respect to each other and calculate their point-wise dissimilarity using the standard *Minimum Squared Error* (MSE) as $\mathcal{L}$. Different dissimilarity functions could be used here, Rosa et al. [8] have suggested the use of noise-related signatures for comparing images.

With a set of n near-duplicate images, the first task for creating an image phylogeny tree is to calculate the dissimilarity between every pair of such images. In this case, we need to consider a set of possible image transformations, $\mathcal{T}$, from which one image can generate an offspring [8,4].

An *Image Phylogeny Forest* represents the structure of transformations and the evolution of a set of semantically similar images. Each tree of the forest represents the structure of transformations and the evolution of a set of near-duplicate images. A forest comprises distinct sets of near-duplicate images but all images are semantically similar.

5. Proposed method

As discussed in [4], there are two steps in the process of reconstructing an image phylogeny tree from a set of near-duplicate images: the dissimilarity function and tree-building algorithm. The same applies for reconstructing an image phylogeny forest with the additional challenge that we need to automatically discover the number of trees in the forest. This paper's contribution is a new algorithm for finding the set of trees related to a set of n semantically similar images such that each tree represents the relationships between image near-duplicates.

In a general setup, we can determine the forest associated with a set of documents either by performing operations directly on the dissimilarity matrix M or by changing the tree building algorithm. In the first case, we could devise clustering algorithms to find the number of trees in the forest automatically from the dissimilarity matrix. In the second case, we could design a forest building algorithm directly over a dissimilarity matrix M. In this paper, we adopt the latter alternative and use the same setup for the dissimilarity matrix M as in [4].

5.1. Creating the dissimilarity matrix

Following [4], for each possible pair of images $\mathcal{I}_A$ and $\mathcal{I}_B$, we estimate $\vec{\beta}$ that minimizes the dissimilarity function of Eq. (1). Then, we follow the steps below:

1. calculate the corresponding points between images $\mathcal{I}_A$ and $\mathcal{I}_B$ using the Speeded-Up Robust Features (SURF) algorithm [36];
2. robustly estimate the affine warping transformation parameters $\vec{\beta}$ for image $\mathcal{I}_A$ with respect to $\mathcal{I}_B$ taking the corresponding points into consideration and using RANSAC algorithm [37], finding
   $\mathcal{I}'_A = T_{\vec{\beta}}(\mathcal{I}_A)$;
3. calculate the mean and variance of each $\mathcal{I}_B$'s color channel and normalize image $\mathcal{I}_A$'s color channels using such measures. For each color channel c,
   $$\mathcal{I}''_{A_c} = (T_{\vec{\beta}}(\mathcal{I}_{A_c}) - \mu_{A_c}) \frac{\sigma_{B_c}}{\sigma_{A_c}} + \mu_{B_c},$$
   where μ_{A_c} and μ_{B_c} are the mean value of the color channel c of $T_{\vec{\beta}}(\mathcal{I}_A)$ and $\mathcal{I}_B$ respectively, while $\mathcal{I}_{A_c}$ and $\mathcal{I}_{B_c}$ are the standard deviations;
4. compress the result of Steps 2 and 3 according to $\mathcal{I}_B$'s quantization table (QT),
   $\mathcal{I}'''_A = \text{compress}\mathcal{I}''_A$ with the QT from $\mathcal{I}_B$.

Steps 1 and 2 find stable interest points in both images, and then calculate the geometric distortions between each pair of images robustly, using RANSAC [37]. With three points in each image we can use triangulations and estimate such geometric transformations. In Step 3, we perform pixel intensity normalization of image $\mathcal{I}_A$ according to the $\mathcal{I}_B$ color channels' mean and variance. Step 3 analyses the color differences of a pair of images and try to quantify them. The idea is to estimate how much color transformation an image undergoes to generate an offspring. Step 4 compresses image $\mathcal{I}_A$ according to $\mathcal{I}_B$'s quantization table. Finally, we uncompress both of them and calculate their point-wise dissimilarity on the domain of the target image. We actually consider $\mathcal{I}_A$'s and $\mathcal{I}_B$'s quantization tables. If we are estimating the cost for compressing $\mathcal{I}_A$ on the domain of $\mathcal{I}_B$, we use $\mathcal{I}_B$'s quantization table. If it is the other way around, we use $\mathcal{I}_A$'s quantization table. The rationale with this step is that if an image $\mathcal{I}_A$ generates an offspring $\mathcal{I}_B$, them if we want to check if $\mathcal{I}_B$ is a descendant of $\mathcal{I}_A$, we need to recompress $\mathcal{I}_B$ using $\mathcal{I}_A$'s quantization table since that is the quantization table originally present.

Fig. 2 depicts the process of mapping one image to another image's domain.

5.2. Automatic reconstruction of image phylogeny forests

The algorithm used to reconstruct an image phylogeny forest is as important as the dissimilarity matrix related to the n semantically similar images. In Section 5.1, we explained how to build a dissimilarity matrix from n semantically similar images

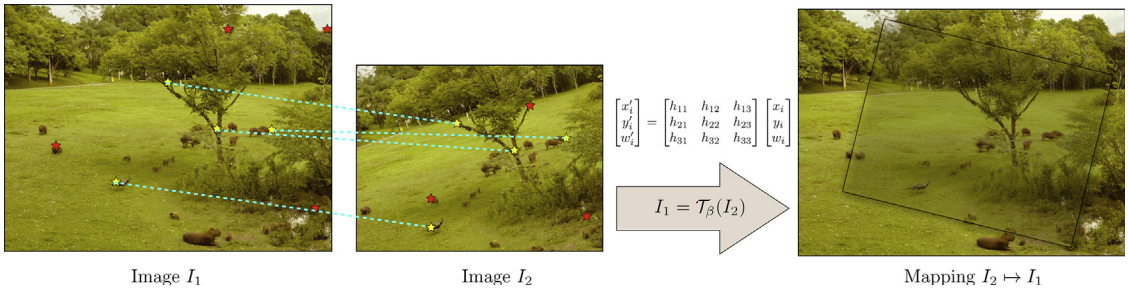


Fig. 2. To calculate image dissimilarities between a pair of images I_1 and I_2, we find robust points of interest in both images and for those which are good matches (yellow stars) we calculate an homography matrix representing the necessary parameters to transform one image to another's domain. Once we perform the mapping, we can compare both images pixel wise within the region of interest they overlap. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

taking into consideration a family of transformations. In this section, we introduce an algorithm for the construction of *Image Phylogeny Forests* based on a modified version of the algorithm proposed by [4]. Their work aimed at finding the phylogeny tree for a set of near-duplicate images, but requiring an input from the user for task involving more than one tree. On the other hand, our extension, named *Automatic Oriented Kruskal*, can determine the number of trees in a forest and reconstruct such trees without any user intervention.

Given a dissimilarity matrix M built upon a set of n semantically similar images, our algorithm first considers that each images in the collection is the root of a tree. Then, the algorithm starts processing good edges (low weight) to connect trees. While the Oriented Kruskal algorithm processes $n - l$ edges continuously (l is a limit asked to the user), our algorithm keeps track of the variance of processed edges and only adds a new one to the forest if the weight of such an edge is lower than k times the standard deviation of the edges processed up to that point. The parameter k is calculated a priori based on the arc-weight distribution of some example trees. In this paper, we set up $k = 2$; see Section 5.4 for the proper justification. The algorithm stops when either it discovers that the new candidate edge to consider is much higher than the ones it already processed and accepted or when it already processed $n - 1$ edges.

Algorithm 1 presents the operations step-by-step. Lines 1–3 of Algorithm 1 initialize the `tree` vector with n initial trees, each one containing a vertex representing an image. Each position `trees[ i]` denotes the parent of a node with `id=i`. At the end, the variable `trees` contains the tree(s) representations. For instance, `trees = [ 1, 1, 2, 4]` represents a forest with two trees, one with three nodes and one with a single node. Vertex 1 is the root of the first tree and also the parent of vertex 2, which in turn, is the parent of vertex 3. Finally, Vertex 4 is the root of the second tree.

Lines 4–5 initialize the auxiliary variable n_{edges} to count the number of accepted edges, and x_1 and x_2 are used to iteratively calculate the standard deviation of accepted edges. Line 6 sorts the arc-weights in the dissimilarity matrix M. The **for** loop in Lines 7–24 examine matrix positions in order of dissimilarity, from lowest to highest. The **for** loop checks, for each position (i, j), if the endpoints i and j do not belong to the same tree and if j is the root of a tree. Next, we need to further investigate if the edge $(j \rightarrow i)$ weight is greater than k times the known standard deviation of previously added edges plus the latest accepted edge. If so, the algorithm stops and returns the forest it calculated. Otherwise, the algorithm includes the new edge to the forest and updates the number of accepted edges, standard deviation of edges in the forest and so on. In the end, the variable *trees* represents all the trees in the forest.

Algorithm 1. Automatic Oriented Kruskal (AOK).

Input: number of semantically similar images n, $n \times n$ dissimilarity matrix M and number of standard deviations used as limit k, default is $k = 2$.

1: **for** $i \in [1..n]$ **do** ▷ Initialization
2: $trees[i] \leftarrow i$
3: **end for**
4: $n_{edges} \leftarrow 0$ ▷ Number of processed edges in the forest
5: $x_1 \leftarrow x_2 \leftarrow 0$ ▷ Variables for dynamically calculating the SD
6: $sorted \leftarrow$ **Sort** positions (i, j) of M into nondecreasing order
7: **for** each position $(i, j) \in sorted$ **do**
8: **if** (`Root`$(i) = i$ and `Root`$(j) \neq i$) **then** ▷ Is the new edge valid?
9: **if** $(n_{edges} > 1)$ **then**
10: $sd \leftarrow \sqrt{\frac{x_1 - \left(\frac{x_2^2}{n_{edges}}\right)}{n_{edges} - 1}}$
11: **if** $(M[i, j] - last > k * sd)$ **then**
12: **return** *trees*
13: **end if**
14: **end if**
15: $n_{edges} \leftarrow n_{edges} + 1$ ▷ Updates the auxiliary variables
16: $last \leftarrow M[i, j]$ ▷ Last processed edge
17: $x_1 \leftarrow x_1 + M[i, j]^2$
18: $x_2 \leftarrow x_2 + M[i, j]$
19: $trees[i] \leftarrow j$ ▷ Adds new edge to the forest
20: **if** $(n_{edges} = n - 1)$ **then** ▷ Algorithm is complete
21: **return** *trees* ▷ Returning the final forest
22: **end if**
23: **end if**
24: **end for**

The algorithm's running time depends on how we implement the `Root` function. If we use a *disjoint-set-forest* with the *union-by-rank* and *path-compression heuristics*, we can implement such a function very efficiently [38] in line with the running time of the algorithm proposed in [4]. The final complexity of the algorithm is $O(n^2 \log n)$ where n is the number of semantically similar images in the collection.

5.3. Simulation of the algorithm for one forest

Fig. 3 depicts the execution of the proposed algorithm for a toy example with $n = 10$ semantically similar images. The algorithm initially receives a dissimilarity matrix M that contains the dissimilarities between each pair of images.

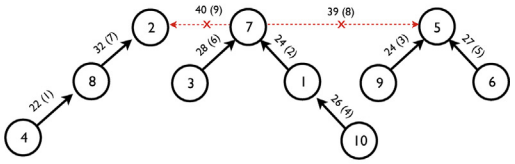
Dissimilarity Matrix

M	1	2	3	4	5	6	7	8	9	10
1	–	52	29	57	43	55	24	52	49	26
2	50	–	45	35	60	68	40	42	63	53
3	42	48	–	56	42	54	28	52	50	29
4	56	23	55	–	65	67	50	22	66	58
5	45	61	43	67	–	29	42	63	43	52
6	54	69	65	70	27	–	48	68	45	65
7	25	50	30	52	39	50	–	50	48	48
8	51	32	49	30	61	67	49	–	62	55
9	52	65	55	69	24	30	49	64	–	61
10	25	54	26	59	50	61	30	56	60	–

Step-by-Step Simulation

Step	Limit	X	Y	D(X,Y)	SD
1	∞	4	8	22	∞
2	∞	4	2	23	∞
3	∞	1	7	24	1.41
4	26.83	9	5	24	1.15
5	26.31	7	1	25	1.15
6	26.31	10	1	26	1.63
7	29.27	1	10	26	1.63
8	29.27	10	3	26	1.63
9	29.27	6	5	27	1.95
10	30.90	3	7	28	2.23
11	32.46	1	3	29	2.23
12	32.46	3	10	29	2.23
13	32.46	5	6	29	2.23
14	32.46	7	3	30	2.23
15	32.46	8	4	30	2.23
16	32.46	9	6	30	2.23
17	32.46	10	7	30	2.23
18	32.46	8	2	32	3.29
19	38.58	2	4	35	3.29
20	38.58	7	5	39	3.29
21	38.58	2	7	40	3.29

Forest [7, 2, 7, 8, 5, 5, 7, 2, 5, 1]



$$Limit = 2\sigma + D'(X, Y)$$

Fig. 3. Step-by-step simulation of the Automatic Oriented Kruskal (AOK) algorithm to construct an Image Phylogeny Forest with three trees and 10 images from a 10 × 10-Dissimilarity Matrix. $D(X, Y)$ denotes the last accepted edge weight. For instance, in Step 6, the last accepted edge has weight 24, therefore the limit is $Limit = 2 \times 1.15 + 24 \cong 26.31$. (For interpretation of the references to color in the text, the reader is referred to the web version of the article.)

The algorithm starts with $n = 10$ trees in the forest and it starts processing edges that are candidates to include in the forest, connecting some of its trees into bigger trees. First, the algorithm processes edge $(4 \rightarrow 8)$ which has the lowest entry in the dissimilarity matrix M and then connects nodes 4 and 8 in the forest. Next, the algorithm test the edge $(4 \rightarrow 2)$ but it is discarded since 4 is not the root of a tree. The algorithm tests the next eligible edge $(1 \rightarrow 7)$ and selects it. At this point, the forest has two selected edges and it is possible to calculate the current standard deviation of such selected edges which is $\sigma \cong 1.41$. Therefore, the dynamic limit for not accepting new edges is $Limit = 2\sigma + D(1, 7) = 2 \times 1.41 + 24 \cong 26.83$. Recall that $D(1, 7)$ refers to the dissimilarity between images 1 and 7.

Next, the algorithm selects the edge $(9 \rightarrow 5)$ since it connects two different trees, does not create a loop and its weight is smaller than the 26.83 and updates σ to $\cong 1.15$. The algorithm proceeds by checking each edge in order until the 20th iteration when it evaluates the edge $(7 \rightarrow 5)$. This edge passes the first two tests (it connects two different trees and does not create a loop). However, its value is above the allowed limit calculated so far for all the previously selected edges in the forest. This edge represents a dissimilarity of 39 but the current limit for entering in the forest is 38.58. This edge is then discarded and the algorithm stops, returning the forest depicted on the far right of Fig. 3.

If we use the algorithm proposed in [4], it will accept two more edges $(7 \rightarrow 5)$ and $(7 \rightarrow 2)$ ending up with a single tree with 10 nodes (dotted red lines in the figure on the right). Our algorithm automatically finds the correct number of trees in the forest and stops when it finds that the edge $(7 \rightarrow 5)$ should not be selected for the forest (iteration 20).

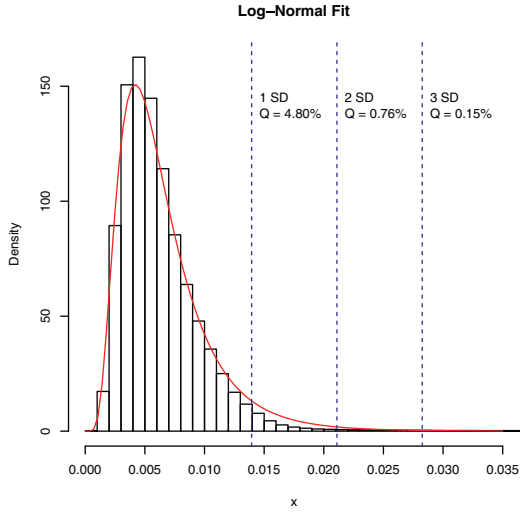


Fig. 4. Log-Normal fitting for valid trees (1 . . . 5) and forests and possible threshold candidates for finding the right size of forest. In this case, we are considering all edge weights of all trees for single and multiple cameras as data from a single distribution.

5.4. Choosing the right distribution cutoff

As discussed above, our algorithm relies on the choice of a good threshold point that will select only edges that belong to valid trees. If this threshold is too small, the algorithm will reject valid edges and we will end up with a forest larger than the correct value. If the threshold is too high, the algorithm will incorrectly accept edges, and find a smaller number of trees in the forest than the right solution.

To come up with a reasonable threshold, we have studied the behavior of the dissimilarity values of valid trees and forests. As we can observe in Fig. 4, the dissimilarities of the edges of the real trees are reasonably described by a Log-Normal distribution

$$p(x)_{\mu,\sigma} = \frac{1}{x\sqrt{2\pi\sigma^2}} e^{-(\ln x - \mu)^2 / 2\sigma^2} \quad (2)$$

with $\mu = -5.178$ and $\sigma = 0.544$. In the figure, we show the behavior of three possible thresholds, $\mu + \sigma$, $\mu + 2\sigma$, and $\mu + 3\sigma$ which, according to the estimated distribution, would each reject 4.8%, 0.76%, and 0.15% of the correct edges.

Given that we want a threshold as tight as possible to avoid incorrect tree merging, we select $\mu + 2\sigma$ as the threshold. A Kolmogorov–Smirnov test for such Log-Normal shows a p-value of 0.0255 (confidence of 97.5%) which demonstrates it is a reasonable choice for the problem we need to solve.

Fig. 5, depicts the Log-Normal fitting for single (OC) and multiple (MC) cameras considering forests of different sizes (1 . . . 5 trees). Note that the Log-Normal distribution reasonably describes the data regardless the number of trees in the forest and the type of image capture (single/multiple cameras).

A branch of research we are further investigating is why this choice holds and also its theoretical implications for the multimedia phylogeny problem as a whole.

6. Experiments and validation

We follow the methodology introduced by Dias et al. [4] for the validation of our new algorithm, and look at four different quantitative metrics (*Root Evaluation*, *Edges Evaluation*, *Leaves Evaluation*, and *Ancestry Evaluation*) to evaluate a reconstructed forest in scenarios where we have Ground Truth. All the metrics are adapted to forests and calculated according to the following

$$M(\mathrm{IPF}_1, \mathrm{IPF}_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}, \quad (3)$$

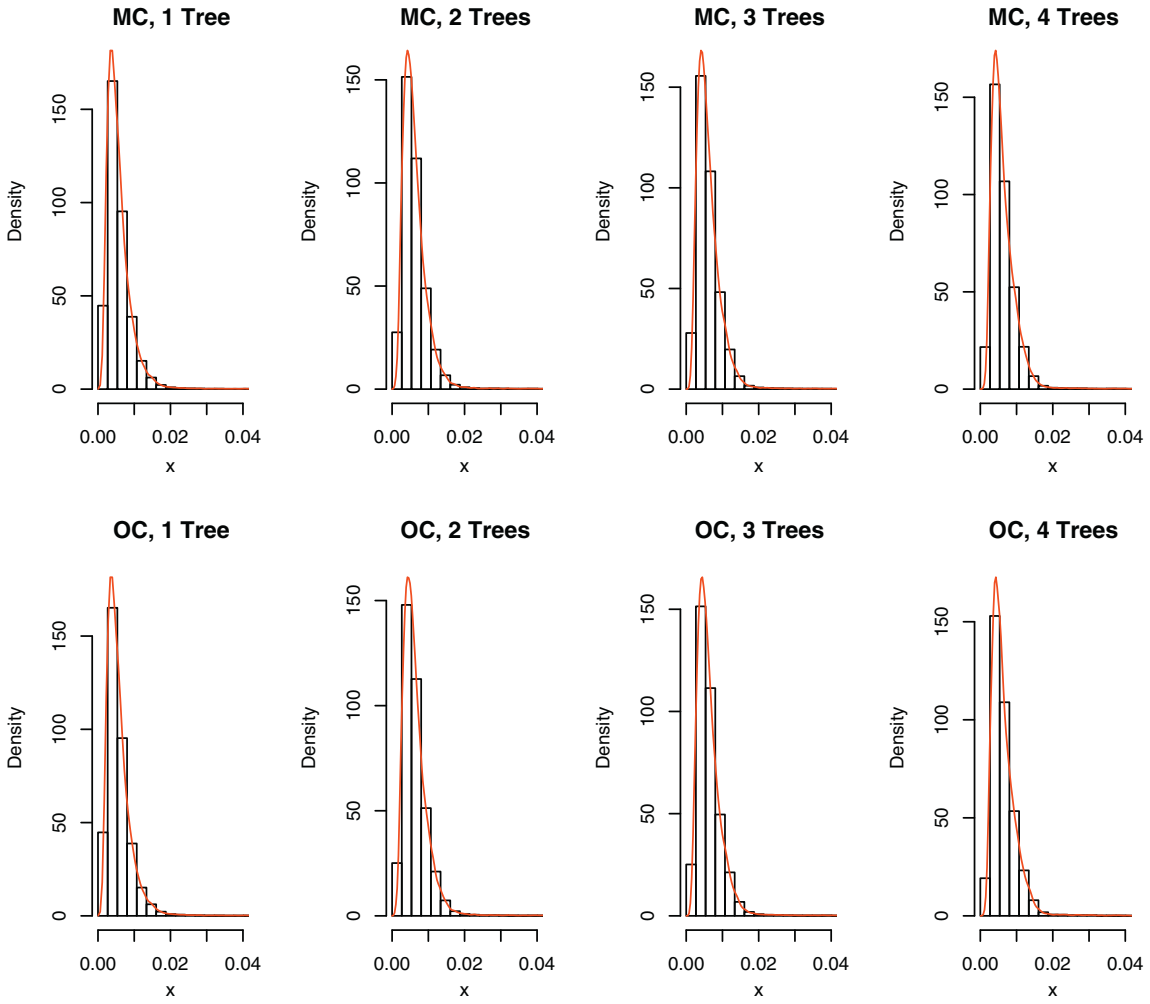


Fig. 5. Log-Normal fitting for valid trees (1 ... 5) and forests for single (OC) and multiple cameras (MC). As we can observe, no matter the forest size or type, the edges weight distributions are reasonably described by a Log-Normal distribution.

where M is the evaluation metric of interest (e.g., root), IPF_i are the calculated forest and the one used as reference (e.g., the forest ground-truth), S_1 is the set of elements in the first forest corresponding to the metric (e.g., set of roots of the first forest) and S_2 is the equivalent for the reference forest. For instance, to obtain the metric *Root*, we calculate the intersection of roots found by algorithm for the first forest with respect to the roots in a second forest (e.g., the reference or ground truth), and normalize by the union of both sets. As an example, consider the algorithm finds three roots $S_1 = (r_1, r_2, r_3)$ in the first forest and two of them turn out to be correct with respect to the reference forest $S_2 = (r_1, r_3)$. Then, the root metric here yields $Root = |S_1 \cap S_2|/|S_1 \cup S_2| = 2/3 = 66.6\,\%$.

6.1. Validation data

We validate the proposed algorithm in two rounds. In the first round, we compare the algorithm to the one presented in [4] using a benchmark we make freely available upon acceptance at http://www.ic. unicamp.br/~rocha/pub/communications.html. In the second round, we compare our method to the method devised in [8] using its two published test benchmarks. We also show an exemplary real case with 75 near-duplicate images.

The corpus we consider in this paper comprises forests of size $|F| \in \{1, \ldots, 5\}$ trees and with 60 semantically similar images in each case. When creating such controlled corpus, we use three different cameras and capture images of three different scenes with three images per camera per scene. For each scene, we consider a near-duplicate scenario in which we have duplicates with different transformation parameters. There are five possible tree topologies for the forest. With all collected images, we create image modifications in such a way we are able to create image descendants in a controlled way up to 60 semantically similar images in each case.

The family of image transformations $\mathcal{T}$ that we consider for applying the image transformations is the same as in [4]: resampling, cropping, affine warping (including rotation, translation, and off-diagonal correction), brightness and contrast adjustments, and lossy compression using the standard lossy JPEG algorithm. In the image processing domain, there are other transformations. However, we strived for accounting for the most common and important ones as they are more frequent in real-world scenarios.

As we aim at evaluating forests instead of single trees (semantically similar images instead of just near-duplicate images), we consider scenarios with a single camera capturing the images and from multiple cameras with similar scene semantics (same content but with small differences in the vantage point, zoom, etc.). In the scenario with multiple cameras, one forest might have images of the same scene with images coming from the same camera and others from different cameras (the images have different acquisition artifacts). Each image produces several near-duplicates representing a tree and with the trees composing the

forest. In the scenario with a single camera, the images representing a scene come from a single camera, each image producing several near-duplicates.

In total, we have 6750 forests. This number refers to two different scenarios (single and multiple cameras), five different forest sizes (1 ... 5), three different scenes, three different cameras, three images per camera, five different tree topologies, and five random variations of parameters when applying the image transformations for creating the image offsprings. Therefore, 6750 forests = $2 \times 3^3 \times 5^3$.

In this paper, we deal with only the JPEG image format, the most widely used image compression format nowadays and present in virtually all digital cameras. In addition, in conversations with our forensics partners in the state and federal policies, we were told that the vast majority of photographic material they apprehend nowadays are in JPEG format.

This controlled corpus is relevant for forensic applications, it will allow researchers in the phylogeny field to benchmark their algorithms and to check their effectiveness for finding the ancestry information, roots of the trees, etc. It is also important for checking the real accuracy of a proposed method comparing its outputs to the ground truth (real tree ancestry relationships). In addition, it serves as a basis for calibrating and fixing parameters for real cases when the algorithm goes operational. In such cases, there is no associated ground truth.

Finally, we also show an example of a resulting phylogeny tree associated with a real case we dealt with in the past regarding a fake and defamatory criminal police record published by the Brazilian newspaper Folha de São Paulo.

6.2. First round

As we previously mentioned, for the first round of analysis, we tested the algorithms with a total number of 6750 forests. The dataset comprises forests of varying size $|F| \in \{1, 2, \ldots, N\}$ trees and with 60 semantically similar images in each case. Particularly, in this paper, we focused on forests of size up to $N = 5$ trees but it can be trivially expanded to more trees. We consider scenarios with a single camera and from multiple cameras but with similar scene semantics.

Table 1 shows the results for the Oriented Kruskal [4] phylogeny algorithm considering a different number of trees per forest. In this case, the algorithm requires the input from the user regarding the number of trees to reconstruct. As we deal with controlled experiments in this case, we feed the algorithm with the correct required parameter k. The algorithm is robust to scenarios with single and different cameras. For instance, with forests with five trees, the algorithm can successfully find the root of such trees in 91.7% of the cases considering a scenario with near-duplicates from semantically similar images coming from multiple cameras.

Table 1
Reconstructing a forest of size $|F| \in \{1, \ldots, 5\}$ trees using the Oriented Kruskal (OK) algorithm [4]. This algorithm requires the input from the user for the size of the forest to reconstruct. Results are relative to the ground truth.

$\|F\|$	Roots	Edges	Leaves	Ancestry
(a) Semantically similar images from a single camera				
Baseline OK($k = \|F\|$) – single camera				
1	0.942	0.815	0.806	0.798
2	0.911	0.793	0.817	0.753
3	0.910	0.822	0.826	0.800
4	0.875	0.821	0.813	0.807
5	0.900	0.786	0.816	0.766
(b) Semantically similar images from multiple cameras				
Baseline OK($k = \|F\|$) – multi camera				
1	0.942	0.815	0.806	0.798
2	0.920	0.792	0.816	0.755
3	0.923	0.822	0.824	0.805
4	0.908	0.820	0.811	0.821
5	0.917	0.788	0.815	0.775

However, the algorithm clearly has a major drawback: it requires the number of trees to look for in the forest. Table 2 shows that if we use the Oriented Kruskal [4] algorithm without knowing the number of trees to reconstruct, its performance decreases with the number of trees in the two most important metrics to consider: roots and ancestry. For edges and leaves, normally a tree reconstruction algorithm behaves similarly for trees and forests since, in the case of forest, there is a difference of only a few edges.

The results shown in Table 2 motivated us to a strategy to automatically reconstruct the forest without any user-provided information.

Fig. 3 depicts the results for Automatic Oriented Kruskal (AOK) algorithm with respect to the baseline proposed in [4]. For instance, AOK is only 2% worse than the baseline when finding the roots of the trees in a forest with five trees. In addition, the algorithm correctly finds the ancestors of all images (parents, grand-parents, grand-grand-parents, etc.) in ≅77% of the cases which represents only a 0.5% decrease when compared with the

Table 2
Reconstructing a forest of size $|F| \in \{1, \ldots, 5\}$ trees using the Oriented Kruskal (OK) algorithm [4] with no information about the size of the forest to reconstruct. Results are relative to the baseline in Table 1. The redder the value the worse the metric while the bluer the better.

OK(k = 1) x Baseline – Single Camera				
$\|F\|$	roots	edges	leaves	ancestry
1	0.00%	0.00%	0.00%	0.00%
2	-48.74%	0.00%	-1.47%	-24.83%
3	-65.05%	0.00%	-2.42%	-35.25%
4	-72.91%	0.12%	-2.83%	-41.51%
5	-78.78%	0.13%	-4.53%	-48.43%

(a) Semantically similar images from a single camera.

OK(k = 1) x Baseline – Multi Camera				
$\|F\|$	roots	edges	leaves	ancestry
1	0.00%	0.00%	0.00%	0.00%
2	-48.80%	0.00%	-1.47%	-25.17%
3	-65.66%	0.00%	-2.31%	-35.03%
4	-74.56%	0.12%	-2.96%	-41.53%
5	-79.61%	0.00%	-4.42%	-48.39%

(b) Semantically similar images from multiple cameras.

Table 3
Reconstructing a forest of size $|F| \in \{1, \ldots, 5\}$ trees using the proposed Automatic Oriented Kruskal (AOK) algorithm. Results are relative to the baseline in Fig. 1. The bluer the value the better.

AOK x Baseline – Single Camera				
$\|F\|$	roots	edges	leaves	ancestry
1	-18.05%	-0.25%	-0.37%	-3.76%
2	-7.14%	-0.25%	-0.24%	-1.06%
3	-3.30%	-0.24%	-0.12%	-0.88%
4	-1.03%	-0.24%	0.00%	0.25%
5	-2.00%	-0.13%	-0.12%	-0.52%

(a) Semantically similar images from a single camera.

AOK x Baseline – Multi Camera				
$\|F\|$	roots	edges	leaves	ancestry
1	-18.05%	-0.25%	-0.37%	-3.76%
2	-8.48%	-0.25%	-0.25%	-1.59%
3	-4.55%	-0.36%	-0.36%	-1.12%
4	-2.42%	-0.24%	-0.12%	-0.61%
5	-2.18%	-0.25%	-0.12%	0.00%

(b) Semantically similar images from multiple cameras.

baseline in Fig. 1. This is a major result of the proposed algorithm since it statistically performs similar to the state-of-the-art approach without requiring any input from the user with respect to the number of trees in the forest. Note also that the algorithm improves the results for finding the roots and ancestors of all trees in the forest without sacrificing the edges and leaves metrics.

6.3. Second round

For the second round of analysis, we evaluate our method against the one proposed in [8]. In their work, the authors propose two controlled test cases. Each test case contains two original images and four near-duplicates (descendants) generated through any combination of compression, histogram stretching, rotation, and scaling operations.

The first test (defined as *easy* by the authors) has two original images (roots) which depict the same scene with a slight difference in perspective and are acquired with different cameras. The second test case (deemed *hard* by the authors) has two original images depicting the very same visual content and are acquired with the same digital camera. In both test cases, the algorithm in [8] builds the same forest resulting in the metrics: *Root* = 0.333, *Edges* = 0.875, *Leaves* = 1.000 and *Ancestry* = 0.667. This means the algorithm finds the root in about 33% of the times or all the ancestry connections in 66.7% of the times.

Our algorithm, in contrast, yields *Root* = 0.750, *Edges* = 0.813, *Leaves* = 0.633, and *Ancestry* = 0.768. Note that our method is more effective for finding the roots and the ancestry relationships showcasing an interesting feature for forensic purposes in which we are interested, for instance, in finding the suspects responsible for breaking copyright laws (roots of the trees) or the chain of suspects involved in an illegal activity (ancestry relationships).

6.4. Real case example

This section shows the resulting phylogeny tree for a total of 75 images related to the case from Section 1.

When we started to analyze the case, we discovered that the image was virally widespread over the internet with hundreds of copies in many different websites, blogs, etc. for at least six months. There were many versions of the questioned image over the internet, which slightly differed from one another. Although these differences were not semantical (they depicted the same content) the differences could fool even well trained detection techniques.

The question then was how to choose the right image for analysis? Equally important, was it possible to hint at who possibly published the image when using associated information such as logs, blog posts, internet provider data, etc.?

This was the turning point for devising approaches to analyze the images and point out which ones were the most probable to be the patient zero (original published files) and which ones were the most probable to be the least interesting and more modified (leaves of an evolution tree).

Using the proposed phylogeny approach, we can infer about the evolutionary process the images underwent overtime and can

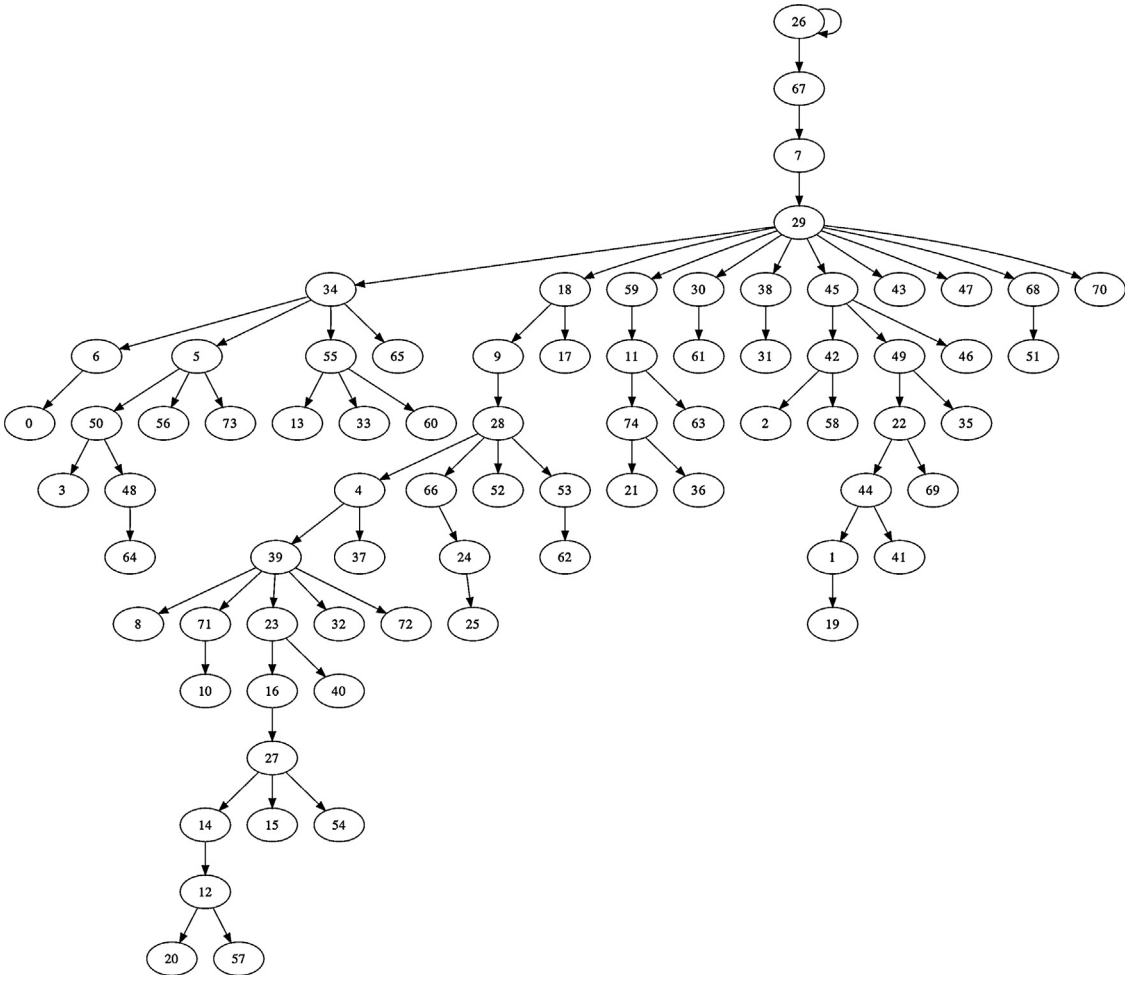


Fig. 6. Phylogeny tree associated with 75 images of the alleged criminal record file published by the Brazilian newspaper Folha de São Paulo as found by our algorithm.

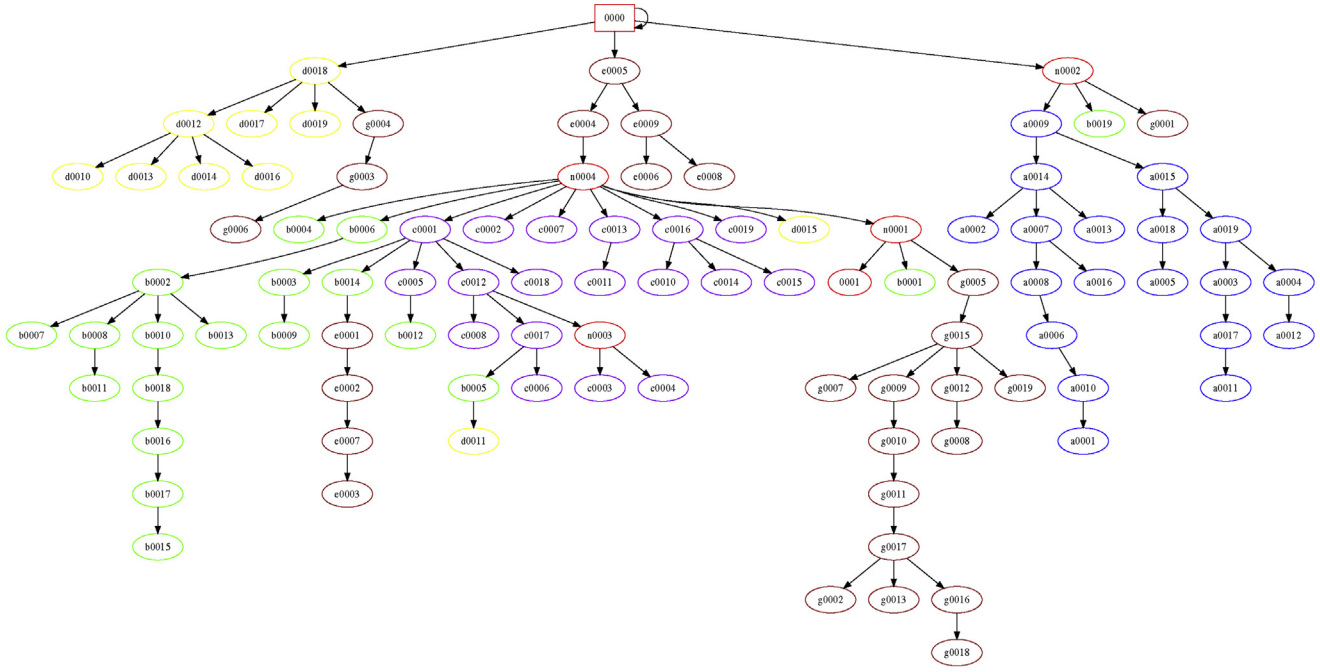
focus the authenticity analyses on fewer candidates. Later on, we could use such tree connections (with auxiliary information not present in the images) to also hint at who released the image online. Fig. 6 shows an example for 75 images. Although we do not have ground truth information for checking if the phylogeny tree is completely correct, it gives us interesting information. For instance, image 14, 12, 20 and 57 are so deep in the tree that probably they have many image modifications and probably are not the originals published online. In the same sense, images 26, 67, 7, 29 and the others at height 5 and 6, are good candidates for being the originals since they contain less image processing artifacts and are closer to the top of the tree.

The sub-trees themselves are also of interest since they tend to put together images that share more history aspects (modification parameters overtime). Consider a subtree such as 39, 8, 71, 23, 32, 72. Such sub-tree might comprise images worth looking at for further exploration and knowledge gathering about the image chain distribution. The images in such sub-tree might have undergone a common image operation that rule them out from being the candidates for root of the tree, for instance. Of course, this is an illustrative example containing a few dozens images but imagine the huge aid phylogeny algorithms can give when dealing with hundreds of thousands of images.

For this particular case, we showed, with convincing arguments, the published image was a fake and it did not came from the Public Archive of São Paulo as the newspaper itself forcefully acknowledged on June 28th, 2009 [39].

On a related case but not directly in the realm of forensics, recently we analyzed the actual evolutionary tree of a famous 2011 photograph captured on May 1st, 2011, by the White House photographer Pete Souza, named *The Situation Room*. The photograph portrays the US President, along with his national security team, receiving live updates of the Operation Neptune Spear, which led to the Osama bin Laden's death. Slightly after its online publishing, this image was heavily reproduced by different communication channels online.

For this experiment, we collected 98 near-duplicate images through Google Images. A quick manual analysis of the images



(a) Phylogeny Tree.

(b) White House Version (c) Balotelli (ID a*) (d) Text Overlay (ID b*) (e) Watermarking (ID c*) (f) Face Swapping (ID d*)

(g) Splicing People (ID e*) (h) Splicing People (ID f*) (i) Splicing Objects (ID g*) (j) Cropping/Zoom (ID h*)

Fig. 7. Phylogeny tree for near-duplicate images portraying the 2011 White House photographer Pete Souza's *The Situation Room*. (For interpretation of the references to color in the text, the reader is referred to the web version of the article.)

show, at least, nine different patterns of image modifications. We have regular near-duplicate images (ID 0*), cases of inserting Italian soccer player Mario Balotelli (ID a*) in the center of the image, text overlay (ID b*), watermarking (ID c*), face swap (ID d*), insertion of elements such as joystick (ID e*), people (ID f*), hats (ID g*), etc.

Fig. 7 depicts the resulting tree using the proposed algorithm as well as the different patterns of modifications present in the set. As expected, the root returned by the algorithm is indeed the original image published online by the White House as we were able to confirm. This image is named here as ID 0000. In addition, the algorithm correctly finds the root of the tree and puts simple near-duplicate images close to it (ID 0*, red ellipses). It also groups most of the cases we discussed above. For example, there are subtrees only containing the Balotelli case (blue ellipses) and face swaps (xxx ellipses) which means the algorithm can reliably identify image similarities and group them accordingly.

The phylogeny algorithm we present here allows us to focus on different aspects of the near-duplicate evolution. In forensics, we often concentrate our attention on the analyses of images on the top of the tree, which supposedly have less modifications or in the evolution structure itself. As for content retrieval, we often focus on identifying the most modified images in the set (leaves) as well as on grouping related modifications on images.

7. Conclusions

In this paper, we introduced a new image phylogeny forest algorithm and compared it to state-of-the-art solutions presented in [4,8].

While Oriented Kruskal [4] provides good results for finding the correct trees in a forest of semantically similar images, it has a major drawback: it needs input from the user regarding the number of trees to seek for. Most of the times in a real scenario, we cannot or do not have such information. The proposed method also outperforms the one presented in [8] for forensics purposes.

Using this paper's solution, we can successfully reconstruct a forest of images with each tree correlating images with a common history background (e.g., with the same original ancestor) without any input from the user. An example of application of this technique would be to automatically find near-duplicate trees among a set of semantically similar images. This would allow the user to trace back one image of interest for forensics purposes without the need to examine too many other semantically similar images.

Future research directions and possibilities to extend this work include expanding our analyses for hundreds of trees in the forest, to include other dissimilarity metrics such as the ones discussed in [8] and other image registration techniques such as the ones discussed in [40–42]. We plan to explore perceptual features and hashes for calculating pixel-wise (dis)similarities between images. We believe this step would become especially important when constructing the phylogeny tree of video documents, where pixel-wise distances are inappropriate.

Finally, there is room also for a deep understanding of the theoretical implications for defining an automatic threshold for finding the trees in a forest of semantically similar documents.

Acknowledgements

We thank the financial support of FAPESP, CNPq, Microsoft and the European Union through the REWIND project. The project REWIND acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Frame-work Programme for Research of the European Commission, under FETOpen grant number: 268478. Finally, we thank Dr. Marina Atsumi Oikawa for helping to proof-read early drafts of this work

References

[1] H. Shen, J. Liu, Z. Huang, C.-W. Ngo, W. Wang, Near-duplicate video retrieval: current research and future trends, ACM Comput. Surv. 45 (4) (2013).
[2] C. Xiao, W. Wang, X. Lin, J.X. Yu, G. Wang, Efficient similarity joins for near-duplicate detection, ACM Trans. Database Syst. 36 (3) (2011), 15:1–15:41.
[3] A. Joly, O. Buisson, C. Frélicot, Content-based copy retrieval using distortion-based probabilistic similarity search, IEEE Trans. Multimedia 9 (2) (2007) 293–306.
[4] Z. Dias, A. Rocha, S. Goldenstein, Image phylogeny by minimal spanning trees, IEEE Trans. Inform. Forensics Secur. 7 (2) (2012) 774–788.
[5] A. Rocha, W. Scheirer, T.E. Boult, S. Goldenstein, Vision of the unseen: current trends and challenges in digital image and video forensics, ACM Comput. Surv. 43 (4) (2011), 26:1–26:42.
[6] Reference omitted due to double-blind submission policies.
[7] L. Kennedy, S.-F. Chang, Internet image archaeology: automatically tracing the manipulation history of photographs on the web, in: ACM MM, ACM, 2008, pp. 349–358.
[8] A.D. Rosa, F. Uccheddua, A. Costanzo, A. Piva, M. Barni, Exploring image dependencies: a new challenge in image forensics, in: Media Forensics and Security II, SPIE, 2010, X1–X12.
[9] Z. Dias, A. Rocha, S. Goldenstein, First steps toward image phylogeny, in: IEEE WIFS, IEEE, 2010, pp. 1–6.
[10] Z. Dias, A. Rocha, S. Goldenstein, Video phylogeny: recovering near-duplicate video relationships, in: IEEE WIFS, 2011, 1–8.
[11] Folha de São Paulo, Grupo de Dilma planejou sequestro de Delfim Netto, Folha de São Paulo, Brazil, 2009, Aprilp. A8.
[12] Microsoft Inc., PhotoDNA – A Technology that Aids in Finding and Removing Some of the "Worst of the Worst" Images of Child Sexual Exploitations from the Internet, 2011 http://www.microsoft.com/en-us/news/presskits/photodna/.
[13] R. Richmond, Facebook's new way to combat child pornography, in: The New York Times, 2011 Online at http://gadgetwise.blogs.nytimes.com/2011/05/19/facebook-to-combat-child-porn-using-microsofts-technology/.
[14] J. Kender, M. Hill, A. Natsev, J. Smith, L. Xie, Video genetics: a case study from youtube, in: ACM MM, 2010, 1253–1258.
[15] E.J. Delp, N. Memon, M. Wu, Digital forensics, IEEE Signal Process. Magaz. 26 (3) (2009) 14–15.
[16] J. Lukas, J. Fridrich, M. Goljan, Digital camera identification from sensor noise sensor, IEEE Trans. Inform. Forensics Secur. 1 (2) (2006) 205–214.
[17] A.E. Dirik, H.T. Sencar, N. Memon, Digital single lens reflex camera identification from traces of sensor dust, IEEE Trans. Inform. Forensics Secur. 3 (3) (2008) 539–552.
[18] B.K. Gunturk, J. Glotzbach, Y. Altunbasak, R.W. Schafer, R.M. Mersereau, Demosaicking: color filter array interpolation, IEEE Signal Process. Magaz. 22 (1) (2005) 44–54.
[19] Y. Huang, N. Fan, Learning from interpolated images using neural networks for digital forensics, in: Intl. Conf. on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, (2010), pp. 1–8.
[20] J. Mao, O. Bulan, G. Sharma, S. Datta, Device temporal forensics: an information theoretic approach, in: IEEE ICIP, Cairo, Egypt, (2009), pp. 1501–1504.
[21] Z. Fan, R. de Queiroz, Identification of bitmap compression history: jpeg detection and quantizer estimation, IEEE Trans. Image Process. 12 (2) (2003) 230–235.
[22] T. Bianchi, A. Piva, Detection of nonaligned double jpeg compression based on integer periodicity maps, IEEE Trans. Inform. Forensics Secur. 7 (2) (2012) 842–848.
[23] S.M.S.M. Tagliasacchi, S. Tubaro, Discriminating multiple jpeg compression using first digit features, in: IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 2012.
[24] M.C. Stamm, W.S. Lin, K.J.R. Liu, Temporal forensics and anti-forensics for motion compensated video, IEEE Trans. Inform. Forensics Secur. 7 (4) (2012) 1315–1329.
[25] C. Barnes, E. Shechtman, D.B. Goldman, A. Finkelstein, The generalized patchmatch correspondence algorithm, in: European Conference on Computer Vision (ECCV), 2010, 29–43.
[26] X. Pan, S. Lyu, Detecting image region duplication using sift features, in: IEEE Intl. Conf. on Accoustics, Speech and Signal Processing (ICASSP), (2010), pp. 1706–1709.
[27] A.C. Popescu, H. Farid, Exposing digital forgeries by detecting traces of re-sampling, IEEE Trans. Signal Process. 53 (2) (2005) 758–767.
[28] T. Bianchi, A. Piva, Analysis of non-aligned double jpeg artifacts for the localization of image forgeries, in: IEEE Intl. Workshop on Information Forensics and Security (WIFS), 2011, 1–6.
[29] M.K. Johnson, H. Farid, Exposing digital forgeries through chromatic aberration, in: ACM Multimedia and Security Workshop, Geneva, Switzerland, (2006), pp. 1–8.
[30] Z. Lin, R. Wang, X. Tang, H.-Y. Shum, Detecting doctored images using camera response normality and consistency, in: Intl. Conf. on Computer Vision and Pattern Recognition (CVPR), New York, USA, (2005), pp. 1087–1092.
[31] M.K. Johnson, H. Farid, Exposing digital forgeries by detecting inconsistencies in lighting, in: ACM Multimedia and Security Workshop, New York, USA, (2005), pp. 1–8.
[32] M.K. Johnson, H. Farid, Exposing digital forgeries in complex lighting environments, IEEE Trans. Inform. Forensics Secur. 2 (3) (2007) 450–461.

[33] H.T. Sencar, N. Memon, Digital Image Forensics: There is More to a Picture than Meets the Eye, Springer, New York, 2013.
[34] H. Farid, Image forgery detection: a survey, IEEE Signal Process. Magaz. 26 (2) (2009) 16–25.
[35] T. Sencar, N. Memon, Overview of State-of-the-Art in Digital Image Forensics, World Scientific Press, Singapore, 2008, Ch. Statistical Science and Interdisciplinary Research.
[36] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (SURF), Comput. Vision Image Understand. 110 (3) (2008) 346–359.
[37] M. Fischler, R. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.
[38] R.E. Tarjan, Efficiency of a good but not linear set union algorithm, J. ACM 22 (2) (1975) 215–225.
[39] Folha de São Paulo, Dilma contrata laudos que negam autenticidade de ficha, Folha de São Paulo, Brazil, 2009, Junep. A12.
[40] A.A. Goshtasby, Image registration methods, in: S. Singh (Ed.), Image Registration, Advances in Computer Vision and Pattern Recognition, Springer, London, 2012, pp. 415–434.
[41] J. Pluim, J. Maintz, M. Viergever, Mutual information based registration of medical images: a survey, IEEE Trans. Med. Imaging 22 (2003) 986–1004.
[42] L. Brown, A survey of image registration techniques, ACM Comput. Surv. 24 (1992) 325–376.