INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 00 (2015) 1–26
DOI: 10.1111/itor.12235

# Distances in multimedia phylogeny

Marina A. Oikawa, Zanoni Dias, Anderson Rocha and Siome Goldenstein

*RECOD Laboratory, Institute of Computing, University of Campinas, Campinas 13083-970, Brazil*
*E-mail: marina.oikawa@ic.unicamp.br [Oikawa]; zanoni@ic.unicamp.br [Dias]; anderson.rocha@ic.unicamp.br [Rocha]; siome@ic.unicamp.br [Goldenstein]*

Received 21 May 2015; received in revised form 29 October 2015; accepted 30 October 2015

**Abstract**

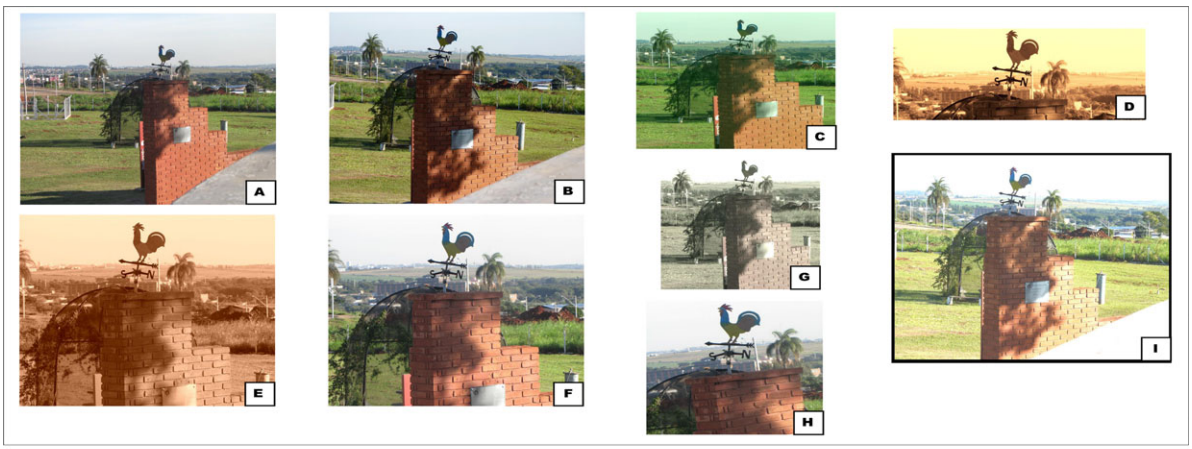
The distance between two objects determines how far apart they are with respect to each other. In this paper, we provide an overview of the distance concept in multimedia phylogeny, a novel research field that aims at discovering the phylogenetic relationships among digital objects that belong to the same population. With applications in digital forensics, copyright enforcement, and security, existing approaches are often based on dissimilarity computations among digital objects in a nonmetric space, but with enough information to correctly reconstruct the underlying relationships of these objects. As we discuss throughout the paper, a proper and well-designed dissimilarity is paramount for differentiating whether two multimedia objects are related and, also, the directionality of such relationship. In phylogeny setups, there is also the additional requirement of low complexity: fast and accurate dissimilarity measures to cope with the massive amount of data we often have to handle.

*Keywords:* computing science; pattern recognition; information systems; tree algorithms

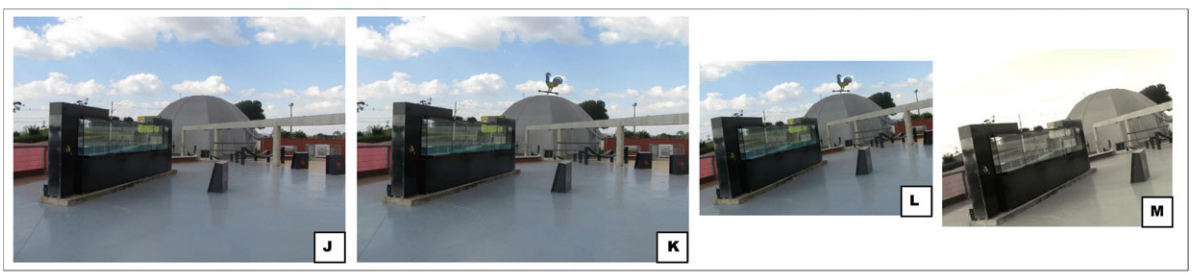
## 1. Introduction

One important task required in many research fields is the selection of a suitable function to measure the distance between the objects under analysis. In clustering, for instance, given a set of unlabeled points, we need to measure how similar or dissimilar they are in order to decide whether the points should be grouped in the same cluster or in different clusters (Xu and Wunsch II, 2005; Kriegel et al., 2009; Jain, 2010). The choice of such function is crucial, as different approaches may result in different clusters. If a similarity function is used, points with high similarity are closer to each other and probably belong to the same cluster. Conversely, objects with high dissimilarity are more distant to each other.

Approaches in near-duplicate detection and recognition (Joly et al., 2007; Maret, 2007; Valle, 2008) also use a similarity function to verify whether two documents are copies of each other (detection), or to retrieve similar copies of one given document (recognition). These approaches




(a) A set of semantically similar images. Using image phylogeny approaches, it is possible to separate images D, E, F, and H from images B and I, since they do not share a common source. In addition, it is also possible to reconstruct the phylogenetic relationship among the images that belong to the same tree.




(b) Image composition and their near duplicates: image K is image J with a splicing of the rooster from image A. Images L and M are near duplicates of K and J, respectively.

Fig. 1. Set of related images.

can be applied in different applications, such as organization of photography collections (Jaimes et al., 2002; Schaffalitzky and Zisserman, 2002), multimedia file matching (Zhang and Chang, 2004), copyright infringement detection (Cheng and Chia, 2010; Ling et al., 2010), and forgery detection (Fridrich et al., 2003; Ke et al., 2004).

In a more complex scenario, it might be necessary to investigate how these near-duplicates are related to each other, and the transformations applied when they were generated. For instance, consider you are given the set of images shown in Fig. 1(a), and asked to separate them into different groups, such that images you think that come from the same source are in the same group. In addition, you are also asked to identify which image is original, and how images belonging to the same group are related, considering the order and type of transformations used for generating each image (geometric transformations, color change, cropping, etc.). As tough as it seems to answer these questions, this scenario is not far from what we find nowadays circulated in the Internet: several cases of tampered digital media, with some level of retouching, content inclusion or removal, and other types of manipulations. Whether these modifications were performed for entertainment

purposes or to forge digital evidence in criminal investigations, the need for the development of methods is undeniable, which can go beyond the simple analysis of whether one document is a copy of the other. The additional knowledge about the history of modifications can give us hints about the original creator and a better understanding of how the content was transformed over time.

For this demand, the "multimedia phylogeny" field was created, aiming at investigating how to infer the relationship among digital objects that belong to the same population (Dias et al., 2012). The main idea is inspired by a similar process observed in Biology, in which an organism inherits characteristics from its ancestors, and can also go through mutations over time, producing new species (Lewin, 1997). If we observe the complete structure related to this population represented by a phylogenetic tree, we are able to identify the ancestral lineage, the descendants of each ancestor, and have an overall idea of their evolution timeline. In this paper, we summarize the approaches developed in multimedia phylogeny, showing how the phylogenetic relationship can be estimated to different types of multimedia objects, also discussing the paramount role of a good dissimilarity function in this scenario.

## 2. Definitions

Recent technological advances have been changing the way digital data are created and consumed in our daily lives. Several cases of digital content are not only created every day all around the world, but also copied, edited, and republished without much effort in multiple channels on the Web. This scenario inevitably presents an alarming situation: besides the many legitimate uses of digital documents, there are also innumerous cases of criminal activities, including copyright infringement, sharing of illegal or abusive contents (e.g., child pornography, a person in a bullying situation, fake and defamatory images of celebrities or politicians), forgeries in scientific publications (Farid, 2006) and digital material used as evidence in court, among others. Therefore, not surprisingly, several research groups have recently dedicated their efforts to the forensic analysis of multimedia data, aiming at understanding their generation process or proving their authenticity (Rocha et al., 2011).

"Multimedia phylogeny" algorithms have emerged as a powerful tool to identify the existent relationship among a set of near-duplicate objects and which transformation parameters were used for creating them (Dias et al., 2010, 2012). The term "near-duplicate" used in this paper has a close relation with the definition provided by Joly et al. (2007), in which a near-duplicate or a copy $\mathcal{D}_n$ is a transformed version of an original document $\mathcal{D}$ (also called "root") that remains recognizable. The "descendant images" are created after applying one or more transformations $\mathcal{T}_{\overrightarrow{\beta}}(\mathcal{D})$, where $\mathcal{T}$ is a set of "tolerated transformations," and $\overrightarrow{\beta}$ represents all possible transformations that an object can go through. Thus, a document $\mathcal{D}_1$ is a copy of $\mathcal{D}$, if $\mathcal{D}_1 = T(D)$, $T \in \mathcal{T}$. These transformations can include several combinations of operators, such as $\mathcal{D}_3 = T_3 \circ T_2 \circ T_1(\mathcal{D})$, $T_\beta = [1, 2, 3] \in \mathcal{T}$. In addition, each document $\mathcal{D}_n$ can originate several other copies, in different branchings. Therefore, although all documents in one tree are copies of $\mathcal{D}$, they are not necessarily copies of each other. The result of this process is a "phylogeny tree," whereby every node represents one digital object, and every branching a transformation or combination of transformations.

In the multimedia phylogeny framework, phylogeny trees are graphically represented by directed acyclic graphs, with weights on each directed edge, pointing the direction of the transformation from the parent to its child. The "edge weights" are essential for the correct reconstruction of the phylogeny trees, being calculated from a "dissimilarity function" $d(\cdot, \cdot)$, which basically yields small values for similar objects and large values for more distinct objects (those that underwent more significant transformations). However, it is noteworthy that in multimedia phylogeny, depending on the media being analyzed, this dissimilarity may or may not be a distance in a metric space, as we will explain in the next sections.

If instead of one tree, there are $m$ trees representing the ancestry relationship in a set of near-duplicate objects, then a "phylogeny forest" is defined. In this case, the set of near-duplicates has $m$-independent subsets, and they are named as "semantically similar images." This happens when there are multiple objects with similar semantic content, but are not directly related to each other, that is, they do not have a common source. It includes, for instance, images or videos from the same scene taken with different cameras or with the same camera but using different settings and in slightly different positions (Dias et al., 2013a; Costa et al., 2014; Oikawa et al., 2016). In case of child pornography, for instance, if several cameras are used for shooting the same scene, from the forensic perspective it is important to identify all cameras and their owners, allowing forensic experts to track down the respective creators and original publishers of this content.

Furthermore, not rarely it is necessary to deal with scenarios whereby an object inherits content from multiple sources (parents). For instance, image montages (the content of each source image is placed in a larger frame, with minimum or no overlapping among them), splicing (a small region of one image is pasted onto another image), blending of images (the content of two or more images are blended through their alpha channels) or combinations of different videos, commonly found in the Internet. These scenarios, referred to as "multiple parenting phylogeny," represents an extension of the original work in image phylogeny forests (IPFs). It aims at finding the relationships not only between images with essentially the same content (near-duplicates), but also those of seemingly unrelated content, with the additional challenge of not having any prior information about the amount of content they share (Oliveira et al., 2014, 2015).

Figures 1 and 2 illustrate all the definitions explained herein. Images D, E, F, and H are near-duplicates and belong to the same tree whose root is image F. On the other hand, images B and I were taken with different cameras and belong to a different tree. Therefore, images that belong to Tree I and Tree II are semantically similar. In Fig. 1(b), image K has a splicing of the rooster from image A on image J configuring a case of multiple parent phylogeny. The corresponding phylogeny tree is presented in Fig. 2 (Tree III), which basically has three types of image families: "host" and "alien" (both source images), and the composition (result of splicing together a portion of the alien image and the host image), each one related to a near-duplicate set.

The representation depicted in Fig. 2 shows how phylogeny trees can be a convenient way to describe the order and chain of transformations used for creating the objects under investigation. However, the reliability of this reconstruction is highly connected to how the edge weights are calculated. Thus, it is of fundamental importance having a proper dissimilarity function to determine how far apart the images are (the amount of transformations that separates them), and hence, their position in the tree. In multimedia phylogeny, this phylogenetic tree is reconstructed using either heuristic-based (Dias et al., 2012, 2013b) or optimum branching (OB) solutions (Dias et al., 2013b). We present an overview of the existent approaches in Fig. 3.

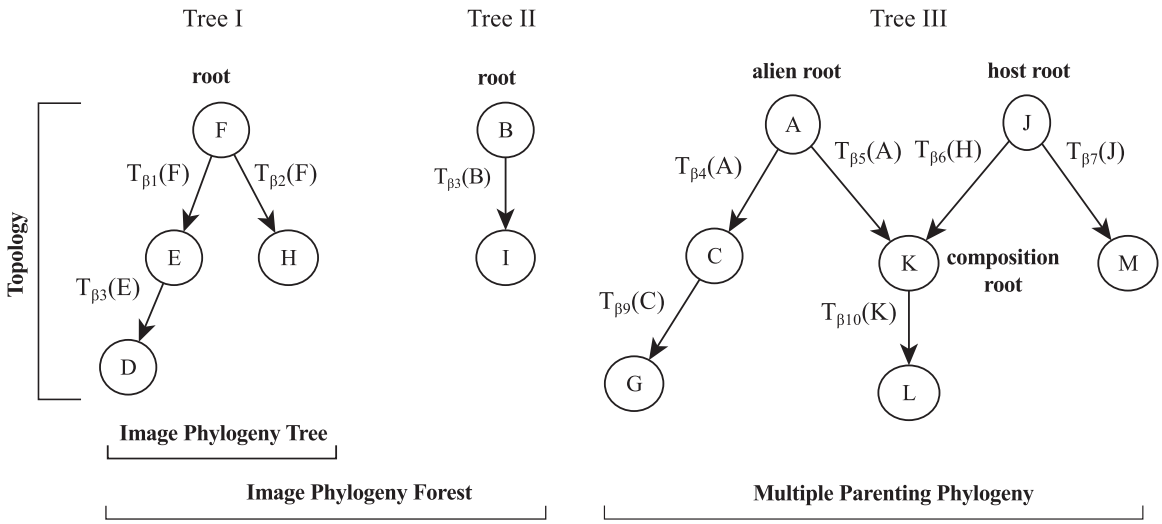



Fig. 2. The reconstructed trees representing the relationship among the images in Fig. 1. Each node corresponds to an image, and the direction of the edges indicates parent–child relationships.

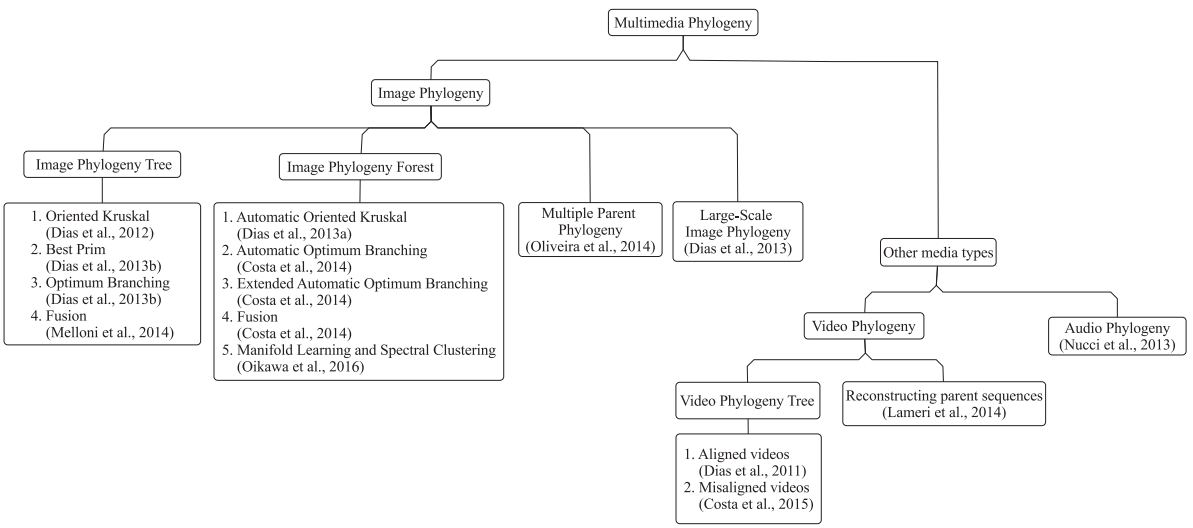



Fig. 3. An overview of multimedia phylogeny approaches.

## 3. Dissimilarity in multimedia phylogeny

Most of the work developed in multimedia phylogeny uses a dissimilarity function to determine how two digital objects are related. Thus, a typical solution consists of first computing the pairwise dissimilarities between all $n$ objects, creating an $n \times n$ dissimilarity matrix $M$, and reconstructing the phylogeny tree using $M$ as input to a minimum spanning tree algorithm.

Given a set $X$ of $n$ objects, and $d(\cdot, \cdot)$ a real function that calculates their dissimilarity, one important discussion is whether $d(\cdot, \cdot)$ meets the criteria of being metric. In this case, for each $x, y, z \in X$, the following properties should be satisfied (O'Searcoid, 2006):

**(P1)** $d(x, y) \geq 0$ with equality if, and only if, $x = y$ (positive property).
**(P2)** $d(x, y) = d(y, x)$ (symmetry).
**(P3)** $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality).

In general, it is desirable to work with distances in the metric space. For instance, consider that $d$ is used in a recognition system, and we find that two arbitrary shapes $\mathcal{O}_1$ and $\mathcal{O}_2$ have high dissimilarity. Then, we compare $\mathcal{O}_2$ to another object $\mathcal{O}_3$, and find their dissimilarity is low. If the triangle inequality holds, $d(\mathcal{O}_1, \mathcal{O}_3) \leq d(\mathcal{O}_1, \mathcal{O}_2) + d(\mathcal{O}_2, \mathcal{O}_3)$, then $d(\mathcal{O}_1, \mathcal{O}_3)$ is high as well, and there is no need to explicitly compare $\mathcal{O}_1$ and $\mathcal{O}_3$. Nonetheless, it is still possible to work with dissimilarity measures that are effective without being metric, as seen in some approaches in classification and computer vision (Jacobs et al., 2000). In machine learning, there is also recent research pointing out that in many cases the performance of recognition algorithms improves when the metric conditions are relaxed (Scheirer et al., 2014).

In special, for image and video phylogeny, due to the nature of the transformations evaluated, the dissimilarity function used is not a distance in a metric space, as it does not satisfy the symmetry and triangle-inequality properties. For the symmetry property, for example, if the family of transformations $\mathcal{T}$ has only the spatial rescaling operation in images, reducing the spatial resolution (downsampling) is different from increasing the spatial resolution (upsampling). When two images $\mathcal{I}_A$ and $\mathcal{I}_B$ have different dimensions, the dissimilarity from $\mathcal{I}_A$ to $\mathcal{I}_B$ would calculate the image residual on $\mathcal{I}_B$'s image dimensions, while the dissimilarity from $\mathcal{I}_B$ to $\mathcal{I}_A$ would calculate the image residual on $\mathcal{I}_A$'s dimensions (Dias et al., 2012). However, this asymmetry is of fundamental importance for the next step (phylogeny tree reconstruction), as it allows the reconstruction algorithms to determine the direction of the modifications (parent–child relationship). If we consider text documents, the same is not true: The edit operations in text documents (word exchange, insertion or removal of typos, adjectives, adverbs, and even full sentences, for instance) are all symmetric, and the functions to calculate their dissimilarity form a metric space. In this case, the reconstruction of text phylogeny trees requires a different strategy, being more challenging than for image and videos, for instance.

Lastly, it is noteworthy that oriented distance functions cannot be applied to the multimedia phylogeny problem, as these functions are used in the analysis of shape optimization problems, targeting problems where evaluation of the geometric properties associated with a set and its boundaries is important (Delfour and Zolésio, 2001). Nonetheless, as we describe in the next sections, the dissimilarity function used in multimedia phylogeny provides enough information to infer the direction of the relationship among the digital objects through the asymmetry of the transformations they underwent.

### *3.1. Image phylogeny*

The calculation of a reliable dissimilarity measure between two images is not a trivial task. In general, the comparison between the images cannot be done by directly performing image subtraction,

for instance. Common approaches for this calculation include first the use of registration algorithms based on feature detection (Zitova and Flusser, 2003), or on the comparison of the statistical dependence between the images (Pluim et al., 2003; Russakoff et al., 2004). In either case, there are several steps to be followed, whose implementation has a large influence on the accuracy of this calculation.

In the multimedia phylogeny literature, there are two main approaches developed to calculate the dissimilarity among a set of related images: (a) a content-based dissimilarity that considers a set of possible image transformations from which one image can generate a descendant, and estimates the cost of these transformations pairwise (Dias et al., 2010, 2012); and (b) a combined approach that also takes into account the content-independent part of the image, related to the peculiarities of the process that produced the image (noise) (Melloni et al., 2014).

### *3.1.1. Content-based dissimilarity*

Dias et al. (2010, 2012) devised a dissimilarity function $d_{img}$ used for comparing each pair of images, based on mapping one image onto another image's domain, and estimating the cost of such operation. Thus, considering $\mathcal{T}$ a family of image transformations, and $T$ a transformation such that $T \in \mathcal{T}$, the dissimilarity $d_{img}$ between two images $\mathcal{I}_i$ and $\mathcal{I}_j$ is defined as

$$d_{img}(\mathcal{I}_i, \mathcal{I}_j) = \min_{T_{\overrightarrow{\beta}}} |\mathcal{I}_j - T_{\overrightarrow{\beta}}(\mathcal{I}_i)|_{\text{comparison method}}\ \mathcal{L}, \tag{1}$$

for all possible values of the parameter $\overrightarrow{\beta}$ in $\mathcal{T}$. Equation (1) calculates the dissimilarity between the best transformation mapping $\mathcal{I}_i$ onto $\mathcal{I}_j$'s domain, according to $\mathcal{T}$. Current approaches consider resampling, cropping, affine transformation, brightness and contrast change, gamma correction, and compression as the set of transformations $\mathcal{T}$. Finally, the comparison between the images can be performed by any point-wise comparison method $\mathcal{L}$ (e.g., sum of squared differences).

In Dias et al.'s (2012) approach, the calculation of the dissimilarity of each pair of images is performed by first finding feature points in each pair of images $(\mathcal{I}_i, \mathcal{I}_j)$ using SURF (Bay et al., 2008). This step gives candidate points to estimate the warping and cropping parameters for image $\mathcal{I}_i$ with respect to $\mathcal{I}_j$ using points filtered by RANSAC (Fischler and Bolles, 1981). With such points, the homography matrix—which maps points from $\mathcal{I}_i$ to $\mathcal{I}_j$—is obtained (as illustrated in Fig. 4). Pixel color normalization parameters are calculated using the color transfer technique proposed by Reinhard et al. (2001). Subsequently, to estimate the cost for compressing $\mathcal{I}_i$ on the domain of $\mathcal{I}_j$, image $\mathcal{I}_i$ is compressed according to $\mathcal{I}_j$'s quantization table. As a final step, both images are uncompressed and compared pixel-wise according to the minimum squared error (MSE) metric. Thus, for objects $\mathcal{I}_1, \ldots, \mathcal{I}_n$ the result is an $n \times n$ asymmetric dissimilarity matrix $M = [d_{img}(\mathcal{I}_i, \mathcal{I}_j)]_{i,j=1,\ldots,n}$, with small values for similar images and large values for more distinct images. This same computation is performed for other image phylogeny approaches, such as IPFs (Dias et al., 2013a; Costa et al., 2014; Oikawa et al., 2016), multiple parenting phylogeny (Oliveira et al., 2014, 2015), and large-scale image phylogeny (Dias et al., 2013c).

It is noteworthy, however, that all these steps may introduce small errors in the dissimilarity calculation. For instance, errors related to the geometric registration process may appear, incorrectly increasing or decreasing the dissimilarity between the images. The reconstruction of the phylogeny tree is affected as a consequence, since in some cases it might cause the inversion of the parent–child relationship. To reduce the influence of this flaw in the dissimilarity calculation, Costa et al. (2014)

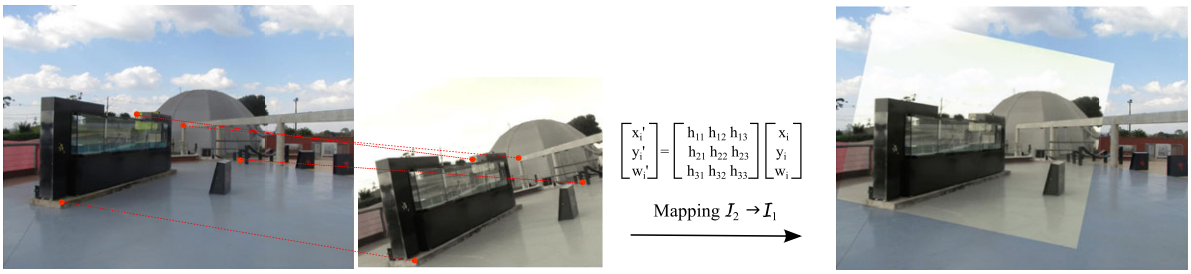



Fig. 4. Mapping image $\mathcal{I}_2$ into image $\mathcal{I}_1$'s domain. Robust feature points are calculated for each image, and used for finding the homography matrix relating them. Once the mapping is performed, both images can be compared pixel-wise within the region they overlap.

proposed to apply different amounts of perturbations through noise addition to the dissimilarity matrix $M$, generating $k$ different variations of $M$, and using a votes scheme to reconstruct the resulting $k$ phylogeny trees. In their approach, $k = 100$, which was chosen in such a way that it was enough to analyze the consistency among the results and achieve statistical significance in the analysis. With this approach, it was possible to introduce robustness to statistical errors in the estimation of the image dissimilarities, taking away some of the importance of extra fine tuning of the dissimilarity calculation.

### *3.1.2. Combined dissimilarity*

According to De Rosa et al. (2010), any image can be described as the composition of two parts: one conveying the semantic information related to the real scene (component), and a content-independent part representing the peculiarities of the process that produced the images (randomness). The first part is responsible for instantiating a family of image processing functions that can be applied from one image to the other (color matching, compression, or image registration, for instance), while the second component is used for calculating the correlation coefficient between the two images. Since the component part can be very similar to many near-duplicate images, they consider two images as dependent (one being produced by the other) only if some form of similarity exists between their content-independent components. A correlation matrix $C$ is used for constructing the dependency graph relating the images, whose nodes represent the images and edges represent their correlation values. Lastly, loops and edges below a specified threshold are eliminated from the final graph.

Inspired by this approach, Melloni et al. (2014) proposed the construction of an additional dissimilarity matrix using the content-independent part of the image, but instead of using correlation, they apply the same approach proposed by Dias et al. (2012). Thus, two dissimilarity matrices are constructed: $d'_{img}(\mathcal{I}_i, \mathcal{I}_j)$, which is the same as described in Equation (1), and $d''_{img}(\mathcal{I}''_i, \mathcal{I}''_j)$, where $\mathcal{I}''$ indicates the randomness part of the image. The randomness part is obtained by subtracting the denoised image from the input image using the wavelet-based denoising approach proposed by Mihcak et al. (1999). Using both matrices, this approach aims at reducing the ambiguity in determining the parent of an image $\mathcal{I}_i$ that has low dissimilarity with more than one image.

To reconstruct the image phylogeny tree (IPT), a ranking value, $r_j$, is introduced to define in which order the dissimilarity matrix is scanned. $r_j$, known as "ordered oriented Kruskal" (OOK-$D^A$-$D^B$), merges information from two different dissimilarity matrices: $D^A$ (used for computing the ranking) and $D^B$ (used for estimating the IPT). In this case, $D^A$ and $D^B$ can be any combination of $d'_{img}$ and $d''_{img}$. The ranking value is defined as

$$r_j = \frac{min_1\left(D^A_{i,j}\right)}{min_2\left(D^A_{i,j}\right)} \tag{2}$$

for every column of $D^A$, where $min_1$ and $min_2$ are the first and second minimum values computed on the $j$th column of $D^A$, respectively. Low values of $r_j$ indicate images that are clearly dissimilar to only one parent. Then, the algorithm scans $D^B$, and selects the column $j$ with the lowest $r_j$ value. Image $\mathcal{I}_j$ is connected as child of the image with the lowest dissimilarity, if this does not create loops. Iteratively, the whole IPT is built.

A variation of this approach was also proposed, which is known as "recursively OOK-$D^A$-$D^B$" (ROOK-$D^A$-$D^B$), in which the vector $r_j$ is recomputed at every iteration, allowing to update its values to take into account links that would cause loops. Both approaches, OOK-$D^A$-$D^B$ and ROOK-$D^A$-$D^B$, can also work combined with other algorithms, by switching from one algorithm to another after some iterations. For instance, the best results reported by Melloni et al. (2014) were obtained using oriented Kruskal (OK) (Dias et al., 2010, 2012) as the second algorithm, since OOK and ROOK reconstruct well the nodes during the first iterations, while OK performs better later on.

### 3.2. Video phylogeny

Similar to images, video files can undergo several transformations, creating new descendants. However, this scenario brings more challenges, as we need to deal with spatial and temporal alignment of the videos. To simplify the problem, Dias et al. (2011) proposed an initial approach to video phylogeny, considering only temporally aligned videos with the same number of frames, all of them encoded with the same compression scheme. Thus, considering $T_{\overrightarrow{\beta}}$, a video transformation from a family $\mathcal{T}$, Dias et al. (2011) defined the dissimilarity function between two videos $\mathcal{V}_i$ and $\mathcal{V}_j$ as

$$d_{vid}(\mathcal{V}_i, \mathcal{V}_j) = \min_{T_{\overrightarrow{\beta}}} |\mathcal{V}_j - T_{\overrightarrow{\beta}}(\mathcal{V}_i)|_{\text{comparison method}}\, \mathcal{L}, \tag{3}$$

for all possible values of $\overrightarrow{\beta}$ that parametrizes $\mathcal{T}$. Similar to image phylogeny, Equation (3) measures the amount of residual between the best transformations of $\mathcal{V}_i$ to $\mathcal{V}_j$, according to the family of operations $\mathcal{T}$, and $\mathcal{V}_j$ itself. In their approach, $\mathcal{T}$ includes resampling, cropping and brightness, contrast, and gamma adjustments. It is considered that these operations are equally performed across the video frames, and the videos are compressed using the standard H.264 compression algorithm.

To estimate $\overrightarrow{\beta}$, Dias et al. (2011) first select $f$ frames from both videos considering they are temporal coherent. Then, corresponding points between $\mathcal{V}_i$ and $\mathcal{V}_j$ are calculated using SURF (Bay et al., 2008). Each frame $f_i^k \in \mathcal{V}_i$ is directly compared to frame $f_j^k \in \mathcal{V}_j$. Then, the affine warping

transformation parameters (color correction, cropping, and resampling operations) are robustly estimated for video $\mathcal{V}_i$ with respect to $\mathcal{V}_j$ taking the corresponding points into account and using RANSAC (Fischler and Bolles, 1981) for each pair of frames. In addition, the color channels of a frame $f_i^k \in \mathcal{V}_i$ are normalized according to the mean and variance of the corresponding frame $f_j^k \in \mathcal{V}_j$. Finally, the dissimilarity between both videos, considering frame point-wise dissimilarity, is calculated using MSE as the comparison method $\mathcal{L}$.

More recently, Costa et al. (2015) proposed an extension of this approach, following the same pipeline presented by Dias et al. (2011), but accommodating the case of time clipped, misaligned, and compressed videos. Videos are aligned using the method proposed by Lameri et al. (2014), and frame registration is performed considering geometric registration, color correction, and coding matching. To handle the case of temporal clipped videos, the frame-wise dissimilarity was calculated considering the average of the corresponding frames from the aligned subsequences of $\mathcal{V}_i$ and $\mathcal{V}_j$. However, the following constraint is imposed: one video $\mathcal{V}_i$ cannot be parent (or ancestor) of another video $\mathcal{V}_j$ if $||frames(\mathcal{V}_i)|| < ||frames(\mathcal{V}_j)||$. In addition, it is considered that all video duplicates were created using the same frame rate and there are no cases of videos composition (multiple parenting).

In a related approach, an attempt to reconstruct parent sequences given a set of partially overlapped near-duplicate video shots used in other sequences was proposed by Lameri et al. (2014). First, a robust hash algorithm is used for detecting whether a group of frames is near-duplicate of another group of frames. Then, an algorithm to automatically find near-duplicate matchings between multiple parts of multiple sequences was developed. Finally, all near-duplicate parts are temporally aligned to reconstruct the parent sequence. Although the goal is not the reconstruction of a complete phylogeny tree, the use of the proposed method allows the study of how content is reused and how to recover original content when it is no longer available.

### 3.3. Audio phylogeny

Going beyond the analysis of conventional multimedia objects, Nucci et al. (2013) proposed an approach for analysis of near-duplicate audio tracks. Following the same pipeline of previous approaches, the goal is to infer the structure of modifications and the operations audio tracks have gone through. Thus, considering two audio tracks $S_A$ and $S_B$, the dissimilarity value is computed as follows:

$$d_{aud}(S_A, S_B) = \arg\min_{\overrightarrow{\beta}} \mathcal{L}(S_B, T_{\overrightarrow{\beta}}(S_A)), \tag{4}$$

where $T_{\overrightarrow{\beta}}$ approximates the sequence of operators applied to $S_A$ to get $S_B$, and the sample-wise measure $\mathcal{L}$ used is the "signal-to-noise ratio" (SNR):

$$\mathcal{L}(S_B, T_{\overrightarrow{\beta}}(S_A)) = 20\log_{10}\left(\frac{\|T_{\overrightarrow{\beta}}(S_A)\|}{\|S_B - T_{\overrightarrow{\beta}}(S_A)\|}\right)^2. \tag{5}$$

In their approach, the set $\mathcal{T}$ of operations considered are perceptual audio coding—trim (removal/trailing parts of a track) and fading. At each step, the method estimates the most likely

operator and the corresponding parameter, and returns the SNR. This is repeated twice (double-step dissimilarity), although it can be generalized to a larger number of steps. The algorithm may finish if a perfect match is found (when $SNR > 65$ dB, tuned manually).

## 4. Reconstruction algorithms

Once the dissimilarity matrix is calculated, different tree reconstruction algorithms can be used for inferring the phylogeny related to the near-duplicate documents. In the next sections, we provide more details regarding the main approaches presented in the literature to reconstruct phylogeny trees and forests.

### *4.1. Oriented Kruskal*

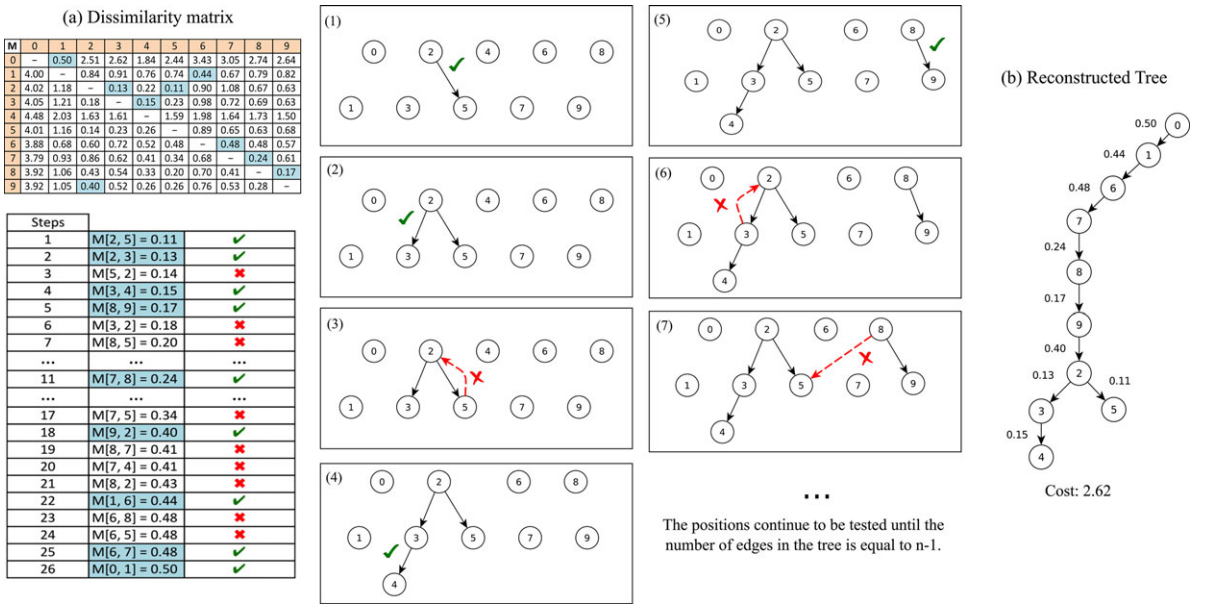
The dissimilarity matrix $M$ calculated for all pairs of images can be seen as a complete, directed graph with weights on each edge. To reconstruct the phylogeny tree associated to $M$, Dias et al. (2010, 2012) proposed to use the classic Kruskal's minimum spanning tree algorithm (Kruskal, 1956) adapted for oriented graphs, which was named OK. This algorithm does not assume that the root of the tree is known beforehand. It finds the root and builds the oriented tree in a single execution.

Let $M$ be the dissimilarity matrix relating $n$ near-duplicate images; the OK algorithm starts considering each near-duplicate as one root of the phylogeny tree. Then, all positions in $M$ are sorted in a nondecreasing order, and each position $M[i, j]$ is evaluated according to this sorted order. To decide which edges will be added to the tree, two tests are performed. First, the algorithm checks if the position connects two disjoint trees. If so, then it checks if the endpoint is a root. When these conditions are fulfilled, the algorithm joins the different trees in a new tree. The algorithm stops when the number of edges added in the final tree is equal to $n - 1$.

In Fig. 5, we illustrate the execution of OK using a toy example considering $n = 10$ near-duplicate images, related by the dissimilarity matrix in Fig. 5(a). After sorting all $M[i, j]$ positions, the algorithm first selects the position $M[2, 5]$ with the lowest dissimilarity value. Since this position connects two disjoint trees and the endpoint is a root, then this position is selected to be added as an edge in the final tree. The same happens with the position $M[2, 3]$. Next, the algorithm tests the position $M[5, 2]$, which is discarded since it does not join different trees. The algorithm proceeds with the selection of positions $M[3, 4]$ and $M[8, 9]$, and discarding position $M[3, 2]$ (as it does not join different trees) and position $M[8, 5]$ (as the endpoint is not a root). This process continues, until the number of edges added in the tree reaches nine, resulting in the tree shown in Fig. 5(b).

### *4.2. Best Prim (BP)*

As a second approach to reconstruct phylogeny trees, Dias et al. (2013b) proposed a heuristic based on the Prim minimum spanning tree algorithm (Prim, 1957), which was named best Prim (BP). Similar to OK, the algorithm initially receives the dissimilarity matrix $M$. Then, it builds $n$ different




| M | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | – | 0.50 | 2.51 | 2.62 | 1.84 | 2.44 | 3.43 | 3.05 | 2.74 | 2.64 |
| 1 | 4.00 | – | 0.84 | 0.91 | 0.76 | 0.74 | 0.44 | 0.67 | 0.79 | 0.82 |
| 2 | 4.02 | 1.18 | – | 0.13 | 0.22 | 0.11 | 0.90 | 1.08 | 0.67 | 0.63 |
| 3 | 4.05 | 1.21 | 0.18 | – | 0.15 | 0.23 | 0.98 | 0.72 | 0.69 | 0.63 |
| 4 | 4.48 | 2.03 | 1.63 | 1.61 | – | 1.59 | 1.98 | 1.64 | 1.73 | 1.50 |
| 5 | 4.01 | 1.16 | 0.14 | 0.23 | 0.26 | – | 0.89 | 0.65 | 0.63 | 0.68 |
| 6 | 3.88 | 0.68 | 0.60 | 0.72 | 0.52 | 0.48 | – | 0.48 | 0.48 | 0.57 |
| 7 | 3.79 | 0.93 | 0.86 | 0.62 | 0.41 | 0.34 | 0.68 | – | 0.24 | 0.61 |
| 8 | 3.92 | 1.06 | 0.43 | 0.54 | 0.33 | 0.20 | 0.70 | 0.41 | – | 0.17 |
| 9 | 3.92 | 1.05 | 0.40 | 0.52 | 0.26 | 0.26 | 0.76 | 0.53 | 0.28 | – |

| Steps | | |
|---|---|---|
| 1 | M[2, 5] = 0.11 | ✔ |
| 2 | M[2, 3] = 0.13 | ✔ |
| 3 | M[5, 2] = 0.14 | ✖ |
| 4 | M[3, 4] = 0.15 | ✔ |
| 5 | M[8, 9] = 0.17 | ✔ |
| 6 | M[3, 2] = 0.18 | ✖ |
| 7 | M[8, 5] = 0.20 | ✖ |
| … | … | … |
| 11 | M[7, 8] = 0.24 | ✔ |
| … | … | … |
| 17 | M[7, 5] = 0.34 | ✖ |
| 18 | M[9, 2] = 0.40 | ✔ |
| 19 | M[8, 7] = 0.41 | ✖ |
| 20 | M[7, 4] = 0.41 | ✖ |
| 21 | M[8, 2] = 0.43 | ✖ |
| 22 | M[1, 6] = 0.44 | ✔ |
| 23 | M[6, 8] = 0.48 | ✖ |
| 24 | M[6, 5] = 0.48 | ✖ |
| 25 | M[6, 7] = 0.48 | ✔ |
| 26 | M[0, 1] = 0.50 | ✔ |

Fig. 5. Phylogeny tree of $n = 10$ near-duplicate objects, reconstructed using OK. After the dissimilarity values are sorted in a nondecreasing order, the algorithm starts the construction of the tree taking each position at a time and performing the required tests to ensure the tested position can be an oriented edge in the final tree. The algorithm stops when the number of edges in the tree is $n - 1$.

trees using an auxiliary method called "oriented Prim," in which each of the $n$ nodes is considered as root once. After evaluating all possible roots, the algorithm chooses the one with the lowest reconstruction cost (sum of dissimilarities) as the final phylogeny tree.

Considering the same dissimilarity matrix given in Fig. 5(a), we illustrate this reconstruction algorithm in Fig. 6. In the first iteration, oriented Prim reconstructs the tree considering node 0 as root, whose cost is 2.73. Next, it reconstructs the tree having node 1 as root, whose cost is 6.02. This process continues until the algorithm finds the last possible tree rooted at vertex 9, whose cost is 6.63. Since the tree with node 0 as root has the lowest cost, it is chosen as the final IPT.

### *4.3. Optimum branching*

Dias et al. (2013b) also proposed an algorithm based on the OB algorithm (Chu and Liu, 1965; Edmonds, 1967; Bock, 1971). In this case, instead of choosing the lowest cost edge at each stage to find the minimum branching as done by the OK, the OB considers the whole dissimilarity matrix to make decisions about the phylogeny reconstruction, always finding one optimum global solution, which contributed to improve the tree reconstruction results.

Considering $G$ as the input graph built from the dissimilarity matrix $M$, and $r$ as one node of the phylogeny tree, the first step in this algorithm consists of assuming node $r$ as a possible root, as in the oriented Prim algorithm. For each node $v_i$ in the graph other than $r$, the edge arriving at $v_i$ with the lowest cost is selected. If there is no circuit, this branching is returned. Otherwise, the algorithm

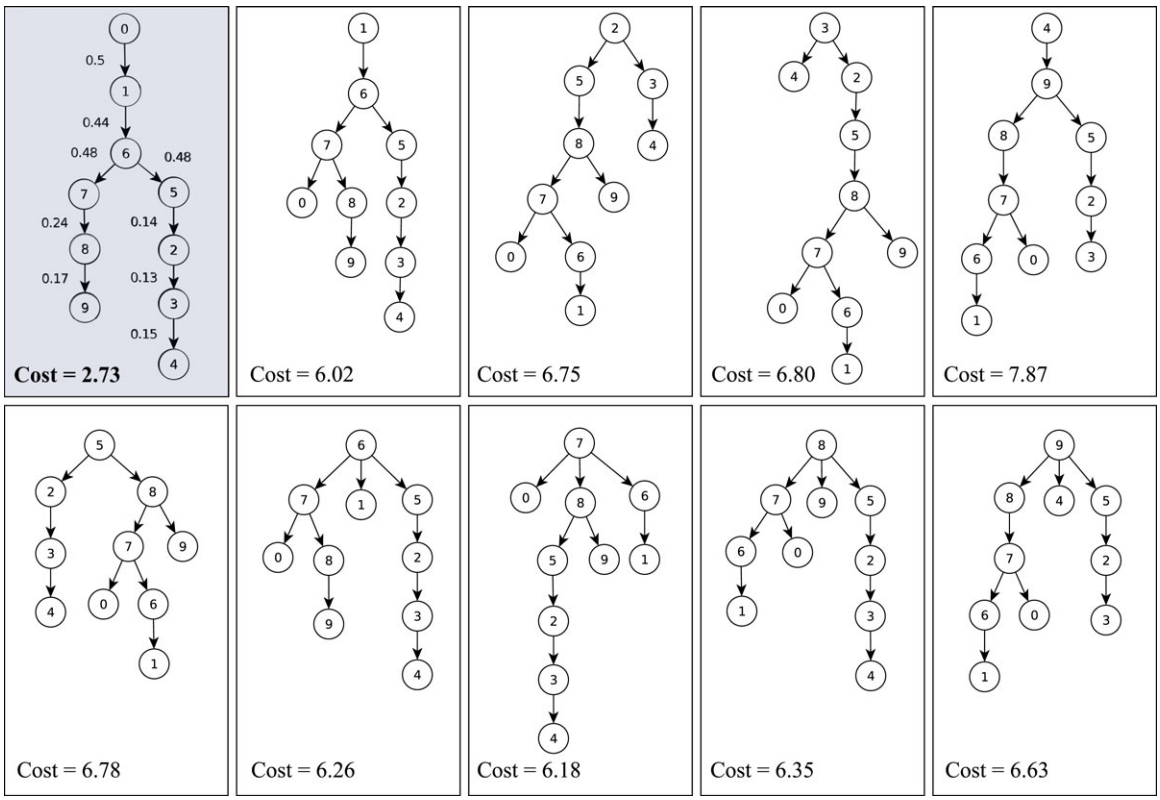



Fig. 6. Phylogeny tree reconstructed using the best Prim algorithm. Oriented Prim, an auxiliary method, is used for calculating the cost of building a tree considering each node as root once. After evaluating all possible roots, the algorithm chooses which tree had the lowest reconstruction cost. In this example, the tree with node 0 as root is chosen as the final reconstructed tree.

deals with it by creating a dummy node $v_d$. In the example of Fig. 7(a), $r = 1$, and the solid-line edges highlight the selected edges with the lowest cost. However, since there is a circuit (indicated in grey), $v_d$ is created as shown in Fig. 7(b).

To cope with the circuit, for each edge connecting a node $v_x$ outside the circuit and a node $v_y$ in the circuit, its weight is updated by considering the edge weight $w(v_x, v_y)$ plus the lowest edge weight in the circuit minus the edge weight of the edge arriving at $v_y$. In Fig. 7, if we consider $v_x = 2$, and $v_y = 4$, then $w(2, 4) = 5$. Since the lowest edge weight inside the cycle is $w(5, 6) = 3$, and the weight of the edge that arrives at $v_y$ is $w(6, 4) = 4$, then we can update the edge weight as $w(2, v_d) = 5 + 3 - 4 = 4$. This procedure is repeated for all the remaining nodes connected to $v_d$. After processing all nodes within and outside the circuit, the algorithm recursively finds the optimal branching in the graph. Next, to cope with the circuit itself, the edge in the current OB connecting a node $v_x$ to the dummy node $v_d$ is discarded, and this edge is updated with the correct one in the original graph. For instance, in Fig. 7(c), the edge $(2, v_d)$ is replaced by edge $(2, 4)$ in the final branching. In addition, the algorithm updates all edges in the OB within the circuit that do not arrive in $v_y$ (the edge $(6, 4)$ is removed). Finally, the algorithm updates possible edges from

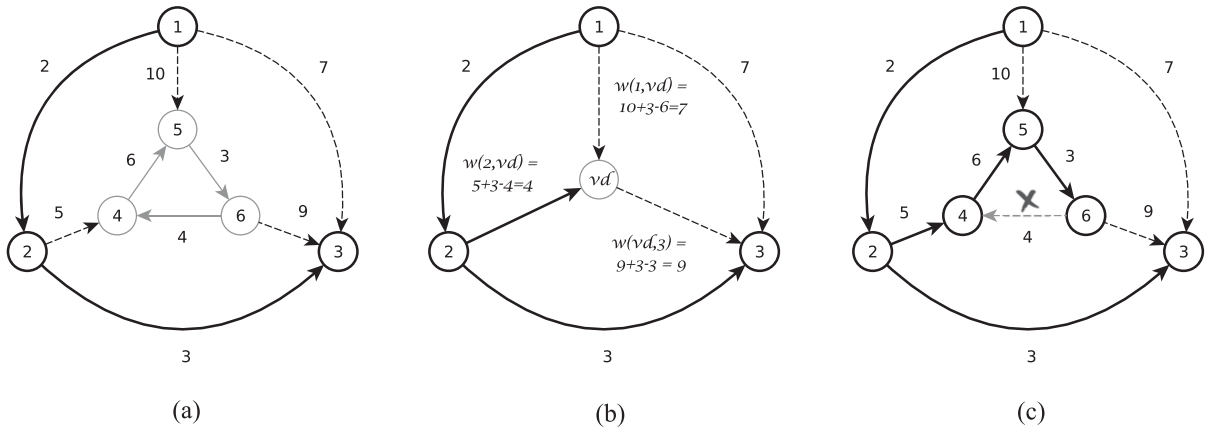



Fig. 7. Simulation of OB algorithm. (a) Considering node $r = 1$, the edges with lowest cost are selected. (b) To deal with the circuit, a dummy node $v_d$ is created, and the edge weights of the nodes connected to $v_d$ are updated. (c) After processing all nodes within and outside the circuit, the algorithm recursively finds the OB.

nodes within the circuit to outside, which are part of the OB. The algorithm ends with the graph in Fig. 7(c) with the solid lines highlighting the edges that belong to the OB.

### *4.4. Automatic OK*

For reconstructing phylogeny forests, similar strategies used for phylogeny trees can be used. The main difference is the need to find the number of existent trees in the set of semantically similar images, and correctly separate the groups of images belonging to each tree. The first approach was developed by Dias et al. (2013a), through the extension of OK algorithm to the automatic OK (AOK), aiming at automatically dealing with sets of images from different sources without any user intervention.

In the AOK algorithm, given a dissimilarity matrix $M$ built upon a set of $n$ semantically similar images, each image in the set is initially considered a root of a tree. Then, the algorithm starts processing good edges (the ones with low weight) to connect trees. To decide which edges will be added to the forest, the following tests are performed. First, the algorithm checks if the nodes connect two different trees, also checking if it does not create a loop. In addition, after having enough number of edges added to the forest (in this implementation, this happens when the number of added edges reaches more than half of possible edges), the algorithm starts to calculate the current standard deviation of such selected edges. By keeping track of the variance of processed edges, the algorithm only adds a new one to the forest if the difference between the current edge and last edge added to the forest is lower than $k$ times the standard deviation $\sigma$ of the edges processed up to that point. This parameter $k$ is calculated *a priori* based on the arc-weight distribution of some example trees (by default, $k = 2$, determined after some experiments). The algorithm stops when either the weight of the new candidate edge is much higher than the weights it already accepted or when it already processed $n - 1$ edges.

(a) Dissimilarity matrix

| M | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | – | 25.79 | 29.71 | 2.39 | 1.09 | 20.80 | 26.05 | 20.77 | 19.16 | 1.79 | 29.31 | 1.41 | 29.63 | 24.98 | 29.56 |
| 1 | 25.89 | – | 5.08 | 33.51 | 30.82 | 1.19 | 2.36 | 2.73 | 1.23 | 27.47 | 7.10 | 22.52 | 6.23 | 5.05 | 7.06 |
| 2 | 26.57 | 12.52 | – | 33.51 | 30.82 | 10.72 | 13.66 | 10.72 | 12.43 | 28.34 | 1.34 | 23.22 | 0.58 | 0.83 | 0.97 |
| 3 | 4.80 | 25.79 | 29.71 | – | 4.56 | 20.80 | 26.05 | 20.77 | 26.34 | 1.12 | 29.31 | 1.65 | 29.63 | 29.23 | 29.56 |
| 4 | 1.58 | 25.79 | 29.71 | 2.03 | – | 20.80 | 26.05 | 20.77 | 26.34 | 1.19 | 29.31 | 0.76 | 29.63 | 36.10 | 29.56 |
| 5 | 30.91 | 3.50 | 5.23 | 33.51 | 30.82 | – | 4.09 | 0.46 | 3.32 | 34.68 | 7.16 | 28.06 | 6.18 | 5.19 | 7.17 |
| 6 | 30.91 | 2.45 | 5.72 | 33.51 | 30.82 | 1.84 | – | 2.91 | 5.47 | 34.68 | 7.57 | 28.06 | 6.61 | 8.06 | 7.57 |
| 7 | 30.91 | 3.55 | 5.21 | 33.51 | 30.82 | 0.77 | 4.07 | – | 3.33 | 34.68 | 7.13 | 28.06 | 9.89 | 5.17 | 7.13 |
| 8 | 24.27 | 1.70 | 5.46 | 33.51 | 30.82 | 1.37 | 5.88 | 2.80 | – | 29.56 | 7.32 | 24.03 | 6.32 | 5.46 | 7.28 |
| 9 | 4.56 | 25.79 | 29.71 | 0.91 | 4.28 | 20.80 | 26.05 | 20.77 | 22.23 | – | 29.31 | 1.42 | 29.63 | 24.13 | 29.56 |
| 10 | 24.27 | 12.50 | 2.19 | 33.51 | 30.82 | 10.69 | 13.64 | 10.70 | 12.43 | 24.79 | – | 20.57 | 1.89 | 1.97 | 0.87 |
| 11 | 4.34 | 25.79 | 29.71 | 1.65 | 4.07 | 18.39 | 26.05 | 20.77 | 25.62 | 1.33 | 29.31 | – | 29.63 | 18.89 | 29.56 |
| 12 | 30.91 | 12.57 | 1.13 | 33.51 | 30.82 | 10.70 | 13.66 | 12.05 | 12.41 | 34.68 | 1.05 | 28.06 | – | 0.94 | 0.47 |
| 13 | 26.57 | 14.00 | 1.09 | 33.51 | 30.82 | 11.37 | 15.53 | 11.52 | 13.93 | 28.34 | 1.96 | 23.22 | 1.14 | – | 1.74 |
| 14 | 30.91 | 12.52 | 2.04 | 33.51 | 30.82 | 10.71 | 13.66 | 10.72 | 12.43 | 34.68 | 0.58 | 28.06 | 1.75 | 1.87 | – |

(b) Edge values test

| M | limit = 2 x SD | condition | |
|---|---|---|---|
| M[5, 7] = 0.46 | ∞ | – | ✔ |
| M[12, 14] = 0.47 | ∞ | – | ✔ |
| M[14, 10] = 0.58 | ∞ | – | ✔ |
| M[2, 12] = 0.58 | ∞ | – | ✔ |
| M[4, 11] = 0.76 | ∞ | – | ✔ |
| M[7, 5] = 0.77 | ∞ | – | ✖ |
| M[2, 13] = 0.83 | ∞ | – | ✔ |
| M[10, 14] = 0.87 | ∞ | – | ✖ |
| M[9, 3] = 0.91 | ∞ | – | ✔ |
| M[12, 13] = 0.94 | ∞ | – | ✖ |
| M[2, 14] = 0.97 | ∞ | – | ✖ |
| M[12, 10] = 1.05 | ∞ | – | ✖ |
| M[13, 2] = 1.09 | ∞ | – | ✖ |
| M[0, 4] = 1.09 | ∞ | – | ✔ |
| M[3, 9] = 1.12 | ∞ | – | ✖ |
| M[12, 2] = 1.13 | ∞ | – | ✖ |
| M[13, 12] = 1.14 | ∞ | – | ✖ |
| M[4, 9] = 1.19 | 0.45 | 0.10 < 0.45 | ✔ |
| M[1, 5] = 1.19 | 0.53 | 0.00 < 0.53 | ✔ |
| M[1, 8] = 1.23 | 0.57 | 0.04 < 0.57 | ✔ |
| ... | ... | ... | ✖ |
| M[1, 6] = 2.36 | 0.60 | 1.13 > 0.60 | ✖ |

(c) Reconstructed forest

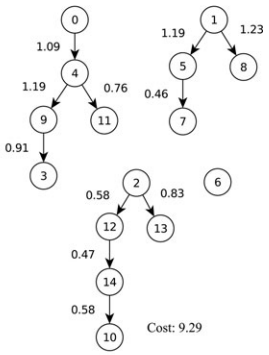
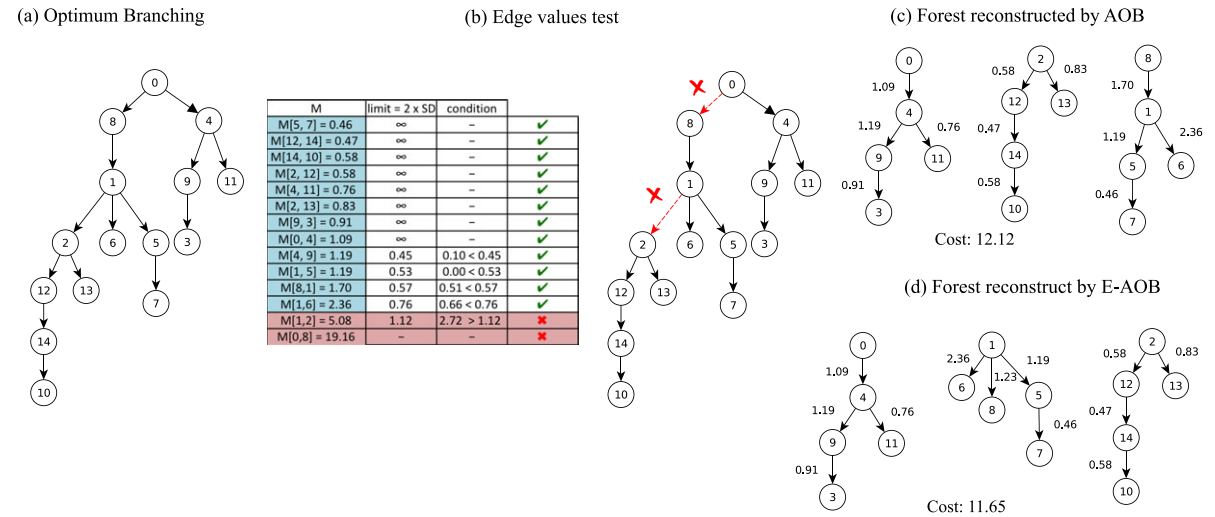



Fig. 8. Simulation of AOK algorithm. (a) Dissimilarity matrix relating 15 × 15 semantically similar images. (b) Evaluation of the edges that can be added to the forest. (c) Reconstructed forest, after eliminating the edges whose values are higher than the calculated limit.

In Fig. 8, we illustrate this algorithm with an example considering $n = 15$ semantically similar images. The algorithm initially receives the dissimilarity matrix $M$ in Fig. 8(a) that contains the dissimilarities between each pair of images. The algorithm starts with $n = 15$ trees in the forest, gradually processing edges that are candidates to include in the forest. In this example, the algorithm starts with the position $M[5, 7]$ that has the lowest entry in the dissimilarity matrix $M$, connecting nodes 5 and 7 in the forest. Next, the algorithm test positions $M[12, 14]$, $M[14, 10]$, $M[2, 12]$, and $M[4, 11]$, which are also included in the forest, since each of them connects disjoint trees. However, the next position $M[7, 5]$ is discarded, since 7 is not the root of a tree. Subsequently, the algorithm tests the next eligible edges, until we have enough number of edges to calculate the current standard deviation of the selected edges. Thus, when edge (4,9) is evaluated, the limit of accepting a new edge is also calculated, as $limit = k \times \sigma = 2 \times 0.225 = 0.45$. Since $M[4, 9] - M[0, 4] = 0.10$ is lower than the $limit = 0.45$, the edge $M[4, 9]$ is also included in the forest. The algorithm proceeds by checking each edge in order until it evaluates edge $M[1, 6]$. This edge passes the first two tests, but its value is above the allowed limit calculated so far for all the previously selected edges in the forest ($M[1, 6] - M[1, 8] = 1.13 > limit = 0.60$). This edge is then discarded and the algorithm stops, reconstructing the forest shown in Fig. 8(c).

### 4.5. *Automatic OB (AOB) and extended AOB (E-AOB)*

Similar to the idea used for the development of AOK, Costa et al. (2014) developed an extended version of OB algorithm to deal with IPFs. In this case, given the dissimilarity matrix $M$ relating $n$ semantically similar images, first the OB algorithm is executed. Then, a threshold using the standard deviation of edges already added to the forest is also used for limiting the number of edges to be included in the final forest ($limit = k \times \sigma, k = 2$, also determined after some experiments). This first approach was named AOB.




| M | limit = 2 x SD | condition | |
|---|---|---|---|
| M[5, 7] = 0.46 | ∞ | – | ✔ |
| M[12, 14] = 0.47 | ∞ | – | ✔ |
| M[14, 10] = 0.58 | ∞ | – | ✔ |
| M[2, 12] = 0.58 | ∞ | – | ✔ |
| M[4, 11] = 0.76 | ∞ | – | ✔ |
| M[2, 13] = 0.83 | ∞ | – | ✔ |
| M[9, 3] = 0.91 | ∞ | – | ✔ |
| M[0, 4] = 1.09 | ∞ | – | ✔ |
| M[4, 9] = 1.19 | 0.45 | 0.10 < 0.45 | ✔ |
| M[1, 5] = 1.19 | 0.53 | 0.00 < 0.53 | ✔ |
| M[8,1] = 1.70 | 0.57 | 0.51 < 0.57 | ✔ |
| M[1,6] = 2.36 | 0.76 | 0.66 < 0.76 | ✔ |
| M[1,2] = 5.08 | 1.12 | 2.72 > 1.12 | ✖ |
| M[0,8] = 19.16 | – | – | ✖ |

Fig. 9. Simulation of AOB and E-AOB algorithm.

However, the direct extension of OB algorithm for IPFs did not improve the results obtained by AOK. This happened because the AOB algorithm considers all edges to construct the minimum branching. Once some edges are removed to build the forest, several partitions that are independent of each other are created. To improve the performance of AOB, Costa et al. (2014) included an additional step that analyzes each of these partitions separately, choosing edge connections that are optimal considering only the edges that belong to the current partition. With this reexecution, it was possible to improve the reconstruction results, as each tree could be analyzed independent of each other. This strategy characterizes the extended automatic OB (E-AOB) algorithm.

Figure 9 illustrates the execution of AOB and E-AOB algorithms for the same dissimilarity matrix $M$ shown in Fig. 8(a), relating $n = 15$ semantically similar images. After executing the OB algorithm once, we obtained the minimum branching shown in Fig. 9(a). We sort its edges from the lowest to the highest weight to decide which edges should be deleted to construct the forest. The table in Fig. 9(b) shows the dynamic calculation of the standard deviation, which starts with edge $M[4, 9]$. Since $M[4, 9] - M[0, 4] = 0.10 < limit = 0.45$, the edge (4,9) is also included in the forest. The algorithm continues to the next edge, $M[1, 5]$, which can also be included in the forest, as $M[1, 5] - M[4, 9] = 0.00 < limit = 0.53$. However, when it reaches the edge $M[1, 2]$, $M[1, 2] - M[1, 6] = 2.72 > limit = 1.12$, which is above the allowed limit. Hence, this edge is discarded and the algorithm stops, returning the forest depicted in Fig. 9(c), which is the result of applying only AOB approach.

Based on the forest reconstructed in the first part, the OB algorithm is executed again for each set represented by the nodes of the trees constructed by AOB, which can fix the direction of some parent–child relationships. For instance, with the execution of this additional step in AOB, it was possible to correct the position of nodes 1 and 8, whose parent–child relationship was reversed in the initial IPF reconstructed by AOB, as shown in Fig. 9(d). Since the weight of edge $M[8, 1]$ is much higher than the weight of edge $M[1, 8]$, it means that image 8 underwent more transformations, and therefore, it is probably a descendant, not an ancestor, of image 1.

### 4.6. Additional remarks

In addition to the approaches explained in the previous section, other attempts to improve the result of the IPF reconstruction were also implemented. Costa et al. (2014) proposed a fusion of approaches, in which the result of each reconstruction algorithm (AOK, AOB, and E-AOB) were combined, aiming at exploring the different properties of each method and their complementarity, in such a way that errors introduced by one method could be fixed by other method(s). Results for the fusion among AOK $\times$ AOB $\times$ E-AOB showed better or equivalent performance to the algorithm with the state-of-art approach up to that point (E-AOB). Although being a competitive solution, depending on the target case, the E-AOB solution can be more efficient as there is no need to generate any perturbation on the dissimilarity matrix or combine different phylogeny approaches.

Oikawa et al. (2016) proposed a strategy to improve the IPF reconstruction using manifold learning and spectral clustering. This approach aimed at correctly separating semantically similar images in their respective groups, prior to the tree reconstruction step. A modified version of Isomap algorithm was used as a preprocessing step, creating an intermediary graph representation of the input data. Next, this new representation was used with spectral clustering, grouping the images on their corresponding phylogeny trees. To reconstruct the phylogeny trees, E-AOB method was used. With this approach, a significant improvement was obtained for finding the roots of the trees, which are very important for forensic purposes, as it allows an investigator to get closer to the source of the content distribution, and the identification of the potential culprits behind it.

Although initially developed for images, the reconstruction approaches explained in the previous sections can also be applied for other media types. For instance, in video phylogeny, for a chosen number $f$ of frames taken from $n$ near-duplicate videos that are temporally coherent, Dias et al. (2011) proposes the construction of $f$ dissimilarity matrices, each of them giving the dissimilarity among the videos evaluated on that frame. For reconstructing the video phylogeny tree (VPT), several attempts were made, including creating a dissimilarity matrix $M'$ where each entry $m_{ij}$ could be the minimum or the average value across the corresponding entries in all $f$ matrices. Other tests included normalizing all the matrices to the interval [0, 1] before calculating the final VPT. Once $M'$ is constructed, the final VPT is built using the OK algorithm. Attempts were also made by dealing with the $f$ frame phylogeny trees directly instead of performing operations on the dissimilarity matrices. In this case, $f$ trees are built, one for each set of frames, and used in a "tree reconciliation" algorithm that adds, at each step, the most used edges across the individual trees, generating the final VPT.

In audio phylogeny, OK is also used for reconstructing the phylogeny trees, similar to the procedure used in the case of near-duplicate images.

## 5. Evaluation metrics

To evaluate the reconstruction methods quantitatively, Dias et al. (2010, 2012) developed four metrics, considering the ground truth (GT) is available. Thus, for IPTs, given the GT tree $\text{IPT}_\text{G}$ and the reconstructed tree $\text{IPT}_\text{R}$, these metrics are defined as follows.

1. **Roots**: It evaluates if the root of the reconstructed tree is the same as the GT tree.

$$\mathrm{R}(\mathrm{IPT_G}, \mathrm{IPT_R}) = \begin{cases} 1, \text{if } \textbf{Root}\,(\mathrm{IPT_G}) = \textbf{Root}\,(\mathrm{IPT_R}) \\ 0, \text{otherwise} \end{cases}. \tag{6}$$

2. **Edges**: It evaluates the edge connections. Therefore, if an edge connecting two nodes in $\mathrm{IPT_R}$ is the same as in $\mathrm{IPT_G}$, then it is considered correct.

$$\mathrm{E}(\mathrm{IPT_G}, \mathrm{IPT_R}) = \frac{|E_1 \cap E_2|}{n-1}. \tag{7}$$

3. **Leaves**: It compares if the documents without descendants (leaves of the tree) of $\mathrm{IPT_R}$ are the same as found in $\mathrm{IPT_G}$.

$$\mathrm{L}(\mathrm{IPT_G}, \mathrm{IPT_R}) = \frac{|L_1 \cap L_2|}{L_1 \cup L_2}. \tag{8}$$

4. **Ancestry**: For each node, it evaluates whether all ancestors (node parent, grandparent, great grandparent, and so on) until the root in $\mathrm{IPT_G}$ are correctly found in $\mathrm{IPT_R}$.

$$\mathrm{A}(\mathrm{IPT_G}, \mathrm{IPT_R}) = \frac{|A_1 \cap A_2|}{A_1 \cup A_2}. \tag{9}$$

For evaluating IPFs, the same quantitative metrics are also considered, but using a more general equation:

$$\mathit{metric}(\mathrm{IPF_G}, \mathrm{IPF_R}) = \frac{\mathrm{S_G} \cap \mathrm{S_R}}{\mathrm{S_G} \cup \mathrm{S_R}}, \tag{10}$$

where *metric* represents the metric being evaluated (root, edges, leaves, or ancestry), $\mathrm{IPF_R}$ is the reconstructed forest with elements represented by $S_\mathrm{R}$, and $\mathrm{IPF_G}$ is the forest GT with elements $S_\mathrm{G}$. This equation calculates the intersection of the result returned by $\mathrm{IPF_R}$, regarding the evaluation metric, with respect to the reference forest $\mathrm{IPF_G}$, and normalizes it by the union of both sets.

In addition, the metric *depth* is used for assessing the average depth in which the correct roots of the IPF are identified. This metric measures the average distance, in number of edges, between each of the original roots in $\mathrm{IPF_G}$, and the reconstructed roots in $\mathrm{IPF_R}$, and *vice versa*. This calculation is done in both ways to also take into account the cases when a nonroot is misjudged as a root in $\mathrm{IPF_R}$. The lower the average depth, the better. When the algorithm finds the roots of the trees at depth zero, it means all of them were correctly identified. This metric is also useful for measuring false-positive cases, that is, nodes wrongly placed as roots in the reconstructed forest (Oikawa et al., 2016).

As an example, consider the GT tree and the trees reconstructed by OK, BP, and OB shown in Fig. 10. Table 1 shows the result of all metrics for each of the reconstructed trees. Since all approaches correctly recovered the root of the tree (node 0), this metric returns 1 for all of them. Since OK and BP methods are greedy algorithms, they make a local optimal choice (the lowest cost edge, and the nearest vertex to a vertex already on the graph, respectively) at each stage to find the

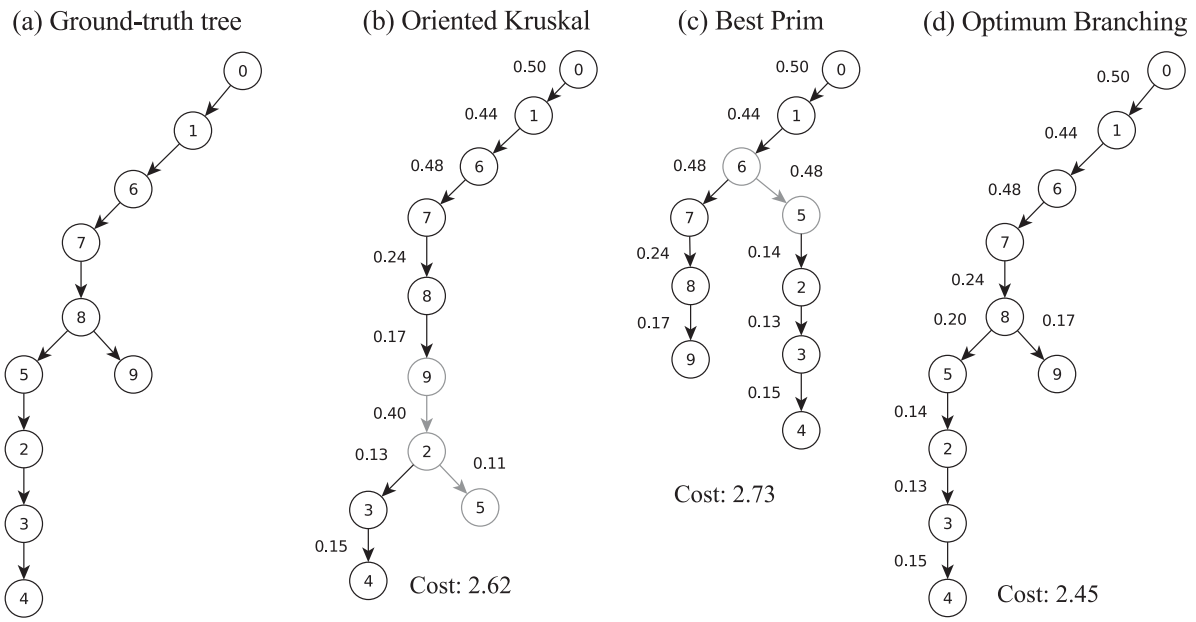



Fig. 10. Comparison among different phylogeny tree reconstruction methods with respect to the (a) ground-truth tree, (b) oriented Kruskal, (c) best Prim, and (d) optimum branching. The edges that were placed in the incorrect position after the reconstruction are highlighted in grey.

Table 1
Calculation of the evaluation metrics for the reconstructed trees depicted in Fig. 10. Values in bold indicate the edges placed in the wrong position

| | |
|---|---|
| **Roots** | |
| | $\text{Root(GT)} = \text{Root(OK)} = \text{Root (BP)} = \text{Root (OB)} = \{0\}$ |
| **Edges** | |
| | $E_{GT}$={(0 → 1), (1 → 6), (6 → 7), (7 → 8), (8 → 5), (8 → 9), (5 → 2), (2 → 3), (3 → 4)} |
| | $E_{OK}$={(0 → 1), (1 → 6), (6 → 7), (7 → 8), (8 → 9), **(9 → 2)**, **(2 → 5)**, (2 → 3), (3 → 4)} |
| | $E_{BP}$={(0 → 1), (1 → 6), (6 → 7), **(6 → 5)**, (7 → 8), (8 → 9), (5 → 2), (2 → 3), (3 → 4)} |
| | $E_{OB}$={(0 → 1), (1 → 6), (6 → 7), (7 → 8), (8 → 5), (8 → 9), (5 → 2), (2 → 3), (3 → 4)} |
| | $\text{E(GT, OK)} = \frac{7}{9} = 0.778$ |
| | $\text{E(GT, BP)} = \frac{8}{9} = 0.889$ |
| | $\text{E(GT, OB)} = 1.0$ |
| **Leaves** | |
| | $L_{GT} = L_{BP} = L_{OB} = \{9, 4\}$ |
| | $L_{OK} = \{5, 4\}$ |
| | $\text{L(GT, OK)} = \frac{\lvert\{4\}\rvert}{\lvert\{4,5,9\}\rvert} = \frac{1}{3} = 0.333$ |
| | $\text{L(GT, BP)} = \text{L(GT, OB)} = 1.0$ |
| **Ancestry** | |
| | $\text{A(GT, OK)} = \frac{38}{46} = 0.826$ |
| | $\text{A(GT, BP)} = \frac{33}{41} = 0.805$ |
| | $\text{A(GT, OB)} = 1.0$ |

Table 2
Calculation of the evaluation metrics for the reconstructed forests depicted in Fig. 11. Values in bold indicate the edges placed in the wrong position

| | |
|---|---|
| **Roots** | |
| | $R_{GT} = \{0, 1, 2\}$ |
| | $R_{AOK} = \{0, 1, 2, 6\}$ |
| | $R_{AOB} = \{0, 2, 8\}$ |
| | $R_{E\text{-}AOB} = \{0, 1, 2\}$ |
| | $R(GT, AOK) = \frac{3}{4} = 0.75$ |
| | $R(GT, AOB) = \frac{2}{4} = 0.50$ |
| | $R(GT, E\text{-}AOB) = \frac{3}{3} = 1.00$ |
| **Edges** | |
| | $E_{GT}$={(0 → 4), (4 → 9), (9 → 3), (4 → 11), (1 → 5), (1 → 8), (5 → 7), (8 → 6), (2 → 12), (2 → 13), (12 → 14), (14 → 10)} |
| | $E_{AOK}$={(0 → 4), (4 → 9), (9 → 3), (4 → 11), (1 → 5), (1 → 8), (5 → 7), (2 → 12), (2 → 13), (12 → 14), (14 → 10)} |
| | $E_{AOB}$={(0 → 4), (4 → 9), (9 → 3), (4 → 11), (2 → 12), (2 → 13), (12 → 14), (14 → 10), **(8 → 1)**, (1 → 5), **(1 → 6)**, (5 → 7)} |
| | $E_{E\text{-}AOB}$ = {(0 → 4), (4 → 9), (9 → 3), (4 → 11), **(1 → 6)**, (1 → 8), (5 → 7), (2 → 12), (2 → 13), (12 → 14), (14 → 10)} |
| | $E(GT, AOK) = \frac{11}{12} = 0.917$ |
| | $E(GT, AOB) = \frac{10}{12} = 0.833$ |
| | $E(GT, E\text{-}AOB) = \frac{11}{12} = 0.917$ |
| **Leaves** | |
| | $L_{GT} = L_{AOB} = \{3, 6, 7, 10, 11, 13\}$ |
| | $L_{AOK} = \{3, 7, 8, 10, 11, 13\}$ |
| | $L_{E\text{-}AOB} = \{3, 6, 7, 8, 10, 11, 13\}$ |
| | $L(GT, AOK) = \frac{5}{7} = 0.714$ |
| | $L(GT, AOB) = \frac{6}{6} = 1.0$ |
| | $L(GT, E\text{-}AOB) = \frac{6}{7} = 0.857$ |
| **Ancestry** | |
| | $A(GT, AOK) = \frac{19}{21} = 0.905$ |
| | $A(GT, AOB) = \frac{20}{24} = 0.833$ |
| | $A(GT, E\text{-}AOB) = \frac{20}{21} = 0.952$ |
| **Depth** | |
| | $D(GT, AOK) = \frac{(\frac{2}{3}+\frac{0}{4})}{2} = \frac{0.666+0.0}{2} = 0.333$ |
| | $D(GT, AOB) = \frac{(\frac{1}{3}+\frac{1}{3})}{2} = \frac{0.333+0.333}{2} = 0.333$ |
| | $D(GT, E\text{-}AOB) = \frac{(\frac{0}{3}+\frac{0}{3})}{2} = \frac{0.0+0.0}{2} = 0.0$ |

minimum branching. Therefore, the evaluation for the other metrics (edges, leaves, and ancestry) is, in general, lower than the results obtained by OB. Since OB is an exact algorithm, it always finds one optimum global solution.

A similar evaluation is performed for IPFs, as shown in Fig. 11 and Table 2. The comparison is made with the GT forest and AOK, AOB, and E-AOB, with the latter presenting the best results in the literature. In this example, although AOK reconstructed the forest with the lowest cost, it wrongly identified node 6 as root, decreasing the accuracy of AOK in comparison with E-AOB in all evaluated metrics.

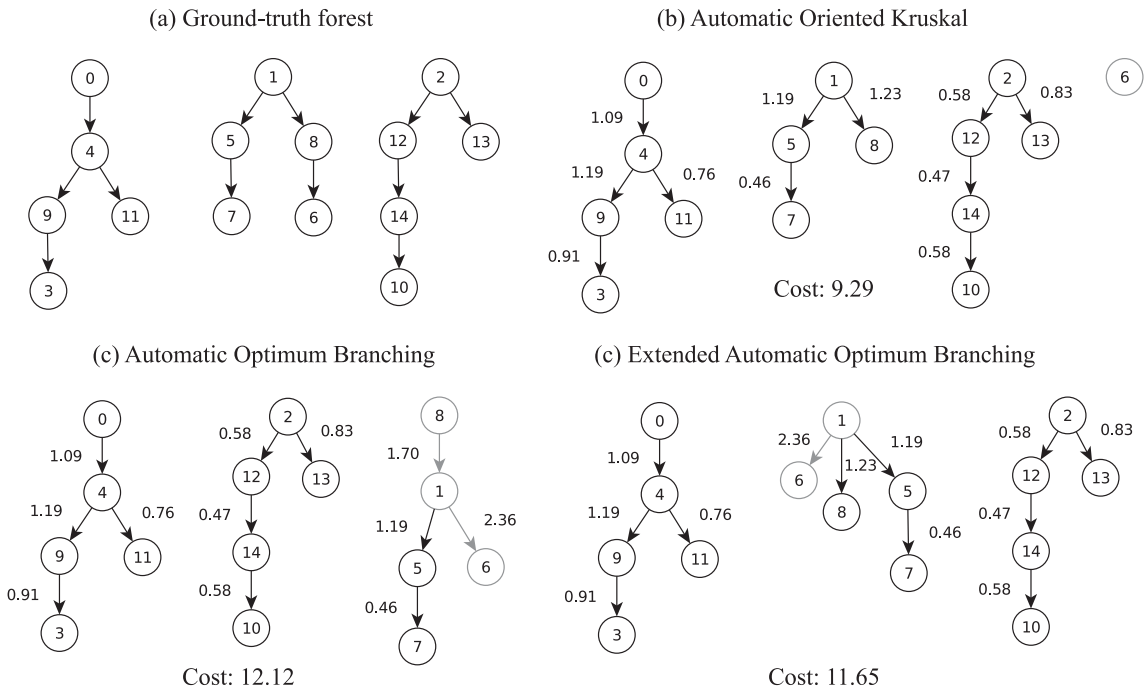



Fig. 11. Comparison among different IPF reconstruction methods. The nodes and edges that are in the wrong position in comparison to the GT forest are highlighted in grey.

## 6. Application in real-case scenarios

To show what can be obtained with approaches in multimedia phylogeny, we consider the case of the image captured by the White House photographer, Pete Souza, on 1 May 2011. This episode, also known as *The Situation Room*, portrays the US President Barack Obama along with his national security team, receiving live updates from Operation Neptune Spear, which resulted in Osama bin Laden's death. After it was first published on the Internet, several edited versions of this picture appeared online on different channels, with cases of text overlay, face swap, insertion of other people, objects, and animals in the picture, among others.

For this experiment, 98 near-duplicate images were collected through Google Images and manually classified in different patterns considering cases of inserting the Italian soccer player Mario Balotelli (ID a*), text overlay (ID b*), watermarking (ID c*), face swapping (ID d*), insertion of a joystick (ID e*) and hats (ID g*), and changes in the image size without splicing (ID n*). Figure 12 illustrates the result of this reconstruction using the E-AOB method. In this case, it is possible to see the root of the tree was correctly identified (image 0000 is the White House version), and related images are, in general, grouped under the same branching, as can be seen by the ID and color of the nodes. Although we do not have the GT of the relationship among these images, it is possible to evaluate the approaches qualitatively, as the image similarities are correctly identified and grouped accordingly.

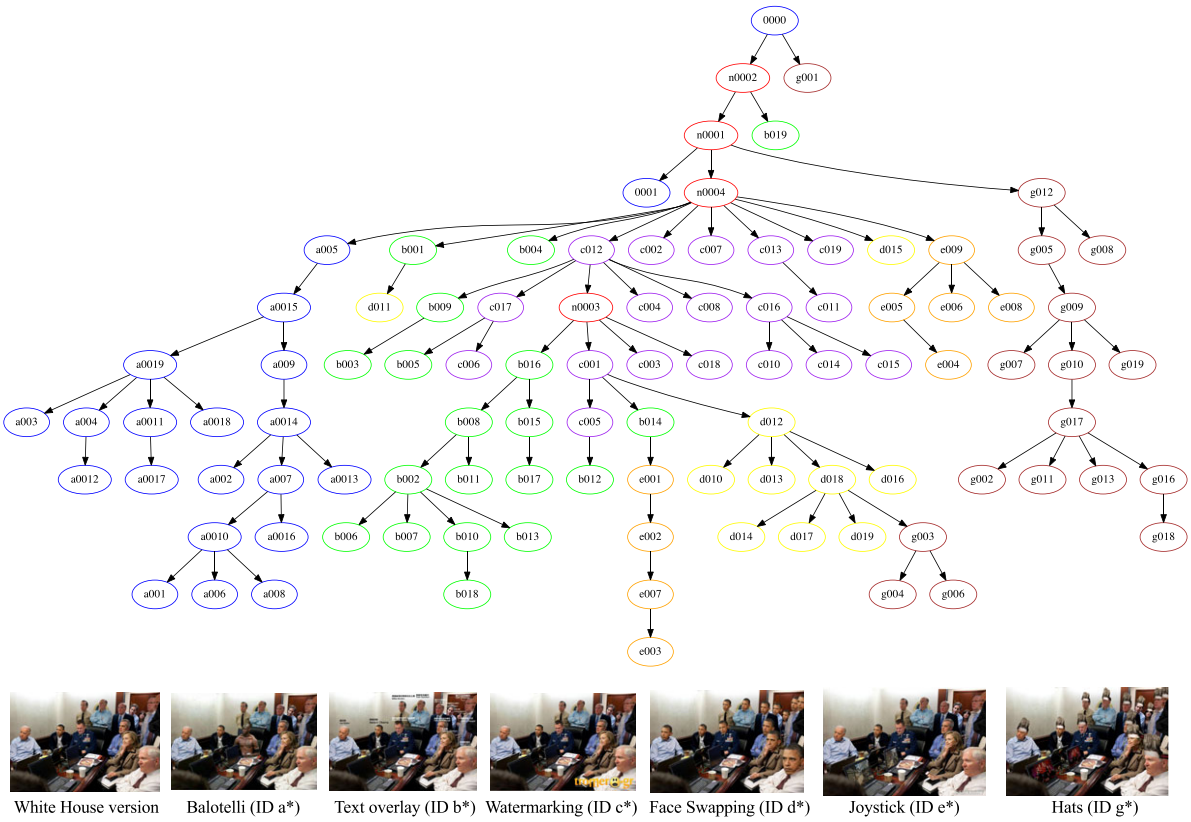



Fig. 12. Reconstruction of the image phylogeny tree of images collected from the Internet portraying *The Situation Room* episode. This IPT was reconstructed using E-AOB method.

### *6.1. Datasets and source code*

All test cases and the source code of the methods developed in multimedia phylogeny are freely available. The datasets are registered at http://dx.doi.org/10.6084/m9.figshare.1012816, and can also be downloaded at http://www.recod.ic.unicamp.br/~oikawa/datasets.html. The source code is available in a public repository at http://repo.recod.ic.unicamp.br/public/projects.

## 7. Open issues

In multimedia phylogeny, most of the work has been mainly developed for images, but other types of media are equally important and also create different phylogenetic structures. Especially, some important issues to be solved are related to video phylogeny. Not rarely, videos are created from the combinations of several shots from other videos, which makes the implementation of multiple parenting phylogeny for videos an important problem to be explored. In addition, for a better performance, instead of evaluating the distances using the entire sequence of a set of video files,

it would be interesting to develop methods for automatically segmenting a video into shots, and evaluate their dissimilarity using these shots.

Another challenge when working with videos is regarding their encoding format. While images, in general, are encoded in one format (JPEG) with only a scalar quality parameter, videos can be encoded in several different formats, each one characterized by a set of parameters. Thus, if the two videos being analyzed are encoded using different methods, it becomes harder to use these evaluated distance for phylogeny purposes. It is also important to consider how videos with different frame rates and temporal clipping, for instance, should be evaluated. Part of these open questions are also valid for audio, which have been tested only in a controlled environment (e.g., Nucci et al., 2013). More experiments are necessary using real-world audio sets, also considering other types of processing operators, such as time or pitch scaling.

Text phylogeny is a special case among the approaches proposed in the literature thus far. The distance among text documents forms a metric space and, as a consequence, brings more challenges on the tree reconstruction, as it becomes harder to determine the parent–child relationship using symmetric dissimilarity values. It also has some particularities, since a document can undergo several transformations, and reverse to its original version without leaving hints, different from the image domain in which reversibility is much harder. Ryu et al.'s (2008) approach is one of the few works that goes beyond the plagiarism detection tools. It calculates the similarity values of two documents and direction of copying, using a directed graph representation to describe the plagiarized pair and its direction. However, the indication of the direction of the documents' modifications is only possible because time stamps are used combined to the document contents similarity. Since this information is neither always reliable nor available, a different approach should be developed to deal with this type of media.

## 8. Conclusions

Several problems in machine learning, computer vision, and information retrieval calculate pairwise (dis)similarities to obtain a measure of closeness among a set of objects. In general, it is desirable that these (dis)similarities satisfy the properties of a metric, although it is not always possible to enforce this property in some applications.

In this paper, we reviewed various dissimilarity functions and reconstruction strategies developed for different types of media in multimedia phylogeny. Each media has its own challenges, and having a good dissimilarity function, which correctly measures their relationship, is of fundamental importance for the reconstruction of phylogeny trees. Although in general, the presented dissimilarity functions are not distances in a metric space, the asymmetry on this calculation is what allows us to determine the parenthood relationships. It might be interesting, however, to explore the behavior of the current approaches if we use a "relaxed triangle inequality" (Fagin and Stockmeyer, 1998), for instance, which could be used for simplifying the dissimilarity calculation, or solve some ambiguity in parent–child relationships.

Albeit current approaches in multimedia phylogeny have presented satisfactory results, both quantitatively and qualitatively, this area is recent and enticing with room for new and elegant math solutions. Future developments in this area include not only expand the current approaches for other media types, but also improvements in the dissimilarity calculation part between pairs of

objects, as well as in the phylogeny reconstruction part itself. The dissimilarity calculation currently represents the most expensive step in the multimedia phylogeny framework, which is not reasonable if we consider that during analysis of real forensic cases, hundreds of images or videos, with various resolutions, need to be processed. The first solution to deal with large-scale sets of images has been proposed by Dias et al. (2013c), in which the original reconstruction algorithm was updated to allow missing values in the dissimilarity matrix, significantly reducing the calculation time. Implementation and evaluation of other approaches, changing the calculation strategy or including the use of graphics hardware, is necessary.

Some additional challenges in the dissimilarity calculation involve designing new and faster ways of putting the two compared objects in a common reference domain for comparison (registration step when dealing with visual content). Would it be possible to come up with a dissimilarity function in which registration would be optional? In addition, when comparing two objects in the same reference domain, will we focus on point-wise differences or come up with a measure taking into consideration the distributions of some well-designed features related to these objects? Regarding the reconstruction itself, when dealing with the graph and arc weights for each comparison of two objects, would it be possible to use some metric properties to filter out the least interesting candidates in a phylogeny pool of candidates? How to eliminate outliers eventually found? All these are open questions that certainly deserve some attention from the community in the near future.

## Acknowledgments

This work was partially supported by the São Paulo Research Foundation, FAPESP (grants #2013/08293-7, #2014/19401-8, and #2014/03535-5), National Counsel of Technological and Scientific Development, CNPq (grants #477692/2012-5 and #483370/2013-4), Capes/COFECUB Program (grant #831/15), and Capes DeepEyes project.

## References

Bay, H., Ess, A., Tuytelaars, T., Gool, L., 2008. Speeded-up robust features (SURF). *Computer Vision and Image Understanding* 110, 3, 346–359.

Bock, F., 1971. An algorithm to construct a minimum directed spanning tree in a directed network. In *Developments in Operations Research*, Gordon & Breach, New York, 29–44.

Cheng, X., Chia, L.-T., 2010. Stratification-based keyframe cliques for removal of near-duplicates in video search results. ACM International Conference on Multimedia Information Retrieval, Philadelphia, PA, pp. 313–322.

Chu, Y.J., Liu, T.H., 1965. On the shortest arborescence of a directed graph. *Science Sinica* 14, 1396–1400.

Costa, F.O., Lameri, S., Bestagini, P., Dias, Z., Rocha, A., Tagliasacchi, M., Tubaro, S., 2015. Phylogeny reconstruction for misaligned and compressed video sequences. Proceedings of The International Conference on Image Processing, Quebec.

Costa, F.O., Oikawa, M., Dias, Z., Goldenstein, S., Rocha, A., 2014. Image phylogeny forests reconstruction. *IEEE Transactions on Information Forensics and Security* 9, 10, 1533–1546.

Delfour, M.C., Zolésio, J.P., 2001. *Shapes and Geometries: Analysis, Differential Calculus, and Optimization*. SIAM (Society for Industrial and Applied Mathematics), Philadelphia, PA.

De Rosa, A., Uccheddu, F., Costanzo, A., Piva, A., Barni, M., 2010. Exploring image dependencies: a new challenge in image forensics. Proceedings of SPIE, Vol. 7541 , No. 2, pp. 1–12.

Dias, Z., Goldenstein, S., Rocha, A., 2013a. Toward image phylogeny forests: automatically recovering semantically similar image relationships. *Forensic Science International* 231, 178–189.

Dias, Z., Goldenstein, S., Rocha, A., 2013b. Exploring heuristic and optimum branching algorithms for image phylogeny. *Journal of Visual Communication and Image Representation* 24, 7, 1124–1134.

Dias, Z., Goldenstein, S., Rocha, A., 2013c. Large-scale image phylogeny: tracing back image ancestry relationships. *IEEE Multimedia* 20, 3, 58–70.

Dias, Z., Rocha, A., Goldenstein, S., 2010. First steps towards image phylogeny. IEEE International Workshop on Information Forensics Security, Seattle, WA, pp. 1–6.

Dias, Z., Rocha, A., Goldenstein, S., 2011. Video phylogeny: recovering near-duplicate video relationships. IEEE International Workshop on Information Forensics Security, Iguacu Falls, Brazil, pp. 1–6.

Dias, Z., Rocha, A., Goldenstein, S., 2012. Image phylogeny by minimal spanning trees. *IEEE Transactions on Information Forensics and Security* 7, 2, 774–788.

Edmonds, J., 1967. Optimum branchings. *Journal of Research of National Institute of Standards and Technology* 71B, 48–50.

Fagin, R., Stockmeyer, L., 1998. Relaxing the triangle inequality in pattern matching. *International Journal of Computer Vision* 28, 3, 219–231.

Farid, H., 2006. Exposing digital forgeries in scientific images. 8th Workshop Multimedia and Security, Geneva, pp. 29–36.

Fischler, M., Bolles, R., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications ACM* 24, 6, 381–395.

Fridrich, J., Soukal, D., Lukas, J., 2003. Detection of copy-move forgery in digital images. Digital Forensics Research Conference, Cleveland, OH, pp. 134–137.

Jacobs, D.W., Weinshall, D., Gdalyahu, Y., 2000. Classification with nonmetric distances: image retrieval and class representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 6, 583–600.

Jaimes, A., fu Chang, S., Loui, A., 2002. Duplicate detection in consumer photography and news video. ACM Workshop on Multimedia and Security, Juan Les Pins, France, pp. 423–424.

Jain, A., 2010. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters* 31, 8, 651–666.

Joly, A., Buisson, O., Frélicot, C., 2007. Content-based copy retrieval using distortion-based probabilistic similarity search. *IEEE Transactions on Multimedia* 9, 2, 293–306.

Ke, Y., Sukthankar, R., Huston, L., 2004. Efficient near-duplicate detection and subimage retrieval. ACM Workshop on Multimedia and Security, New York, pp. 869–876.

Kriegel, H.-P., Kröger, P., Zimek, A., 2009. Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data* 3, 1, 1–58.

Kruskal, J., 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society* 7, 1, 48–50.

Lameri, S., Bestagini, P., Melloni, A., Milani, S., Rocha, A., Tagliasacchi, M., Tubaro, S., 2014. Who is my parent? Reconstructing video sequences from partially overlapping matching shots. Proceedings of The International Conference on Image Processing, Paris, pp. 5342–5346.

Lewin, B., 1997. *Genes VI*. Oxford University Press, New York.

Ling, Z.X.H., Zou, F., Lu, Z., Li, P., 2010. Robust image copy detection using multi-resolution histogram. ACM International Conference on Multimedia Information Retrieval, Philadelphia, PA, pp. 129–136.

Maret, Y., 2007. Efficient duplicate detection based on image analysis. PhD thesis, École Polytechinique Fédérale de Lausanne, Switzerland.

Melloni, A., Bestagini, P., Milani, S., Tagliasacchi, M., Rocha, A., Tubaro, S., 2014. Image phylogeny through dissimilarity metrics fusion. Proceedings of European Workshop on Visual Information Processing, Paris, pp. 1–6.

Mihcak, M., Kozintsev, I., Ramchandran, K., Moulin, P., 1999. Low-complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Signal Processing Letters* 6, 12, 300–303.

Nucci, M., Tagliasacchi, M., Tubaro, S., 2013. A phylogenetic analysis of near-duplicate audio tracks. IEEE International Workshop on Multimedia Signal Processing, Pula, Italy, pp. 99–104.

Oikawa, M.A., Dias, Z., Rocha, A., Goldenstein, S., 2016. Manifold learning and spectral clustering for image phylogeny forests. *IEEE Transactions on Information Forensics and Security* 11, 1, 5–18.

Oliveira, A., Ferrara, P., DeRosa, A., Piva, A., Barni, M., Goldenstein, S., Dias, Z., Rocha, A., 2014. Multiple parenting identification in image phylogeny. Proceedings of The International Conference on Image Processing, Paris, pp. 5347–5351.

Oliveira, A., Ferrara, P., DeRosa, A., Piva, A., Barni, M., Goldenstein, S., Dias, Z., Rocha, A., 2015. Multiple parenting phylogeny relationships in digital images. *IEEE Transactions on Information Forensics and Security*, DOI: 10.1109/TIFS.2015.2493989.

O'Searcoid, M., 2006. *Metric Spaces*. Springer, New York.

Pluim, J.P.W., Maintz, J.B.A., Viergever, M.A., 2003. Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging* 22, 8, 986–1004.

Prim, R.C., 1957. Shortest connection networks and some generalizations. *Bell System Technical Journal* 36, 6, 1389–1401.

Reinhard, E., Ashikhmin, M., Gooch, B., Shirley, P., 2001. Color transfer between images. *IEEE Computer Graphics Applications* 21, 34–41.

Rocha, A., Scheirer, W., Boult, T.E., Goldenstein, S., 2011. Vision of the unseen: current trends and challenges in digital image and video forensic. *ACM Computing Surveys (CSUR)* 42, 26, 26:1–26:42.

Russakoff, D.B., Tomasi, C., Rohlfing, T., Maurer Jr., C.R., 2004. Image similarity using mutual information of regions. *Proceedings of the 8th European Conference on Computer Vision*, Prague, pp. 596–607.

Ryu, C.-K., Kim, H.-J., Ji, S.-H., Woo, G., Cho, H.-G., 2008. Detecting and tracing plagiarized documents by reconstruction plagiarism-evolution tree. *IEEE International Conference on Computer and Information Technology*, Sydney, pp. 119–124.

Schaffalitzky, F., Zisserman, A., 2002. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?" *European Conference on Computer Vision*, Copenhagen, pp. 414–431.

Scheirer, W.J., Wilber, M.J., Eckmann, M., Boult, T.E., 2014. Good recognition is non-metric. *Pattern Recognition* 47, 8, 2721–2731.

Valle, E., 2008. Local descriptor matching for image identification systems. PhD thesis, Universite de Cergy-Pontoise, France.

Xu, R., Wunsch II, D., 2005. Survey of clustering algorithms. *IEEE Transactions on Neural Networks* 16, 3, 645–678.

Zhang, D.-Q., Chang, S.F., 2004. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. ACM Workshop on Multimedia and Security, New York, pp. 877–884.

Zitova, B., Flusser, J., 2003. Image registration methods: a survey. *Image and Vision Computing* 21, 11, 977–1000.