Manifold Learning and Spectral Clustering for Image Phylogeny Forests

Marina A. Oikawa, Zanoni Dias, Anderson de Rezende Rocha, *Senior Member, IEEE*, and Siome Goldenstein, *Senior Member, IEEE*

***Abstract*— The ever-increasing number of gadgets being used to create digital content, as well as the easiness in sharing, editing, and republishing this content, brings the problem of dealing with a large amount of digital objects (e.g., images or videos) whose content is very similar. Some issues faced by investigators of digital crimes when analyzing this type of data include finding the original source of a suspect image, and the responsible for first publishing it. It is also challenging to determine how these objects are related to each other. Recent efforts in developing algorithms to find automatically the underlying relationship among groups of digital media objects with similar content have been explored in the multimedia phylogeny field. A tree structure is used to represent the relationship among these objects, inspired by the phylogenetic trees in biology. Discovering whether these objects came from the same source or from different sources is fundamentally a clustering problem: 1) related objects belong to the same cluster (tree) and 2) unrelated objects should fit in different clusters. In this paper, we address the problem of finding these clusters in sets of semantically similar images, prior to tree reconstruction. We propose the combination of manifold learning and spectral clustering approaches, which have been successfully used in different applications embedding the original data into a lower, but meaningful, dimensional space. Experiments with more than 40 000 test cases show that the proposed approach improves the accuracy in finding the correct number of trees in the set, as well as the reconstruction of the phylogeny trees.**

***Index Terms*— Image forensics, multimedia phylogeny, manifold learning, phylogeny forests, spectral clustering.**

I. Introduction

DATA clustering is one important strategy to deal with a large amount of unlabeled data. The main goal of clustering is to distinguish to which group (cluster) each data item belongs, usually based on some similarity measure calculated among them. Therefore, objects in the same cluster have higher similarity to each other than to objects assigned to a different one. Due to its importance, methods for finding good clusters have been the focus of investigation from several researchers [1]–[5].

In this work, we take advantage of data clustering techniques in the multimedia analysis context, as a first and important step to further improve algorithms in multimedia phylogeny [6]. Inspired on the study of the evolutionary history of an organism or a group of organisms in Biology, multimedia phylogeny aims at finding the evolutionary structure that better describes the history of modifications of a set of digital objects. In this sense, clustering is helpful to group objects coming from the same source (root), while placing unrelated objects in other clusters, as they probably belong to a different phylogeny tree. With improvements in this step, the performance of the algorithms for phylogeny trees reconstruction increases as a consequence. When there are m different trees to be reconstructed, the problem is known as *phylogeny forests reconstruction* [7], [8]. Nevertheless, if there is only one tree in the group, the algorithm should also make this distinction.

Our main goal is the reconstruction of image phylogeny forests, considering images that inherit content from a single parent (i.e., an image resulting of the composition of more than one source is not included). A new approach using *manifold learning* and *spectral clustering* is devised to obtain a better representation of the data points distribution and, hence, produce good image clusters. Both are graph-based approaches that work with the idea of dimensionality reduction, that is, the mapping of data from a high-dimensional space D onto a lower dimension d. In general, through dimensionality reduction methods, it is possible to simplify the original problem, eliminate redundancy, and improve data visualization.

Finding the low dimensional structure embedding the original high-dimensional data is a strategy that has been used in many research domains. In Computer Vision, for instance, it can be used for shape, appearance, and motion parametrization [9]–[11], face recognition [12], [13], or image categorization [14]–[16], with data sets consisting of a large number of samples, each of them composed by several features. More recently, dimensionality reduction has also found applications in Deep Learning, for performing semi-supervised learning [17], and for learning the manifold of 3D brain images [18].

In Figure 1, we present an overview of our proposed approach. Similar to other algorithms in image phylogeny, we begin with a set of n semantically similar images, from

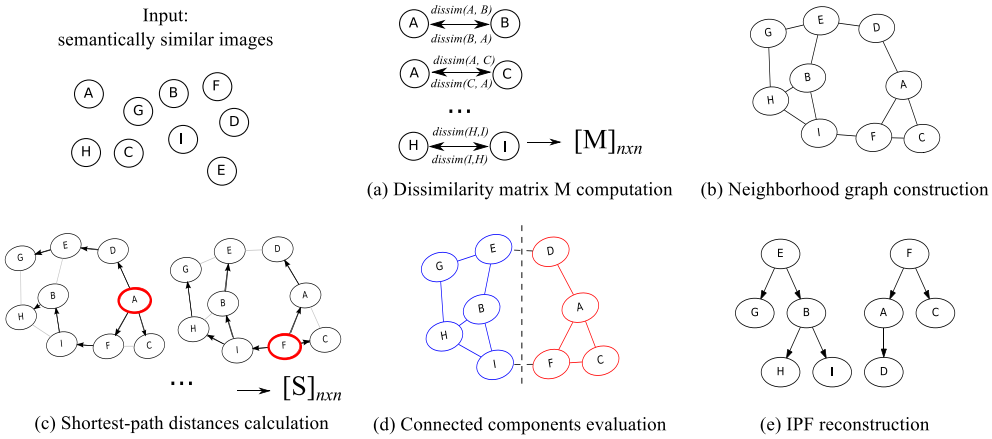
Manuscript received December 25, 2014; revised April 11, 2015; accepted June 3, 2015. Date of publication June 5, 2015; date of current version October 30, 2015. This work was supported in part by the São Paulo Research Foundation under Grant 2014/03535-5 and Grant 2014/19401-8, in part by the Coordination for the Improvement of Higher Education Personnel through the DeepEyes Project, in part by Capes–Cofecub under Grant 831/15, and in part by the National Council for Scientific and Technological Development under Grant 477692/2012-5, Grant 304352/2012-8, Grant 308882/2013-0, Grant 483370/2013-4, Grant 477662/2013-7, and Grant 454082/2014-2. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Teddy Furon.

The authors are with the Institute of Computing, University of Campinas, Campinas 13083-970, Brazil (e-mail: marina.oikawa@ic.unicamp.br; zanoni@ic.unicamp.br; anderson.rocha@ic.unicamp.br; siome@ic.unicamp.br).

This paper has supplementary downloadable material at http://ieeexplore.ieee.org, provided by the authors. The file consists of additional results regarding the methods developed in this paper, along with details about the datasets used in the experiments. The material is 677 KB in size.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2015.2442527



(a) Dissimilarity matrix M computation (b) Neighborhood graph construction

(c) Shortest-path distances calculation (d) Connected components evaluation (e) IPF reconstruction

Fig. 1. Our algorithm starts with a set of n semantically similar images, from which we do not have any knowledge about their phylogenetic relationship. (a) A dissimilarity function $dissim(\mathcal{I}_i, \mathcal{I}_j)$ calculates the amount of residual between the best transformation of $\mathcal{I}_i$ to $\mathcal{I}_j$, for all pairs (i, j) of images, resulting in an $n \times n$ asymmetric matrix M (small values indicate a closer and more plausible relationship). (b) These values are used to construct a neighborhood graph, from which we (c) calculate the shortest-path distances between all pairs of nodes, resulting in a new $n \times n$ matrix of distances S. (d) By applying spectral clustering on S, we calculate the number of connected components k, and cluster the images in a lower dimension $(n \times k)$ using k-means algorithm. Finally, in (e) we reconstruct each IPT using a minimal spanning tree algorithm [6], [19].

which we do not have any knowledge about how they are related to each other. The algorithm starts with (a) the calculation of the dissimilarity between each pair of images, generating an $n \times n$ asymmetric dissimilarity matrix M. Previous works in IPFs [7], [8] go directly to Step (e), sorting the dissimilarity values, and using a threshold to delimit the number of edges used to reconstruct the forest. In our proposed approach, we add new steps to this pipeline, using the dissimilarity values to (b) construct an intermediary graph representation, using the input data points as vertices of the graph, and whose edges are defined according to a proximity criteria (k-NN or ε-neighborhood, for instance). Subsequently, we (c) calculate the shortest-path distance between all pairs of points, resulting in a new matrix of distances. This matrix is used to (d) determine the number of connected components k of the graph, that is, the set of its connected subgraphs, which is equivalent to partition the graph in k clusters. Afterwards, as we have the images correctly grouped, we can finally reconstruct the phylogeny forest using a minimal spanning tree algorithm. The phylogeny trees are graphically represented by directed acyclic graphs (DAGs), as Figure 1 (e) depicts, in which the reconstructed forest is composed by two trees. Nodes E and F are the roots of these trees, and the directed edges show the direction of the parent-child relationship.

II. Definitions and Notations

In this section, we present an overview on multimedia phylogeny, along with a general idea of how manifold learning and spectral clustering algorithms are used herein.

A. Multimedia Phylogeny

Over the past years, it is noteworthy the exponential increase of data being published every day on the Internet. These include not only new content, but also exact copies of digital documents, as well as their modified versions in different levels of editing. Early works exploring a tree structure to represent the relationship among transformed versions of a document include the approach of Joly *et al.* [20] for content-based copy detection of videos, and the work of Kender *et al.* [21], which makes an analogy of how videos from YouTube evolve in the Internet from a genetic viewpoint.

Multimedia phylogeny emerged in this scenario as a new approach to infer ancestor and descendant relationships among objects from the same population, going beyond conventional near-duplicate detection methods [6]. Solutions in this field can be strategically used in several applications, such as tracing the illegal distribution of copyrighted multimedia files or finding the original document among a set of related ones.

Consider, as an example, the case of an image $\mathcal{I}$ being published by one user in a social network. After some time, other users may download this image and create new versions, by changing its color or applying any combination of image transformations, such as cropping, scaling, rotation, etc. All near-duplicates generated from $\mathcal{I}$ may also be downloaded and modified again by other users, in an uncontrolled manner. This continuous process creates several ramifications, but with each step retaining the transformations applied in previous steps, creating a *parent-child* relation among the near duplicates. Multimedia phylogeny analyzes this evolutionary process, looking for causal and ancestry relationships, the type of transformations, and the order in which they were applied [6]. Although being a relatively new research field, several approaches have been developed targeting at different phylogenetic structures and media types, such as Image Phylogeny Trees (IPTs) [6], [22], [23] and Forests (IPFs) [7], [8], multiple parenting relationships [24], large-scale scenarios [25], audio phylogeny [26], and video phylogeny [27], [28].

In image phylogeny, the relationships among a set of related digital objects are represented by DAGs, with weights on each directed edge, calculated from the dissimilarity function

$$dissim(\mathcal{I}_i, \mathcal{I}_j) = \min_{T_{\vec{\beta}}} |\mathcal{I}_j - T_{\vec{\beta}}(\mathcal{I}_i)|_{\text{ point-wise comparison } \mathcal{L}}, \quad (1)$$

for all possible values of the parameter $\overrightarrow{\beta}$ in a family of transformations $\mathcal{T}$ [6]. This function measures the amount of residual between the best transformation from image $\mathcal{I}_i$ to image $\mathcal{I}_j$, according to $\mathcal{T}$, and $\mathcal{I}_j$ itself. Currently, $\mathcal{T}$ includes re-sampling, cropping, affine transformation, brightness and contrast changes, gamma correction, and compression [6]. To estimate $\overrightarrow{\beta}$, feature points are calculated for each pair of images, using SURF [29]. Affine warping and cropping parameters for image $\mathcal{I}_i$ with respect to $\mathcal{I}_j$ are robustly estimated using RANSAC [30], and pixel color normalization parameters are calculated using the color transfer technique of Reinhard *et al.* [31]. Then, one of the images is compressed with the same compression parameters as the other. Finally, both images are uncompressed and the point-wise comparison $\mathcal{L}$ is made by using the standard minimum squared error. More details can be found in previous works [6]–[8].

Since the result of some of the operations appplied on the images are not symmetric (an image that underwent cropping or lossy compression is probably descendant from one that has not, for instance), the result is an asymmetric dissimilarity matrix M, with smaller values for similar images and larger values for more distinct images (with more significant transformations). As a subsequent step, those values are used to reconstruct a phylogenetic tree, using either heuristic-based [6] or optimum branching solutions [19], [23].

When dealing with multiple objects having similar semantic content but that are not directly related to each other (e.g., images taken at different points in time or with different cameras), instead of one tree, several trees (a forest) exist in the set being analyzed. To deal with this type of structure, it is possible to either give to the algorithm the number k of trees to reconstruct, or use a strategy to automatically decide this number. Since it is very unlikely k is known in real-case scenarios, a robust approach should be able to correctly identify if the set is composed by only one or by several trees, without requiring any input from the user.

In the literature, IPF approaches were developed upon the existing IPT framework [6], [23], with strategies devised to automatically find k. They extended the heuristic-based solution Oriented Kruskal to the Automatic Oriented Kruskal (AOK) [7], and the Optimum Branching approach to the Automatic Optimum Branching (AOB) and Extended Automatic Optimum Branching (E-AOB) [8]. These approaches introduce a threshold γ to control the number of edges to be included in the forest, whose calculation is based on the variance of processed edges. Results were promising, specially when considering a fusion of the aforementioned approaches [8].

In this context, it is worth mentioning that there are two tasks of utmost importance when dealing with IPFs: (i) finding the correct number of trees and (ii) grouping elements belonging to the same tree on the same cluster. Both tasks are interconnected and directly affect the IPF reconstruction.

Through controlled experiments, Dias *et al.* [7] showed that when the Oriented Kruskal algorithm is fed with the correct number of trees k, it can successfully find the roots of the trees in 90.0% of the cases when the semantically similar images come from the same camera, and in 91.7% for images from multiple cameras (for forests up to five trees). Without knowing k, the performance decreases with the number of trees. On the other hand, experiments performed by Costa *et al.* [8] showed the importance of having the correct separation of the image groups: with AOB algorithm, the optimum branching B is first split into k groups, each group corresponding to one tree in the forest. Although it was possible to correctly estimate the number of trees, the topology of each individual tree was not accurate in some cases, since it had been done considering all edges of B. Nonetheless, since the images were already separated in groups, an iterative execution of the optimum branching algorithm in each group separately (E-AOB) was enough to solve this problem.

In this paper, we focus on the development of a new approach for IPFs toward obtaining the correct number of trees k prior to the reconstruction step. Earlier, there was not an explicit image clustering phase; the nodes of the trees were simply connected according to a threshold γ. Once this threshold was reached, the topology of the trees were already set, with exception of E-AOB, which had an additional step for refining the trees. AOB played the role of grouping the images to be used as input to E-AOB algorithm which, in turn, improved the results by analyzing each group individually to reconstruct its corresponding tree. That was our first hint toward the design of a new approach that could better capture the nature of the relationship among the images.

B. Manifold Learning

Manifold learning is a mathematical tool for Nonlinear Dimensionality Reduction (NLDR), whose main goal is the projection of one topological space onto another, in which the data distribution is more regular than originally [32]. More formally, given a data set X with n samples $x_i, ..., x_n \in \mathbb{R}^D$ embedded on a manifold $\mathcal{M}$, manifold learning consists in finding the corresponding coordinates $y_i, ..., y_n \in \mathbb{R}^d$ ($d < D$), through a mapping $f : x \mapsto y$ that preserves certain properties of $\mathcal{M}$. This strategy has been successfully applied to a number of applications in face recognition [33], [34], hyperspectral image classification [35]–[37], speech processing [38], [39], and motion tracking [40].

Approaches for learning nonlinear manifolds can be either global or local, depending on which geometric characteristics are preserved. Some examples are Isomap [41], LLE (Local Linear Embedding) [42], Laplacian Eigenmaps [43], Diffusion Maps [44], Hessian Eigenmaps [45], and their variations [46]–[50]. It is possible to summarize these algorithms in three steps:

1. Search the nearest neighbors of each data point, and construct a weighted graph representing this neighborhood;
2. Transform this graph into another representation, keeping local or global properties in the manifold;
3. Compute the low dimensional graph-embedding.

Step (1) finds which points are neighbors on $\mathcal{M}$, according to some criteria (usually k-NN or ε-neighborhood). The result is a weighted graph $G = (V, E)$, with the vertices V being the

input data points X, and E indicating the set of edges $e(i, j)$ connecting neighbor points (x_i, x_j). The weight $w(x_i, x_j)$ of each edge can be the distance between each pair of points, as measured by the Euclidean distance, for instance.

In Step (2), the main goal is to transform this data into a suitable input for the next step, in a global (preserving the geometry of the data at all scales), or in a local manner (small neighborhood), in the form of a square symmetric matrix, also called *feature matrix* [51]. For instance, Isomap computes a global optimal approach by constructing an embedding using the length of the shortest path between *all* pairs of vertices in G, returning an $n \times n$ symmetric matrix of graph distances.

Lastly, in Step (3), a low dimensional graph-embedding is computed, using the set of eigenvectors associated with the top or bottom few eigenvalues of the feature matrix.

Among the aforementioned approaches, Isomap is the simplest NLDR method. Once the nearest-neighbor graph is constructed, Isomap estimates the *geodesic distance* between all pairs of points, that is, the length of the shortest curve on the manifold connecting two points [48]. These distances are stored in a new $n \times n$ matrix S, in which the final step of graph embedding is performed by applying multidimensional scaling (MDS) [52]. We take advantage of the simplicity and intuitiveness of Isomap, and use its main idea as a foundational step for our approach, keeping its first and second step: the neighborhood graph construction and the calculation of the geodesic distances. In addition, as it tends to give a more faithful representation of the global structure of the data, it is more appealing for the phylogeny forests clustering problem. To complete our proposed approach and solve the third step mentioned above (graph embedding), we use the matrix of distances S as input to a spectral clustering algorithm, building an undirected weighted graph, and mapping its nodes onto the spectral space formed by the smallest eigenvectors of the Laplacian of this graph, further discussed in Section II-C.

C. Spectral Clustering

Spectral clustering (SC) has its origin on the spectral graph theory, a field dedicated to the study of graphs through the eigenvalues and eigenvectors of matrices naturally associated with them [53], [54]. Spectral methods are closely related to manifold learning and dimensionality reduction, as these approaches include the solution of a sparse eigenvalue problem.

The core idea of SC is to consider clustering as a graph partitioning problem, i.e., a graph is divided into k disjoint groups such that nodes more similar to each other remain connected. Since it makes few or no assumptions regarding the shape of the clusters, SC became a popular approach, being able to handle clusters with convex regions, intertwined spirals, arbitrary nonlinear shapes, and clusters with holes, outperforming traditional clustering algorithms such as k-means [55], [56]. Some applications where it has been successfully used include image segmentation [57], [58], clustering of words [59], [60], and medical image analysis [61].

The main tool used in spectral clustering is the Laplacian matrix L, which is a matrix representation of a graph G, basically defined by the difference of the diagonal matrix of the vertex degrees of G and its adjacency matrix [62], [63]. The eigenvalues and eigenvectors of L are used to find many properties in the graph [53]–[56]. For instance, one important property directly related to the IPF problem and spectral clustering is the multiplicity of the 0 eigenvalue – it corresponds to the number of connected components in the graph, which is a good estimate of the number of clusters (trees) of the IPF being analyzed.

In general, SC algorithms follow the steps below:

1. Construct an undirected similarity graph including all data points as nodes of this graph;
2. Embed the data points into a lower dimensional space d, using the eigenvalues and eigenvectors calculated from the graph Laplacian;
3. Cluster the data points in d using k-means, for instance.

In Step (1), the algorithm receives as input an $n \times n$ symmetric similarity or affinity matrix A, whose values determine how close two points are in the space. A common approach to measure the affinity between the points uses the Gaussian kernel based on the Euclidean distance, with width σ to control the size of the neighborhood. To construct the similarity graph, k-NN or ε-neighborhood can be applied on A to connect the closest neighbors, with the nodes of the graph being the input data points, and the weight of the edges being their similarity values. Once the graph is constructed, the Laplacian of this graph can be calculated. In its standard version, spectral clustering uses the normalized graph Laplacian to transform data before clustering [55], which is given by

$$L = D_G^{-1/2} \times (D_G - W) \times D_G^{-1/2}. \tag{2}$$

In this equation, D_G is a diagonal degree matrix whose element $D_G(i, i)$ is the sum of the i-th row of a $n \times n$ weight matrix W, which can be either an adjacency matrix or a matrix containing the similarity values.

With the calculation of the eigenvalues λ_i in Step (2), such that $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$, the number of clusters k of the set can be obtained by checking the multiplicity of λ_i that are equal to 0. Subsequently, the corresponding k eigenvectors $\mathbf{v}_i$ of λ_1 to λ_k are used to project the data onto a lower dimensional space, using a matrix $U \in \mathbb{R}^{n \times k}$ containing the eigenvectors $\mathbf{v}_1$ to $\mathbf{v}_k$ as columns. Let $(u_i)_{i=\{1..n\}} \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U. In Step (3), cluster u_i using k-means algorithm, for instance, with each point being classified in one of the output clusters $U_1, ..., U_k$.

As spectral clustering uses the top eigenvectors of a feature matrix, in our implementation, we simply consider the geodesic distance matrix S calculated by the two first steps in Isomap as being our affinity matrix A. All subsequent steps follow the standard SC algorithm, whose output are clusters that correspond to the trees of the image phylogeny forest.

III. Image Phylogeny Forests Through Manifold Learning and Spectral Clustering

In previous works, the trees belonging to an IPF are reconstructed iteratively, with edges being added to them at

Algorithm 1 IPF Clustering Using Manifold Learning and Spectral Clustering

Input: n semantically similar images, $n \times n$ dissimilarity matrix M, and the τ value, used to construct the neighborhood graph.
Output: reconstructed IPF $forest$
1: Get the value of the n-th smallest dissimilarity e, and calculate $\varepsilon = \tau \times e$.
2: Construct graph G encompassing all images, such that images m_i and m_j are connected if $d_{ij} < \varepsilon$.
3: Set the graph's edge weights to d_{ij}.
4: Compute the shortest-path distances between all pairs of points using Dijkstra's algorithm [67], starting at each m_i. The result is an $n \times n$ matrix S, with $S(i, j)$ storing the geodesic distance from node i to j.
5: Symmetrize matrix S, obtaining S', and calculate the normalized graph Laplacian L using Equation 2, with $W = S'$.
6: Perform a spectral decomposition on L and select the k eigenvalues, such that $\lambda_1, ..., \lambda_k = 0$. In this case, k represents the number of connected components of the graph.
7: Compute the first k eigenvectors $\mathbf{v}_1, ..., \mathbf{v}_k$ of L, and construct a new $n \times k$ matrix U, with the eigenvectors as columns.
8: Let $(u_i)_{i=\{1..n\}} \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U. Cluster u_i using k-means and assign each of them to one of the output clusters $U_1, ..., U_k$.
9: Reconstruct IPF in $forest$ using the Optimum Branching algorithm [19].
10: **return** $forest$

each step according to a threshold. In this paper, our main goal consists in obtaining a good clustering of images belonging to the same tree prior to the reconstruction step.

Our proposed framework is divided in three main steps, detailed in Algorithm 1: (i) estimation of the pairwise geodesic distances of the data points on the underlying manifold, using as input the dissimilarity matrix M; (ii) projection of the data onto a low dimensional space using spectral clustering, which provides the number of trees in the phylogeny forest, later used in the k-means algorithm to place each image in its cluster; and (iii) reconstruction of each tree in this forest.

A. Calculating Point Distances on the Underlying Manifold

In the multimedia phylogeny framework [6]–[8], [22], [23], [25], the calculation of an asymmetric $n \times n$ dissimilarity matrix M is the starting point to reconstruct phylogeny trees and forests. Up to this point, all approaches have considered the full matrix for the phylogeny reconstruction.

Different to all existing phylogeny methods, our implementation starts by first creating a sparse graph G from M (similar to Isomap). Considering $m_i : \{m_1, ..., m_n\}$ data points in M with pairwise dissimilarities d_{ij}, there are different strategies to construct G, the most common being the k-NN or the ε-neighborhood graph [55]. In the first approach, an integer k is selected and each point has exactly k neighbors, which are the k closest points to itself; in the second approach, a real number ε is selected, and G is constructed by connecting the neighbor points whose distances are smaller than ε. In our implementation, the ε-neighborhood graph presented better results, with $\varepsilon = \tau \times e$, where τ is a parameter obtained from a training dataset (see Section IV), and e is the n-th smallest edge of the set. The edges of the graph are weighted by the dissimilarity values from M, such that we connect only points whose $d_{ij} < \varepsilon$. The result of this step can be also a set of disconnected subgraphs. Subsequently, we applied Dijkstra's algorithm for each m_i as the starting node to infer a matrix S with the shortest-path distances between all pairs of points.

B. Image Clustering

After obtaining matrix S, the next step consists in applying the SC algorithm to find the number of clusters (trees in the forest), as well as assigning each point to its corresponding cluster. However, most of the work found in SC deals with symmetric matrices, and since S is an asymmetric matrix, it is necessary to design a strategy to make it symmetric. A common approach calculates a symmetric matrix S' from the original asymmetric matrix S by making $S' = S + S^T$ [64]. Once matrix S' is obtained, we calculate the normalized Laplacian L using Equation 2, with $W = S'$. To define the number of connected components, we calculate the eigenvalues of L, and the graph is partitioned according to the multiplicity of eigenvalues that are equal to zero [53]. In practice, we have to numerically decide which eigenvalues are close enough to zero. Finally, we apply Ng's algorithm [65] that uses *k-means* on the projected subspace of eigenvectors.

C. IPF Reconstruction

In the phylogeny problem, k represents the number of trees in the forest, and each cluster encloses all images that belong to the same tree in a phylogeny forest. To reconstruct these trees, we simply used the Optimum Branching algorithm [19], taking as edge weights the values from matrix S.

D. Algorithm's Complexity Analysis

In Algorithm 1, Line 1 takes $O(n^2)$ to obtain the n-th smallest dissimilarity value to calculate ε. Lines 2-3 describes the construction of the neighborhood graph G, and takes $O(n^2)$ in the worst case, as all values of M (excluding the main diagonal) are compared to ε. The computation of pairwise shortest-path distances on G in Line 4, using Dijkstra's algorithm as implemented by the Boost Library,[1] takes $O(m + n \log n)$, for a graph with m edges and n vertices. As it is executed n times, each time starting from one of the input data points, the resulting complexity is $O(mn + n^2 \log n)$. For the spectral decomposition described in Lines 5-7, the most expensive step is the eigen decomposition. We used the GSL library,[2] which solves the eigen-system through symmetric bidiagonalization and QR factorization [66], running in $O(n^2)$. In Line 8, the application of k-means in the matrix resulting from the eigenvalue decomposition of L, costs $O(ntk)$, where n is the number of input data points, t is a fixed parameter representing the number of k-means iterations, and k is the number of clusters. The forest reconstruction in Line 9 uses an implementation that follows Tarjan's description[3] [19], running in $O(m \log n)$. In the worst case, considering $m = \Theta(n^2)$ and t as a constant independent of the input size, the approach's complexity is $O(n^3)$.

E. Step-by-Step Example

Consider the set with $n = 16$ semantically similar images illustrated in Figure 2. Given the dissimilarity matrix M in

[1] http://www.boost.org
[2] http://www.gnu.org/software/gsl/
[3] http://edmonds-alg.sourceforge.net

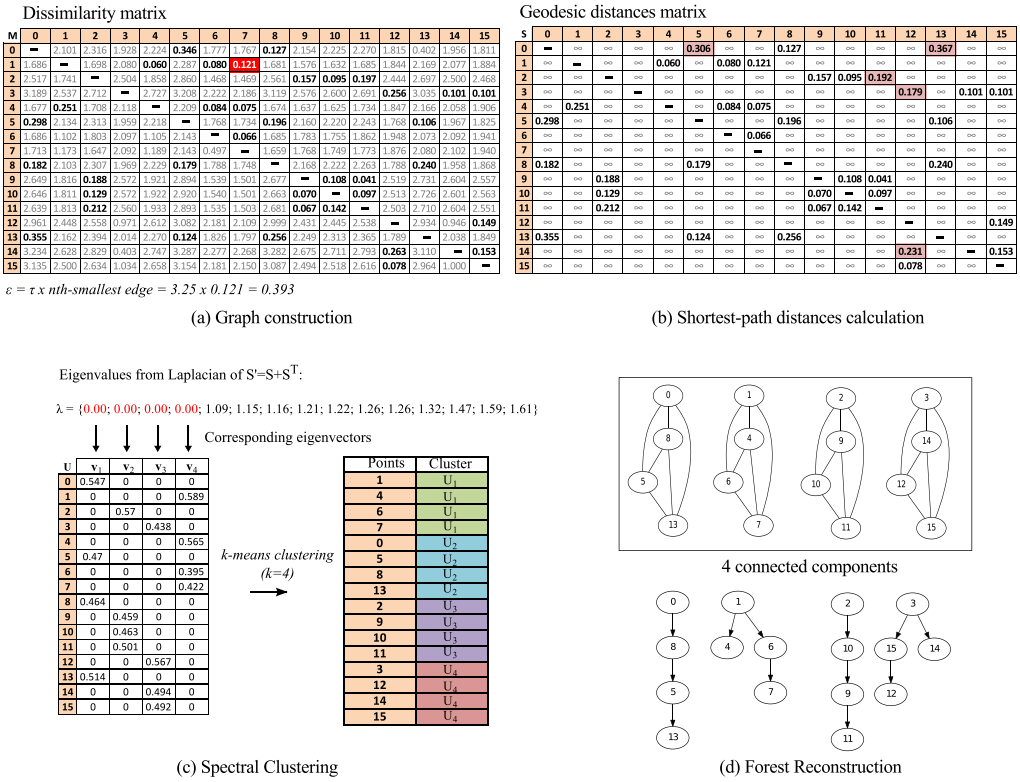


Fig. 2. Step-by-step example: (a) The dissimilarity matrix M is transformed to a sparse matrix by keeping only the edges lower than a threshold ε. The n-th smallest edge used to calculate the value of ε is highlighted in red and, in bold, we highlight the edges kept for the graph construction. The next step consists in (b) computing the matrix of shortest-path distances S between all pairs of points. Highlighted values in S indicate which values from M changed after this calculation. In (c), SC is applied on the symmetrized version of S. Since the number of eigenvalues $\lambda_i = 0$ is four, this is the number of connected components in the graph. Once each point is placed in its cluster using k-means, (d) the phylogeny trees are reconstructed.

Figure 2(a), we start by sorting the edges in ascending order to obtain the value of the n-th smallest edge e. In this example, $e = 0.121$, and its value is used to define the threshold $\varepsilon = \tau \times e = 3.25 \times 0.121 = 0.393$ (by default, $\tau = 3.25$, learned in a training dataset). Therefore, all edges whose values are higher than ε are eliminated from M. The next step consists in calculating the shortest-path distances among all pairs of nodes, creating a new matrix S, depicted in Figure 2(b). After calculating a symmetric version of S, we apply spectral clustering on it, as shown in Figure 2(c): first, we calculate the normalized Laplacian matrix using Equation 2, and obtain its eigenvalues to decide the number of clusters. Since $\lambda_i = 0$ in four of the cases, this is the number of connected components of this graph. Subsequently, we calculate the eigenvectors corresponding to the eigenvalues $\lambda_1 = ... = \lambda_4 = 0$ and stack them column-wise to form a 15×4 matrix. This new matrix is clustered using the k-means algorithm, with $k = 4$ (number of connected components or number of trees in the forest), with each point being assigned to one of the clusters. Lastly, we use the optimum branching algorithm to reconstruct the trees, resulting in the forest depicted in Figure 2(d).

IV. Experiments and Results

This section presents the methodology used to evaluate our proposed approach. In a controlled setup, quantitative metrics were used in three different scenarios, with results being compared to the current state-of-the-art approach. In addition, a qualitative evaluation was performed in an uncontrolled setup using real cases of images published on the Internet.

A. Datasets and Evaluation Metrics

In our setup, we used two types of datasets: a *controlled dataset*, in which we know a priori the ground-truth of the image relationships, and a *real cases dataset*, with semantically similar images collected from the Internet, and from which we often do not have a prior knowledge about their relationship. In the latter case, although we do not have the ground-truth, we can perform a qualitative analysis of the results.

1) Controlled Dataset: In this scenario, we performed experiments considering the standard and publicly available datasets provided by previous works in IPFs [7], [8]. Images in this dataset comprise two different scenarios: with a single camera (OneCam), as well as with different cameras (MultCam), both cases having similar scene semantics (the main content of the image is the same, but with small variations in the position or the camera parameters). For completeness, we also included forests with one tree, and used an expanded training set. Therefore, our training set is composed by the Training Dataset + Dataset A (following the nomenclature used in [7] and [8]), which comprises images from single and multiple camera scenarios, three different scenes,

three different cameras, three images per camera, five forest sizes, four different tree topologies, and ten random variations of parameters for creating the near-duplicate images. In total, $2 \times 3^3 \times 5 \times 4 \times 10 = 13,500$ forests were used to find the parameters ε and τ for the neighborhood graph construction, and the width σ of the Gaussian kernel for spectral clustering.

To find the value of parameter ε, we studied the behavior of Algorithm 1 using the training dataset, varying this threshold in the interval [2..4], separated by steps of 0.25. From this experiment, we defined $\tau = 3.25$. Regarding the parameter σ, we followed the steps of a standard SC algorithm, described in Section II-C, in the interval [1..5], separated by steps of 0.5. Best results were obtained with $\sigma = 1.5$.

To evalute the proposed approach, we used *Dataset B*, which comprises images randomly selected from a set of 20 different scenes, 10 different cameras, 10 images per camera, 10 different tree topologies, 10 random variations of parameters for creating the near-duplicate images, and forests with 10 trees each. The family of image transformations $\mathcal{T}$ considered are the same used in previous approaches [6]–[8]. For each of the cases, single and multiple cameras, a total of 2,000 forests within this set were randomly selected with forests of size $|F| = \{1..10\}$. Therefore, this set comprises $2 \times 2,000 \times 10 = 40,000$ test cases.

2) Real Cases Dataset: For evalution in real case scenarios, we used three different datasets:

(a) *10 Different Target Image Groups From the Internet:* This dataset was originally used to evaluate the reconstruction of IPTs [6]. It comprises images from 10 different target image groups from the Internet that became viral at the time of their publishing. Their descritption can be found in Table II (TG_1 to TG_{10}), and in the supplementary material. The number of images across all groups is 187, and evaluation is performed by creating a direct descendant for each image using five variations of parameters in $\mathcal{T}$. In total, this dataset comprises $187 \times 6 = 1122$ images.

(b) *The Situation Room:* This dataset was introduced in the first work of IPFs [7], and comprises the image taken by the White House photographer Peter Souza on May 1st, 2011, and its variants, collected from the Internet (this episode is also known as *The Situation Room*). 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*).

(c) *The Ellen DeGeneres' Selfie Taken at the 2014 Oscar:* In this work, we introduce a new dataset composed by images related to the *selfie* taken by the host Ellen DeGeneres and some famous actors during the 86th Academy Awards held on March 2nd, 2014. The original image became increasingly popular right after it was posted on her Twitter account. To this day, it has been retweeted more than three million times, and in addition to the retweets, several edited versions of this picture appeared on the Internet, with cases of text overlay, face swap, and insertion of other people and animals in the picture. This dataset is a clear example of an IPF, as the images are semantically similar but they were either taken with different cameras or in different points in time. We manually collected 44 pictures from Twitter, blogs, and news websites, and divided it in five groups (Figure 6):
   - *Group a*:* Edited versions of the picture posted at DeGeneres' Twitter account (@TheEllenShow).
   - *Group b*:* The moment the selfie was being taken but from the viewpoint of another camera.
   - *Group c*:* Similar to group b*, but with slight differences on the posture of the people in the picture. For instance, Angelina Jolie moves her arms and Brad Pitt straightens his back.
   - *Group d*:* Similar to group b* and c*, but the main differences are on their facial expressions.
   - *Group e*:* The moment before the selfie was taken, when the people on the picture starts gathering.

3) Evaluation Metrics: To evaluate the reconstructed IPFs with respect to their ground-truth in the controlled scenario, we consider the four quantitative metrics proposed by Dias *et al.* [7], adapted from the evaluation of IPTs [6]. The ground-truth comprises the parameters used to construct the artificial forests, informing the image relationships (which nodes are the roots, which are the correct edge connections, etc.).

Considering IPF_O the ground-truth forest and IPF_R the reconstructed forest, the *roots* and *leaves* metric evaluate if the same images are pointed as roots and leaves, respectively, in IPF_O and IPF_R. The *edges* metric tests if the same edges (i, j) exist in both sets, and the *ancestry* metric checks if, for each image, the ancestors (parent, grandparent, etc.) are the same in both sets. The evaluation metric

$$\text{EM}(\text{IPF}_\text{O}, \text{IPF}_\text{R}) = \frac{S_\text{O} \cap S_\text{R}}{S_\text{O} \cup S_\text{R}} \tag{3}$$

calculates the intersection of the result returned by the reconstructed forest IPF_R with respect to the ground-truth forest IPF_O, and normalizes it by the union of both sets. S_O and S_R represent the elements in IPF_O and IPF_R, respectively. In the example of Figure 3, (a) illustrates the ground-truth forest, whose roots are nodes $\{0, 1, 2\}$, and (b) illustrates the reconstructed forest, with the corresponding score of each metric calculated using Equation 3.

In addition to these metrics, we also included the metric *depth* to assess the average depth in which the correct roots of the forest are identified. This metric measures the average distance, in number of edges, between each of the original roots in IPF_O and the reconstructed roots in IPF_R, and vice versa. This calculation is done in both ways to also take into account the cases when a non-root is misjudged as a root in 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 to measure false positives cases, i.e., nodes wrongly placed as roots in the reconstructed forest.

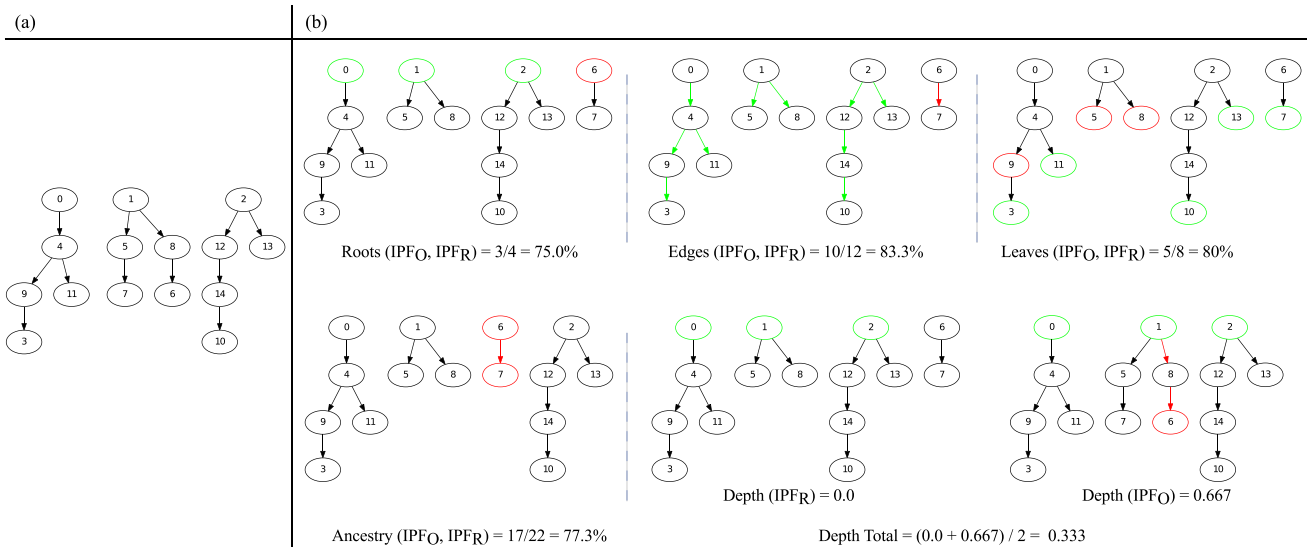


Fig. 3. Evaluation metrics used for IPFs. In green, parts of the trees that coincide with the ground-truth, and in red otherwise. (a) Ground-truth IPF$_O$. (b) Reconstructed forest IPF$_R$.

In the example in Figure 3, we first calculate the depth of the roots in IPF$_R$ with respect to the IPF$_O$. Since $\{0, 1, 2\}$ are the roots of IPF$_O$, and are also correctly found as roots in IPF$_R$, then the depth of each root is 0, and hence, $depth(\text{IPF}_R = 0)$. However, the fact that $\{6\}$ is misdjudged as a root is not considered in this case. To penalize this non-root placed as a root, we also calculate the depth of the roots in IPF$_O$ with respect to the IPF$_R$, which is $depth(\text{IPF}_O) = (0 + 0 + 0 + 2)/3 = 0.667$ (nodes $\{0, 1, 2\}$ have depth zero, but node $\{6\}$ is two edges away to be a root in IPF$_O$. The result is averaged by the number of roots in IPF$_O$). The final result for the metric depth is the average of both values, that is, $depth(\text{IPF}_O, \text{IPF}_R) = \frac{depth(\text{IPF}_R)+depth(\text{IPF}_O)}{2} = (0 + 0.667)/2 = 0.333$.

B. Results and Discussion

1) Controlled Scenarios: We performed the evaluation of the reconstructed phylogeny forests in three different setups:

- *Using Only the Shortest-Path Distances (SPD):* We construct the neighborhood graph G and the shortest-path distances matrix S, but we do not apply spectral clustering on S. Instead, since our goal is the construction of minimal spanning trees, these distances are used as the new weights of the edges of the graph, and given as input to E-AOB algorithm [8].
- *Using Only Spectral Clustering (SC):* We first construct an asymmetric similarity matrix from M, using a Gaussian kernel with $\sigma = 1.5 \times e$ (obtained through experiments in the training dataset). To calculate the graph Laplacian, we symmetrize this similarity matrix and apply SC on it. Lastly, we cluster the images using k-means, and for each cluster, we apply the Optimum Branching algorithm [19] to reconstruct the trees.
- *Combining SPD With SC (SPD+SC):* Combination of approaches (i) and (ii), implemented by following the steps described in Algorithm 1.

In Table I, we present the results obtained for each setup described above, along with results from the state-of-art approach for IPF reconstruction (Fusion AOK $\times$ AOB $\times$ E-AOB) [8]. To compare these results, and assess whether their means differ, a Wilcoxon signed-rank test was performed, as all results are in the same interval, involving repeated measures, and can be paired. We found statistical difference among metrics roots, leaves, and ancestry for SPD and SPD+SC. These results are statistically significant, at 95% confidence level, in favor of the methods proposed in this paper (represented by the blue circles in the last row). On the other hand, for metric edges, no statistical difference was found (represented by the green dashes). The SC used alone presented no statistical difference with respect to the Fusion, in both, OneCam and MultCam scenarios, except for the metric edges in the latter case. For metric leaves, it was not possible to infer statistical difference.

In the first setup using only SPD, we obtained a good result for all number of trees evaluated, except for the case with one tree. This can be explained by the fact that, in the final step, we used E-AOB with the threshold γ [8] for the forest reconstruction. With this threshold, a forest that originally has only one tree ends up, in some cases, being split into two or more trees. We can also observe these results in the Fusion [8], which presents lower results for forests with one tree as well.

On the other hand, for experiments using only SC, as the number of trees increases, the performance of the algorithm decreases, showing that simply using a symmetric version of the dissimilarity matrix as input is not enough to solve the problem. Since our data is naturally asymmetric, this symmetrization step may induce the clustering to errors. As the number of data points n increases, a clustering present in the original asymmetric data may become partially or completely invisible after symmetrization [64], [68]. This problem is more noticeable for the OneCam case. However, with the combination of both approaches, SPD+SC, we are able to obtain improvements specially for forests with a low number of trees, as presented in the corresponding column of Table I. For metric *depth*, SPD+SC also shows robustness in finding

TABLE I

IPF RECONSTRUCTION COMPARISON: FUSION [8], SPD, SC, AND SPD+SC. BEST RESULTS ARE IN BOLD. (A) SEMANTICALLY SIMILAR IMAGES FROM THE SCENARIO USING A SINGLE CAMERA (OC). (B) SEMANTICALLY SIMILAR IMAGES FROM THE SCENARIO USING MULTIPLE CAMERAS (MC)

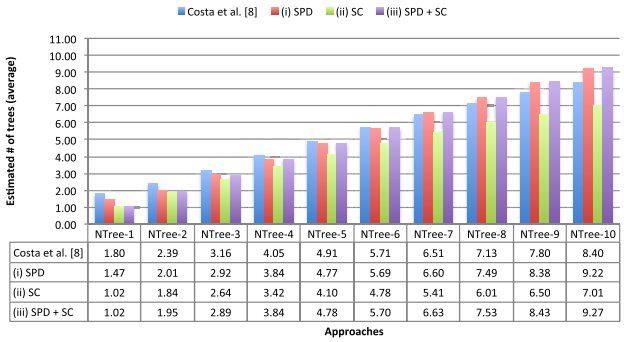
(a)

	Fusion [8]				**Shortest-Path Distances (SPD)**				**Spectral Clustering (SC)**				**SPD+SC**				
$\|F\|$	**Roots**	**Edges**	**Leaves**	**Ancestry**	**Roots**	**Edges**	**Leaves**	**Ancestry**	**Roots**	**Edges**	**Leaves**	**Ancestry**	**Roots**	**Edges**	**Leaves**	**Ancestry**	**Depth**
1 *	0.730	0.829	0.806	0.769	0.826	0.829	0.833	0.830	**0.963**	**0.902**	0.896	**0.907**	**0.963**	**0.902**	**0.899**	**0.907**	0.070
2	0.867	0.884	0.879	0.862	0.912	0.897	0.899	0.883	0.875	**0.901**	0.899	0.866	**0.920**	**0.901**	**0.902**	**0.887**	0.111
3	0.900	0.900	0.896	**0.886**	0.906	0.904	0.904	0.885	0.836	**0.905**	0.900	0.846	**0.909**	**0.905**	**0.905**	**0.886**	0.112
4	0.901	0.905	0.898	0.889	**0.907**	**0.908**	**0.905**	**0.890**	0.808	0.907	0.898	0.833	**0.907**	**0.908**	0.904	**0.890**	0.115
5	0.893	0.905	0.897	0.879	**0.903**	**0.906**	**0.901**	**0.883**	0.775	**0.906**	0.893	0.804	0.902	**0.906**	**0.901**	**0.883**	0.115
6	0.883	0.905	0.894	0.873	**0.897**	**0.906**	**0.897**	**0.881**	0.754	0.905	0.888	0.789	0.895	0.905	0.896	**0.881**	0.122
7	0.870	**0.906**	0.892	0.865	**0.893**	**0.906**	**0.894**	**0.879**	0.728	0.904	0.884	0.774	0.890	0.905	0.893	0.878	0.124
8	0.837	**0.911**	0.894	0.853	**0.885**	0.910	**0.896**	**0.880**	0.704	0.909	0.885	0.763	0.881	0.910	0.895	**0.880**	0.138
9	0.818	**0.911**	0.893	0.838	**0.879**	0.910	**0.896**	**0.874**	0.675	0.908	0.882	0.739	0.875	0.909	0.894	**0.874**	0.145
10	0.796	0.909	0.890	0.820	**0.871**	**0.908**	**0.894**	**0.867**	0.654	0.906	0.879	0.720	0.867	0.907	0.892	**0.867**	0.156
	Wilcoxon				•	–	•	•	–	–	–	–	•	–	•	•	

(b)

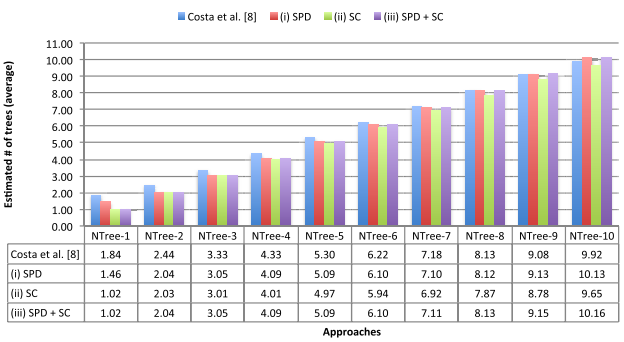
	Fusion [8]				**Shortest-Path Distances (SPD)**				**Spectral Clustering (SC)**				**SPD+SC**				
$\|F\|$	**Roots**	**Edges**	**Leaves**	**Ancestry**	**Roots**	**Edges**	**Leaves**	**Ancestry**	**Roots**	**Edges**	**Leaves**	**Ancestry**	**Roots**	**Edges**	**Leaves**	**Ancestry**	**Depth**
1 *	0.734	0.836	0.796	0.772	0.847	0.837	0.831	0.832	**0.977**	**0.911**	0.900	**0.914**	0.976	0.910	**0.901**	**0.914**	0.060
2	0.871	0.889	0.871	0.869	0.954	**0.906**	**0.896**	**0.906**	**0.956**	**0.906**	**0.896**	0.905	0.955	**0.906**	**0.896**	**0.906**	0.079
3	0.900	0.898	0.880	0.884	**0.947**	**0.905**	0.893	**0.901**	0.941	**0.905**	**0.894**	0.895	**0.947**	**0.905**	0.893	**0.901**	0.082
4	0.905	0.899	0.879	0.891	**0.942**	**0.904**	0.887	**0.903**	0.930	**0.904**	**0.888**	0.894	**0.942**	**0.904**	0.887	**0.903**	0.082
5	0.916	0.900	0.882	0.891	**0.940**	**0.903**	0.887	**0.900**	0.925	**0.903**	**0.888**	0.886	**0.940**	**0.903**	0.887	**0.900**	0.075
6	0.921	0.901	0.883	0.895	**0.939**	**0.902**	**0.885**	**0.901**	0.924	**0.902**	**0.885**	0.887	**0.939**	0.901	**0.885**	**0.901**	0.071
7	0.928	**0.905**	0.885	0.900	**0.940**	**0.905**	**0.887**	**0.906**	0.923	**0.905**	0.886	0.890	0.939	**0.905**	0.886	**0.906**	0.067
8	0.931	**0.910**	**0.888**	0.904	**0.942**	0.909	**0.888**	**0.911**	0.921	**0.910**	0.887	0.891	0.941	0.909	**0.888**	**0.911**	0.066
9	0.926	**0.911**	**0.891**	0.900	**0.938**	0.910	0.890	**0.908**	0.907	0.910	0.888	0.882	0.936	0.910	0.890	**0.908**	0.069
10	0.917	**0.910**	**0.891**	0.890	**0.939**	0.909	0.891	**0.905**	0.897	0.909	0.888	0.871	0.937	0.909	0.890	**0.905**	0.068
	Wilcoxon				•	–	•	•	–	•	–	–	•	–	•	•	

• : There is statistical difference in favor of the proposed approach.
– : There is no statistical difference.
* Results for $|F| = \{1\}$ were not originally presented in the paper by Costa et al. [8]. We performed these experiments for comparison purposes, following the same protocol used to construct the other forest sizes.



	NTree-1	NTree-2	NTree-3	NTree-4	NTree-5	NTree-6	NTree-7	NTree-8	NTree-9	NTree-10
Costa et al. [8]	1.80	2.39	3.16	4.05	4.91	5.71	6.51	7.13	7.80	8.40
(i) SPD	1.47	2.01	2.92	3.84	4.77	5.69	6.60	7.49	8.38	9.22
(ii) SC	1.02	1.84	2.64	3.42	4.10	4.78	5.41	6.01	6.50	7.01
(iii) SPD + SC	1.02	1.95	2.89	3.84	4.78	5.70	6.63	7.53	8.43	9.27

(a)



	NTree-1	NTree-2	NTree-3	NTree-4	NTree-5	NTree-6	NTree-7	NTree-8	NTree-9	NTree-10
Costa et al. [8]	1.84	2.44	3.33	4.33	5.30	6.22	7.18	8.13	9.08	9.92
(i) SPD	1.46	2.04	3.05	4.09	5.09	6.10	7.10	8.12	9.13	10.13
(ii) SC	1.02	2.03	3.01	4.01	4.97	5.94	6.92	7.87	8.78	9.65
(iii) SPD + SC	1.02	2.04	3.05	4.09	5.09	6.10	7.11	8.13	9.15	10.16

(b)

Fig. 4. Comparison regarding the estimated number of trees for (a) OneCam and (b) MultCam scenarios.

the correct root, as well as against false positive roots (cases where a non-root is misdjudged as a root) as, on average, the depth value is very low ($depth(OneCam) = 0.121$, and $depth(MultCam) = 0.072$), meaning we are nearly getting all roots of the IPF correctly. To avoid cluttering Table I, we only presented results for the SPD+SC case. The score for the other methods are described in the supplementary material.

For a better comparison among the number of trees k returned by each method, and to highlight the importance of estimating it correctly, consider the graphics in Figure 4. They show the average of the number of trees returned by the evaluated approaches against the Fusion proposed by Costa *et al.* [8]. In the OneCam case, up to forests with four trees, this Fusion returns, on average,

TABLE II

RESULTS FOR SPD+SC IN THE UNCONTROLLED SCENARIO FOR THE 10 DIFFERENT TARGET IMAGE GROUPS FROM THE INTERNET

	Description	# of Cases	%Er1$^{(a)}$	%Er2$^{(b)}$	%Er3$^{(d)}$	%Er4$^{(d)}$	%P$^{(e)}$
TG1	Iranian Missiles	90	38.89%	11.11%	8.89%	0.00%	55.56%
TG2	Bush reading	95	23.16%	13.68%	7.37%	6.32%	68.42%
TG3	WTC Tourist	95	31.58%	6.32%	6.32%	0.00%	65.26%
TG4	BP Oil Spill	100	25.00%	0.00%	0.00%	0.00%	75.00%
TG5	Israeli-Palestinian peace talks	95	18.95%	9.47%	3.16%	0.00%	77.89%
TG6	Criminal record	90	35.56%	12.22%	6.67%	10.00%	56.67%
TG7	Palin and Rifle	100	22.00%	6.00%	2.00%	0.00%	73.00%
TG8	Beatles	100	9.00%	1.00%	1.00%	0.00%	90.00%
TG9	Kerry and Fonda	80	20.00%	10.00%	10.00%	0.00%	72.50%
TG10	OJ Simpson	90	22.22%	1.11%	1.11%	0.00%	76.67%
	Average	93.5	24.64%	7.09%	4.65%	1.63%	71.10%

(a) **Er1:** The proposed approach is not able to find the correct parent of the aritificial node.
(b) **Er2:** The original tree structure changes after reconstruction.
(c) **Er3:** The solution inserts the artificial node not as a leaf of the tree.
(d) **Er4:** The proposed approach changes the original root after the insertion of the artificial node.
(e) **P:** Perfect reconstruction is achieved, with the algorithm finding the correct relationship for the artificial node.

TABLE III

$\Delta error$ (SPD+SC, FUSION [8]) FOR DATASET B

	OneCam				**MultCam**			
$\|F\|$	**Roots**	**Edges**	**Leaves**	**Anc.**	**Roots**	**Edges**	**Leaves**	**Anc.**
1	-86.2%	-42.9%	-47.8%	-59.9%	-90.9%	-45.7%	-51.6%	-62.4%
2	-40.2%	-14.4%	-18.9%	-18.1%	-64.9%	-15.1%	-19.6%	-28.0%
3	-9.0%	-4.8%	-8.7%	-0.4%	-46.8%	-7.0%	-10.5%	-14.6%
4	-5.8%	-2.9%	-5.9%	-1.2%	-38.5%	-4.2%	-6.4%	-11.5%
5	-8.7%	-1.2%	-3.6%	-3.6%	-28.1%	-2.7%	-4.4%	-8.0%
6	-10.6%	-0.1%	-1.9%	-6.1%	-22.3%	-0.7%	-2.0%	-6.4%
7	-15.1%	0.7%	-1.3%	-9.8%	-15.3%	-0.1%	-1.2%	-5.7%
8	-27.1%	1.3%	-1.1%	-18.4%	-14.3%	0.4%	-0.1%	-6.4%
9	-31.2%	2.1%	-0.8%	-22.1%	-13.5%	1.3%	1.0%	-8.2%
10	-34.9%	2.0%	-1.7%	-26.3%	-23.7%	1.3%	0.8%	-13.9%

more trees than expected, with forests having one tree being the most prone to error. All other approaches return, on average, less or almost the expected number of trees. For forests with more than four trees, excluding SC, the approaches behave similarly, but with SPD and SPD+SC achieving better estimation of the correct value of k. In fact, SC only outperforms the other approaches for forests with one tree. In the MultCam case, a better performance is observed for all methods, with the proposed approaches obtaining better results than the state-of-the-art approach. Overall, with SPD+SC, we obtained significant improvements when calculating the correct number of trees for both, OneCam and MultCam scenarios.

Similar to the comparison made by Costa *et al.* [8], we present results for the error reduction $\Delta error$ between SPD+SC and the Fusion [8], calculated for metrics roots, edges, leaves, and ancestry. For instance, for metric roots

$$\Delta error_{roots}(SPD+SC, Fusion) = \frac{1-(SPD+SC)_{roots}}{1-Fusion_{roots}} - 1. \quad (4)$$

A $\Delta error < 0$ value means that our algorithm SPD+SC performed better than the Fusion, reducing the error according to the percentage shown in Table III.

TABLE IV

RESULTS FOR F={5}, EACH TREE HAVING A DIFFERENT NUMBER OF NODES RANDOMLY CHOSEN IN THE INTERVAL [5], [20]

	SPD+SC			
$\|F\| = \{5\}$	**Roots**	**Edges**	**Leaves**	**Ancestry**
OneCam	0.918	0.908	0.892	0.904
MultCam	0.964	0.898	0.893	0.892

Furthermore, Table IV shows that the number of nodes in the trees does not affect the forest reconstruction. We performed experiments with forests having five trees, each tree having the number of nodes randomly chosen considering a minimum of five and a maximum of 20 images. In total, we created 500 forests, using 50 phylogenies, with 5 variations for each phylogeny in each of the cases, OneCam and MultCam.

2) Real Case Scenarios: Our experiments in these scenarios include only a qualitative evaluation of the proposed approach, since we often do not have any prior knowledge about the existing relationships among the images.

Table II presents the results for the dataset comprising the 10 different target image groups from the Internet, using the proposed SPD+SC approach. The evaluation is performed by artificially generating a near-duplicate and comparing the tree before the insertion of this artificial node with the tree after the insertion of such duplicate. This way, we are able to assess the robustness of the phylogeny methods used.

Following the notation used by Dias *et al.* [6], we evaluated five error metrics, described at the bottom of Table II. Results show that our approach finds the correct relationship of the artificial node, with no further change to the original tree, in $P = 71.10\%$ of the cases. Similar to the results in Dias *et al.* [6], the target groups $TG_{i\in\{1,6\}}$ presented the worse results, but even for them, the solution achieves a reconstruction rate of $P \geq 55.56\%$. In addition, our approach finds the correct parent of the artificial node in about 75.36% of the cases (Er1=24.64%), and in only 7.09% of the cases it changes the original tree structure. Finally, only in 4.65%

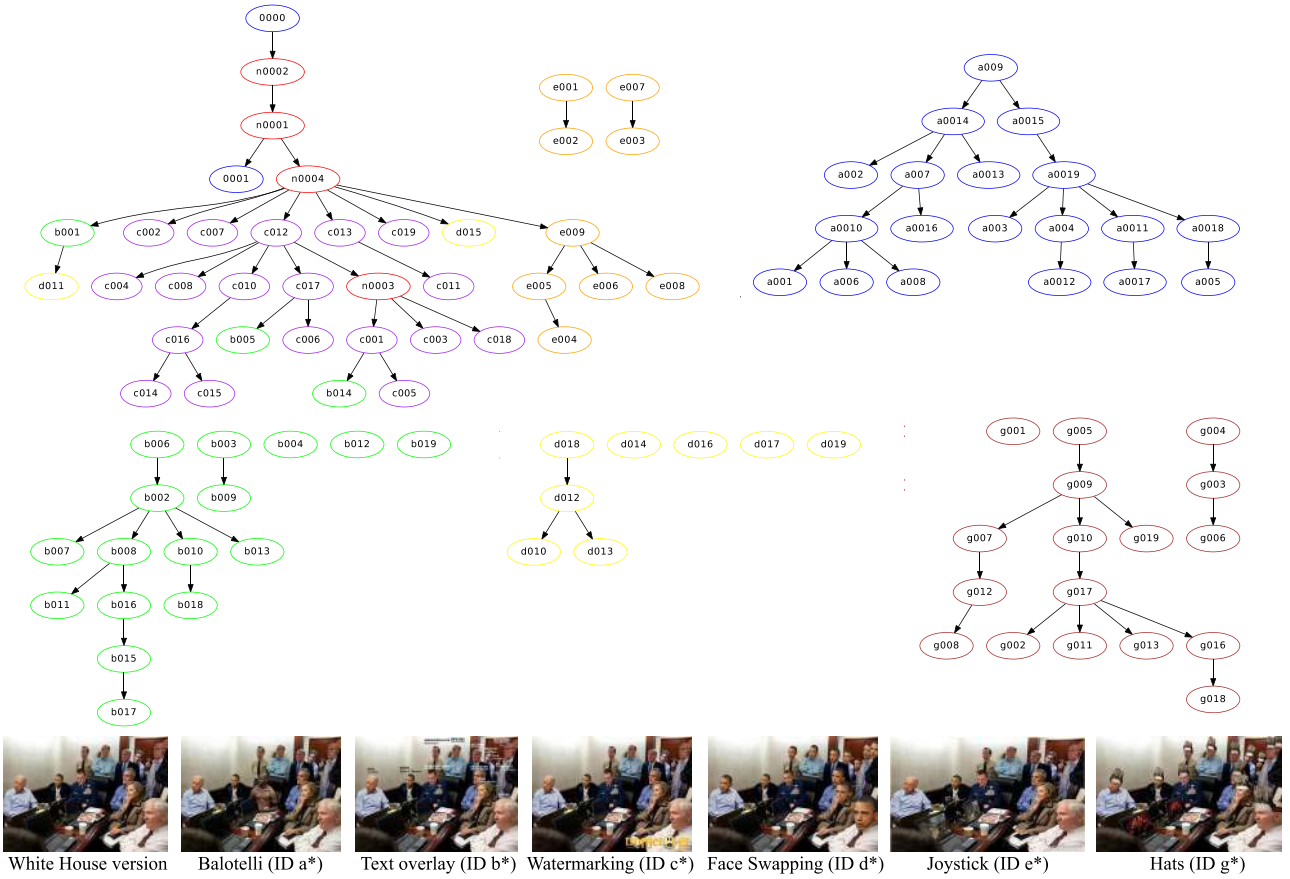


Fig. 5. Reconstructed phylogenetic forest for *The Situation Room* scenario.

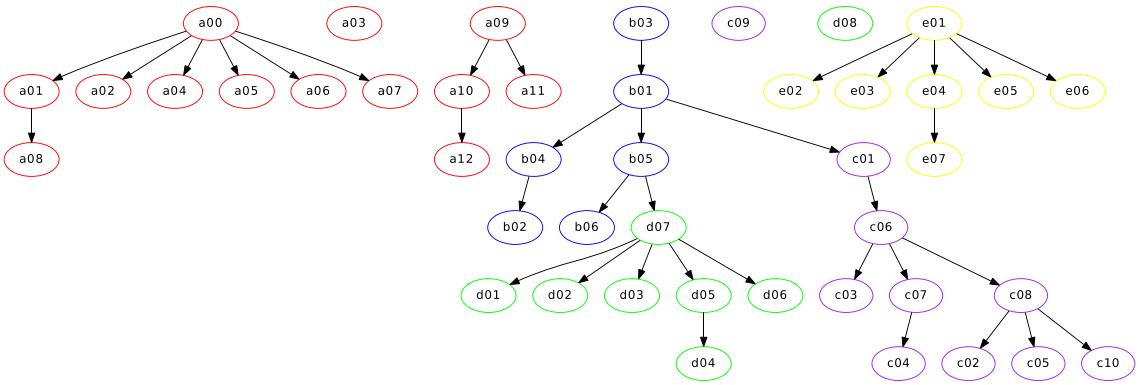
of the cases it inserts the artificial node in another position that is not as a leaf of the tree, and in 1.63% of the cases our approach changes the original root after the insertion of the artificial node.

For *The Situation Room* scenario, Figure 5 shows the reconstructed forest obtained with our approach. The algorithm correctly identified image with ID 0000 (the White House version) as the root of one of the trees. Furthermore, images with IDs n0001, n0002, and n0004 were correctly grouped under image 0000, as the only difference among them is on the image resolution. Althought the expected result was that all images were grouped under the same tree (with image 0000 as the root), it is worth mentioning that this dataset is mostly composed by images with splicing, which is in fact a special case of IPFs (multiple parenting phylogeny [24]). Thus, our approach separated the images into several trees, but with most of the images classified in their correct group. Images from groups with ID c* and e* were the exception (along with isolated cases of images from other groups), being grouped under the same tree, as the difference among the images were very subtle (watermarking and splicing of a small joystick in Obama's hand). Regarding the other trees, our algorithm was robust enough to group together most of the related images.

Figure 6 depicts the reconstructed IPF for the DeGenere's selfie dataset. The images are correctly organized according to their groups, with the roots (for the cases we know which are the correct roots) also correctly identified. For instance, node $a00$ is the picture originally posted at DeGeneres' Twitter account,[4] and it is correctly reconstructed as the root of images $a01$ to $a07$ in the same group. The tree with image $a09$ as root should also be placed as a child of node $a00$, but it has a splicing of a cat in the picture, and the algorithm ended up classifying it in another tree. However, all the images grouped as child of this node are correctly grouped, since image $a09$ is actually a montage also extracted from a Twitter's official account (@RealGrumpyCat[5]), and $a10$, $a11$, and $a12$ are all variants of this image. Groups b*, c* and d* are the hardest to analyze, since there is a subtle difference among them. Although all of them are placed on the same tree, we can see by the color of the nodes that all images belonging to the same group are together, which would help the work of a forensics expert. If we change the threshold τ, we can correctly separate these groups' trees (see supplementary material), albeit the algorithm also removes more nodes from the trees and place them as isolated nodes. Finally, group e* is correctly grouped in a separate tree as the nodes in yellow show.

3) Computational Time: Table V shows a comparison among the average time (in miliseconds) for the IPF reconstruction for all forest sizes. For a fair comparison, results for the Fusion [8] in this Table were computed again following the same protocol used to calculate the computational time of the SPD+SC approach. In all cases, it is possible to notice that the computational time of the new approach was reduced

[4]https://twitter.com/TheEllenShow/status/440322224407314432/photo/1
[5]https://twitter.com/RealGrumpyCat/status/440335332265848835/photo/1



(a) Reconstruction of the Phylogeny Forest

(b) Group a* (c) Group b* (d) Group c* (e) Group d* (f) Group e*

Fig. 6. Reconstructed phylogenetic forest for the *DeGeneres' 2014 Oscar Selfie*, and the respective groups in which the images were divided into: (a) the selfie, and (b)-(f) the same occasion taken from other cameras at different times.

TABLE V
AVERAGE TIME (*milisec.*)

$\|F\|$	Fusion [8]	SPD + SC	$\|F\|$	Fusion [8]	SPD + SC
1	44.09	7.96	6	257.85	71.52
2	84.08	16.66	7	310.56	95.83
3	125.00	26.19	8	366.07	121.46
4	163.77	38.31	9	423.73	154.66
5	209.35	53.03	10	485.02	194.60

in more than half: for the case with the largest number of trees, $|F| = \{10\}$, the proposed approach takes on average 0.195 seconds, against 0.485 seconds of the former approach, for instance. The experiments were performed in a machine with processor Intel Xeon E5645, 2.40 GHz, with 16GB of memory, and running Ubuntu 12.04.5 LTS.

Finally, to allow reproducibility of our experiments, the test cases and the entire source code will be freely available. The datasets are registered on the address: http://figshare.com/articles/Image_Phylogeny_Forests_Reconstruction/1012816 under the accession number http://dx.doi.org/10.6084/m9.figshare.1012816. The source code and documentation are available in a public repository at http://repo.recod.ic.unicamp.br/public/projects.

V. CONCLUSIONS

Multimedia phylogeny is a research field with several applications in digital forensics, helping investigators to find the publisher of an illegal or abusive content, or the original document among a set of related ones, for instance.

In this paper, we explored a new approach for dealing with image phylogeny forests, aiming at correctly separating semantically similar images in their respective groups, each group representing one phylogeny tree. We used a modified version of Isomap algorithm as a pre-processing step, creating an intermediary graph representation of our input data. We later used this new representation with spectral clustering, grouping the images on their corresponding phylogeny trees.

Our approach was first validated and compared with the state-of-art method presented in the literature [8] using quantitative metrics in a controlled scenario. We used a cross-dataset validation protocol in our experiments, training the algorithms in one dataset with different acquisition conditions, and testing them in another dataset, larger and much more complex. Thus, we were able to evaluate our results in realistic conditions and their behavior in such situations, showing that the method generalizes well for different images.

Through the evaluation in real case scenarios, we were able to validate our method regarding different aspects: the phylogeny reconstruction after the insertion of artificial nodes (*10 different target image groups from the Internet*), correct identification of image similarities in several image groups, and their classification accordingly (*The situation room* and *The Ellen DeGeneres' selfie taken at the 2014 Oscar*), validation of the parameters used for the manifold learning, and correct identification of the roots of the trees (for the cases we knew the images source).

The results show that our new approach is a competitive solution for image phylogeny forests reconstruction, achieving better or equivalent performance when compared to the state-of-the-art approach. The use of a sparse version of the dissimilarity matrix to better determine the neighborhood relationship among the images allowed us to work with less data without affecting the accuracy of the phylogeny

trees reconstruction. Furthermore, with the inclusion of experiments with only one tree, we showed that our algorithm is able to deal with IPTs and IPFs without distinction. Another advantage of the new framework is that there is no need to generate any perturbation on the dissimilarity matrix nor combine different phylogeny approaches, as done by the Fusion [8], which requires three different algorithms (AOK, AOB, and E-AOB).

We achieved a significant improvement on the metric *roots*, considered one of the most important for forensic purposes, allowing an investigator to get closer to the source of the content distribution, and find the potential culprits behind it. The metric *ancestry* also presented good results, being important to trace the chain of suspects involved in an illegal activity, for instance. However, depending on the nature of the problem, other approaches might be more appropriate. For instance, if metrics *edges* and *leaves* are more important to some investigation, or if we are dealing with a very large set of images, SPD approach might be more efficient than the proposed SPD+SC, since the latter includes an eigen decomposition in the SC step, while SPD only computes the shortest-distances and reconstruct the trees using E-AOB.

With the improvements obtained on the IPF reconstruction, we can further improve other methods that include this reconstruction step in its pipeline, such as the multiple parenting problem recently tackled by Oliveira *et al.* [24]. The extension to other types of media, such as video and text, is also possible given a proper calculation of their dissimilarity matrix.

As future work, one interesting problem to be explored is how to deal with asymmetric matrices in the SC step. Most of the existing approaches use symmetric matrices, with few works dealing with asymmetric data. In these cases, the feature matrix is symmetrized before applying spectral clustering, as we have also done in this paper, which can be the cause of the low performance for a large number of trees for the case of SC used alone. Therefore, it is paramount to develop methods that can deal with an asymmetric Laplacian matrix, not only for the multimedia phylogeny framework, but also to improve this type of clustering in other research fields.

REFERENCES

[1] R. Xu and D. Wunsch, II, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.

[2] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Trans. Knowl. Discovery Data*, vol. 3, no. 1, 2009, Art. ID 1.

[3] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[4] A. Fahad *et al.*, "A survey of clustering algorithms for big data: Taxonomy and empirical analysis," *IEEE Trans. Emerg. Topics Comput.*, vol. 2, no. 3, pp. 267–279, Sep. 2014.

[5] C. C. Aggarwal and C. K. Reddy, Eds., *Data Clustering: Algorithms and Applications*. Boca Raton, FL, USA: CRC Press, 2014.

[6] Z. Dias, A. Rocha, and S. Goldenstein, "Image phylogeny by minimal spanning trees," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 774–788, Apr. 2012.

[7] Z. Dias, S. Goldenstein, and A. Rocha, "Toward image phylogeny forests: Automatically recovering semantically similar image relationships," *Forensic Sci. Int.*, vol. 231, nos. 1–3, pp. 178–189, 2013.

[8] F. D. O. Costa, M. A. Oikawa, Z. Dias, S. Goldenstein, and A. Rezende de Rocha, "Image phylogeny forests reconstruction," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 10, pp. 1533–1546, Oct. 2014.

[9] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognit. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.

[10] M. J. Black, D. J. Fleet, and Y. Yacoob, "Robustly estimating changes in image appearance," *Comput. Vis. Image Understand.*, vol. 78, no. 1, pp. 8–31, 2000.

[11] F. De La Torre and M. J. Black, "A framework for robust subspace learning," *Int. J. Comput. Vis.*, vol. 54, nos. 1–3, pp. 117–142, 2003.

[12] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[13] K. Simonyan, O. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher vector faces in the wild," in *Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 8.1–8.12.

[14] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1470–1477.

[15] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[16] R. Benmokhtar, J. Delhumeau, and P.-H. Gosselin, "Efficient supervised dimensionality reduction for image categorization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2013, pp. 2425–2428.

[17] J. Weston, F. Ratle, H. Mobahi, and R. Collobert, *Deep Learning via Semi-supervised Embedding*. Berlin, Germany: Springer-Verlag, 2012, pp. 639–655.

[18] T. Brosch and R. Tam, "Manifold learning of brain MRIs by deep learning," in *Proc. Med. Image Comput. Comput.-Assist. Intervent.*, 2013, pp. 633–640.

[19] R. E. Tarjan, "Finding optimum branchings," *Networks*, vol. 7, no. 1, pp. 25–35, 1977.

[20] A. Joly, O. Buisson, and C. Frélicot, "Content-based copy retrieval using distortion-based probabilistic similarity search," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 293–306, Feb. 2007.

[21] J. R. Kender, M. L. Hill, A. Natsev, J. R. Smith, and L. Xie, "Video genetics: A case study from YouTube," in *Proc. Int. Conf. Multimedia*, 2010, pp. 1253–1258.

[22] Z. Dias, A. Rocha, and S. Goldenstein, "First steps toward image phylogeny," in *Proc. IEEE Int. Workshop Inf. Forensics Security*, Dec. 2010, pp. 1–6.

[23] Z. Dias, S. Goldenstein, and A. Rocha, "Exploring heuristic and optimum branching algorithms for image phylogeny," *J. Vis. Commun. Image Represent.*, vol. 24, no. 7, pp. 1124–1134, 2013.

[24] A. Oliveira *et al.*, "Multiple parenting identification in image phylogeny," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 5347–5351.

[25] Z. Dias, S. Goldenstein, and A. Rocha, "Large-scale image phylogeny: Tracing image ancestral relationships," *IEEE Multimedia*, vol. 20, no. 3, pp. 58–70, Jul./Sep. 2013.

[26] M. Nucci, M. Tagliasacchi, and S. Tubaro, "A phylogenetic analysis of near-duplicate audio tracks," in *Proc. IEEE 15th Int. Workshop Multimedia Signal Process.*, Sep./Oct. 2013, pp. 99–104.

[27] Z. Dias, A. Rocha, and S. Goldenstein, "Video phylogeny: Recovering near-duplicate video relationships," in *Proc. IEEE Int. Workshop Inf. Forensics Security*, Nov./Dec. 2011, pp. 1–6.

[28] S. Lameri *et al.*, "Who is my parent? Reconstructing video sequences from partially matching shots," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 5342–5346.

[29] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.

[30] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[31] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Comput. Graph. Appl.*, vol. 21, no. 5, pp. 34–41, Sep./Oct. 2001.

[32] A. N. Gorban, B. Kégl, D. C. Wunsch, and A. Zinovyev, Eds., *Principal Manifolds for Data Visualization and Dimension Reduction*, vol. 58. Berlin, Germany: Springer-Verlag, 2008.

[33] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, Mar. 2005.

[34] J. Chen, R. Wang, S. Yan, S. Shan, X. Chen, and W. Gao, "Enhancing human face detection by resampling examples through manifolds," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 37, no. 6, pp. 1017–1028, Nov. 2007.

[35] L. Ma, M. M. Crawford, and J. Tian, "Local manifold learning-based k-nearest-neighbor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4099–4109, Nov. 2010.

[36] X. Wang, L. Ma, and F. Liu, "Laplacian support vector machine for hyperspectral image classification by using manifold learning algorithms," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2013, pp. 1027–1030.
[37] Y. Y. Tang, H. Yuan, and L. Li, "Manifold-based sparse representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 12, pp. 7606–7618, Dec. 2014.
[38] V. Jain and L. K. Saul, "Exploratory analysis and visualization of speech and music by locally linear embedding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 3. May 2004, pp. III-984–III-987.
[39] J. Chen and S. Zhang, "Manifold learning-based phoneme recognition," in *Proc. Int. Conf. Image Anal. Signal Process.*, Apr. 2009, pp. 308–312.
[40] J. Martinez-del-Rincon, M. Lewandowski, J.-C. Nebel, and D. Makris, "Generalized Laplacian eigenmaps for modeling and tracking human motions," *IEEE Trans. Cybern.*, vol. 44, no. 9, pp. 1646–1660, Sep. 2014.
[41] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
[42] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
[43] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 14. 2001, pp. 585–591.
[44] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis, "Diffusion maps, spectral clustering and eigenfunctions of Fokker–Planck operators," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 18. 2005, pp. 955–962.
[45] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 10, pp. 5591–5596, 2003.
[46] V. de Silva and J. B. Tenenbaum, "Unsupervised learning of curved manifolds," in *Nonlinear Estimation and Classification*, vol. 171. New York, NY, USA: Springer-Verlag, 2003, pp. 453–465.
[47] Y. Bengio, J.-F. Paiement, and P. Vincent, "Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 177–184.
[48] M. H. C. Law and A. K. Jain, "Incremental nonlinear dimensionality reduction by manifold learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 377–391, Mar. 2006.
[49] B. Ribeiro, A. Vieira, and J. C. das Neves, "Supervised Isomap with dissimilarity measures in embedding learning," in *Proc. 13th Iberoamer. Congr. Pattern Recognit.*, 2008, pp. 389–396.
[50] B. Peherstorfer, D. Pfluger, and H.-J. Bungartz, "A sparse-grid-based out-of-sample extension for dimensionality reduction and clustering with Laplacian eigenmaps," in *Proc. Adv. Artif. Intell.*, vol. 7106. Berlin, Germany: Springer-Verlag, 2011, pp. 112–121.
[51] H. Strange and R. Zwiggelaar, *Open Problems in Spectral Dimensionality Reduction*. Berlin, Germany: Springer-Verlag, 2014.
[52] M. A. A. Cox and T. F. Cox, "Multidimensional scaling," in *Handbook of Data Visualization*. Berlin, Germany: Springer-Verlag, 2008, pp. 315–347.
[53] F. Chung, *Spectral Graph Theory*. Providence, RI, USA: AMS, 1997.
[54] D. Spielman, "Spectral graph theory," in *Combinatorial Scientific Computing*. London, U.K.: Chapman & Hall, 2012, ch. 18, pp. 495–524.
[55] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
[56] J. Liu and J. Han, "Spectral clustering," in *Data Clustering: Algorithms and Applications*. Boca Raton, FL, USA: CRC Press, 2014, ch. 8, pp. 177–199.
[57] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[58] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," *Int. J. Comput. Vis.*, vol. 43, no. 1, pp. 7–27, 2001.
[59] C. Brew and S. S. im Walde, "Spectral clustering for German verbs," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2002, pp. 117–124.
[60] Y. Liu, W. Nan, and T. Zheng, "Spectral clustering for Chinese word," in *Proc. 6th Intl. Conf. Fuzzy Syst. Knowl. Discovery*, Aug. 2009, pp. 529–533.
[61] T. Schultz and G. L. Kindlmann, "Open-box spectral clustering: Applications to medical image analysis," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2100–2108, Dec. 2013.
[62] R. Bapat, "The Laplacian matrix of a graph," *Math. Student*, vol. 65, nos. 1–4, pp. 214–223, 1996.
[63] R. Merris, "Laplacian graph eigenvectors," *Linear Algebra Appl.*, vol. 278, nos. 1–3, pp. 221–236, 1998.
[64] M. Meila and W. Pentney, "Clustering by weighted cuts in directed graphs," in *Proc. SIAM Int. Conf. Data Mining*, 2007, pp. 135–144.
[65] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 849–856.
[66] G. H. Golub and C. F. Van Loan, "Symmetric eigenvalue problems," in *Matrix Computations*, 4th ed. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 2013.
[67] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, 1959.
[68] W. Pentney and M. Meila, "Spectral clustering of biological sequence data," in *Proc. 20th Nat. Conf. Artif. Intell.*, 2005, pp. 845–850.

Marina A. Oikawa received the B.Sc. degree in computer science from the Federal University of Para, Brazil, in 2006, and the master's and Ph.D. degrees in engineering from the Nara Institute of Science and Technology, Japan, in 2010 and 2013, respectively. She is currently a Post-Doctoral Researcher with the Institute of Computing, University of Campinas. Her main research interests include digital forensics, computer vision, and computer graphics.

Zanoni Dias received the B.Sc. and Ph.D. degrees in computer science from the University of Campinas (Unicamp), Brazil, in 1997 and 2002, respectively. He is currently an Associate Professor with the Institute of Computing, Unicamp. His main interests include theoretical computer science, bioinformatics, and computational molecular biology.

Anderson de Rezende Rocha received the B.Sc. degree from the Federal University of Lavras, Brazil, in 2003, and the M.S. and Ph.D. degrees from the University of Campinas (Unicamp), Brazil, in 2006 and 2009, respectively, all in computer science. He is currently an Associate Professor with the Institute of Computing, Unicamp. His main interests include digital forensics, reasoning for complex data, and machine intelligence.

Siome Goldenstein received the M.Sc. degree in computer science from the Pontifícia Universidade Católica do Rio de Janeiro, Brazil, in 1997, the Electronic Engineering degree from the Federal University of Rio de Janeiro, in 1995, the Ph.D. degree in computer and information science from the University of Pennsylvania, in 2002. He is currently an Associate Professor with the Institute of Computing, University of Campinas, Brazil. His research interests lie in computer vision, computer graphics, forensics, and machine learning.