Pre-print of article that will appear in T-IFS.

# Image Phylogeny by Minimal Spanning Trees

Zanoni Dias, Anderson Rocha, *Member, IEEE,* Siome Goldenstein, *Senior Member, IEEE*

***Abstract*—Nowadays, digital content is widespread and also easily redistributable, either lawfully or unlawfully. Images and other digital content can also mutate as they spread out. For example, after images are posted on the internet, other users can copy, resize and/or re-encode them and then repost their versions, thereby generating similar but not identical copies. While it is straightforward to detect exact image duplicates, this is not the case for slightly modified versions. In the last decade, some researchers have successfully focused on the design and deployment of near-duplicate detection and recognition systems to identify the cohabiting versions of a given document in the wild. Those efforts notwithstanding, only recently have there been the first attempts to go beyond the detection of near-duplicates to find the structure of evolution within a set of images. In this paper, we tackle and formally define the problem of identifying these image relationships within a set of near-duplicate images, what we call Image Phylogeny Tree (IPT), due to its natural analogy with biological systems. The mechanism of building IPTs aims at finding the structure of transformations and their parameters if necessary, among a near-duplicate image set, and has immediate applications in security and law-enforcement, forensics, copyright enforcement, and news tracking services. We devise a method for calculating an asymmetric dissimilarity matrix from a set of near-duplicate images and formally introduce an efficient algorithm to build IPTs from such a matrix. We validate our approach with more than 625,000 test cases, including both synthetic and real data, and show that when using an appropriate dissimilarity function we can obtain good IPT reconstruction even when some pieces of information are missing. We also evaluate our solution when there are more than one near-duplicate set in the pool of analysis and compare to other recent related approaches in the literature.**

***Index Terms*—Image Phylogeny Tree; Image Phylogeny; Image Dependencies; Image's Ancestry Relationships; Near-Duplicate Detection and Recognition; Near-Duplicates Kinship**

## I. Introduction

The Internet and social networking have become popular in the last two decades. Because of the simplification of authoring and sharing tools that are now available, publishing and sharing new ideas online has never been so easy. As an example, the Flickr online photo sharing service receives 5.2 thousand high-resolution images per minute and the Youtube video service receives a staggering 24 minutes of video per second[1]. Unfortunately, the ease of sharing and distribution also has its consequences. Images and video content can "mutate" as they spread out and some of the modified versions are not always authorized. For example, after images are posted on the internet, other users can copy, resize and/or re-encode them and then repost their versions, thereby generating similar but not identical copies. Sometimes the creation of such duplications infringe copyrights or even spread illegal content over the internet.

Although the detection of exact image duplicates is straightforward, this is not the case for slightly modified duplicates [1], [2]. In the last decade, several research groups have successfully focused on the design and deployment of systems to identify the cohabiting versions of a given document in the wild. The literature has named this family of problems as *near-duplicate detection and recognition* (NDDR) of a document. The problem of *detection* verifies whether or not two documents are near copies of each other. The problem of *recognition* finds all member documents that are near copies of a given query in a large collection of documents.

NDDR techniques are a first step for several applications such as:

1) reducing the number of versions of a document – for storage and management purposes;
2) tracking the legal distribution and spread of a document on the Internet;
3) copyright and intellectual property protection; and
4) illegal material detection and apprehension.

A far more challenging task which has been vastly overlooked until recently, arises when we want to identify which document is the *original* within a set of near-duplicates, and the structure of generation of each near-duplicate. Only recently have there been the first attempts to go beyond the detection and recognition part of the problem to identify the structure of relationships within a set of near-duplicates [3], [4], [5]. In this scenario, given a set of near-duplicate images, we are interested in the history of the transformations that generated these images. This problem has direct applications in the following areas:

- **Security:** the modification's graph of a set of documents provides information of suspects' behavior, and points out the directions of content distribution.
- **Forensics:** better results can be achieved if the analysis is performed in the original document instead of in a near-duplicate [6].
- **Copyright enforcement:** traitor tracing without the requirement of source control techniques such as watermarking or fingerprinting.
- **News tracking services:** the near-duplicate relationships can feed news tracking services with key elements for determining the opinion forming process across time and space [4], [5].

The identification of the original document's generating structure of modifications has a natural analogy with the biological evolution process. In biology, we can look at the process of evolution as a branching process, whereby populations

The authors are with the Institute of Computing, University of Campinas, Av. Albert Einstein, 1251 – Campinas, São Paulo, Brazil, 13083-970.

**Corresponding authors**: The authors can be reached through {zanoni, anderson.rocha, siome}@ic.unicamp.br

Manuscript submitted on May 4th, 2011.

[1]Statistics collected directly on the websites on January 25th, 2011.

change over time and may speciate into separate branches. We can visualize the branching process as a phylogenetic tree [7].

Just as organisms evolve in biology, a document can change over time to slightly different versions of itself, where later on each of these versions can generate other versions. Sometimes the changes are unintentional such as those resulting from copying errors inserted during the process of preservation or duplication of manuscripts, such as the famous case of the *Book of Soyga* [8], common errors on logarithm tables replications [9], [10], or even copyist errors in ancient documents [11], [12]. In other situations, the changes are intentional such as when a forger modifies a document with the intent of deceiving the viewer [13].

In this context, given a set of near-duplicate images, we would like to identify their causal relationships and transformations. We name this problem 'Image Phylogeny'. This structure (Phylogeny Tree) tells the evolving history of the image, even when some pieces, or connections, are missing. With an Image Phylogeny Tree (IPT) we aim at finding the structure of transformations, and possibly their parameters, that embodies the phylogenetic relationships within a given set of near-duplicate images.

Under proper assumptions, we could approach this problem with *watermarking* and *fingerprinting* techniques [14]. We would be able to analyze the document's markings to recover its history in case the document leaks onto the internet, potentially having multiple variations. This is known in the literature as *traitor tracing*.

However, traitor tracing and watermarking solutions are not always possible:

1) transformations on the document can destroy its markings;
2) awareness of markings allows attempts to circumvent them;
3) watermarking only works for documents reproduced after their adoption, leaving copies before adoption unidentifiable;
4) it is not always possible to assume knowledge about the ownership of the source.

In this paper, we show that to build an image phylogeny tree with respect to a set near-duplicate images, we need to think of this problem under two vantage points: first, we need to find a robust and informative dissimilarity function to compare the image duplicates; second, we need to devise a proper algorithm capable of creating such a tree from the dissimilarities available to it. For the first problem, we devise a method for calculating an asymmetric dissimilarity matrix from a set of near-duplicate images and, for the second problem, we introduce an algorithm to find the phylogenetic relationships of the images based on the calculated dissimilarities. Our new algorithm is based on a modified *Minimum Spanning Tree* method that has a low computational footprint (polynomial complexity). As we believe there is further interest for this problem, we also define and use an evaluation methodology that includes several quantitative metrics to evaluate a reconstructed tree for a given set of near-duplicate images even when some pieces of information are missing.

Section II presents related work in the literature of near-duplicate detection and recognition as well as the most recent community efforts to trace down the image relationships within a set of near-duplicate images. Section III defines the problem of image phylogeny and its properties. Section IV formally introduces our solution toward image phylogeny trees. Section V formalizes the methodology used in the experiments. Finally, Section VI presents the performed experiments while Section VII concludes the paper and discusses possible future work and further improvements.

## II. Related Work

A near-duplicate is a transformed version of a document that remains recognizable. Joly et al. [2] formally proposes a definition of what a duplicate is, based on the notion of *tolerated transformations*. According to the authors, a document $\mathcal{D}_1$ is a near-duplicate of a document $\mathcal{D}$, if $\mathcal{D}_1 = T(\mathcal{D})$, $T \in \mathcal{T}$, where $\mathcal{T}$ is a set of tolerated transformations. $\mathcal{D}$ is called the original document, patient zero, or the root of the document evolution tree.

A family of transformations $\mathcal{T}$ can contain several combinations of transformations such as $\mathcal{D}_3 = T_3 \circ T_2 \circ T_1(\mathcal{D})$, $T_{\beta=1,2,3} \in \mathcal{T}$. Given an original document $\mathcal{D}$, we can construct a tree of all its near-duplications as Figure 1 depicts.

A duplicate is a pairwise equivalence relationship. It links the original document to any of its variations through a transformation (e.g., warping and cropping, compression, color and intensity correction and adjustments). If a document $\mathcal{D}$ has a direct duplicate $\mathcal{D}_1$ and $\mathcal{D}_1$ has a direct duplicate $\mathcal{D}_2$, then document $\mathcal{D}_2$ is, in turn, a duplicate of document $\mathcal{D}$.

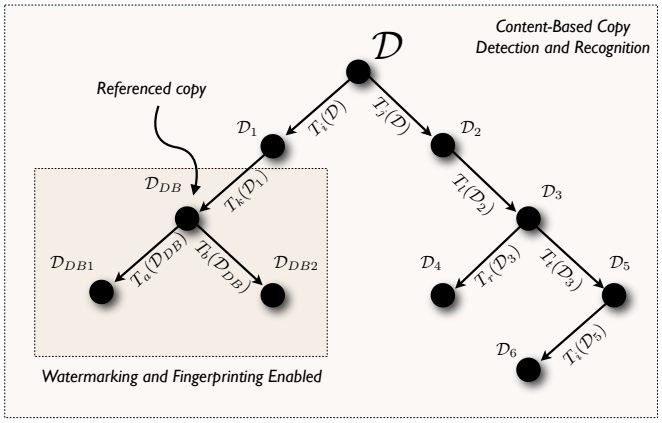



Figure 1. Near-duplicate tree of a document $\mathcal{D}$ and its transformations according to the formalism presented in Joly et al [2]. If we can embed a marking on a given document, we can track its transformations easily. On the other hand, when no markings are available or possible, we can only use content-based copy detection and recognition methods.

Basically, there exists two different near-duplicate detection philosophies: *watermarking-* and *fingerprinting-based* and *content-based* approaches. Watermarking and fingerprinting methods rely on the embedding of a signature within the original document before its dissemination [14]. With watermarking and fingerprinting methods, it is possible to detect the original marked artwork by checking the signature's presence and the modifications patterns within the documents. In

contrast, content-based methods rely on the analysis of the document's content in order to extract relevant visual features. These methods identify when a set of features are close to those of the original document [1], [2]. In Figure 1, if we have a document $\mathcal{D}_{DB}$ which is watermarked and fingerprinted (referenced) we are able to actively trace it as well as its descendants by analyzing the patterns left on the markings themselves which, by construction, we have access to. On the other hand, those documents that do not contain any markings only can be traced through content-based copy detection and recognition approaches such as documents $\mathcal{D}_{1...6}$ in Figure 1.

Regardless of the general philosophy, several near-duplicate detection and retrieval methods (NDDR) have been introduced in the last few years for a variety of applications. NDDR methods have been used for consumer photograph collections organization [15], [16], multimedia linking [17], copyright infringement detection in images or videos [18], [19], [20], and forged image detection [21], [13].

As a matter of fact, in the past decade we have seen some exceptional progress toward the development of efficient and effective systems to identify the cohabiting versions of a given document in the wild. However, only recently have there been the first attempts to go beyond NDDR, with attempts to identify the structure of relationships within a set of near-duplicates [3], [4], [5].

Kennedy et al. [5] were the pioneers at addressing the problem of parent-child relationships between pairs of images. The authors claim that although we might not know the correct parent-child relationship between two images, their low-level pixels content give significant clues about plausible parent-child relationships. In addition, there is an important observation that image manipulations are directional: a highly manipulated version of an image cannot derive a child with less manipulation artifacts.

Kennedy et al. propose to detect plausible parent-child relationships within a set of images with the use of several specialized detectors (e.g., scaling, cropping, overlay, color processing). Across several related images, the detectors responses give rise to a graph structure, denoted a Visual Migration Map (VMM), representing an approximation of the history of the images. On a VMM, a directional edge exists between two images if and only if all the detectors agree about the direction of manipulations ($\mathcal{I}_A \rightarrow \mathcal{I}_B$). Nodes with a high number of incoming edges may correspond to images with high probability of being the result of image manipulations. On the other hand, nodes with several outgoing edges may represent images closer to the original version.

Relying only on directional processing information of pairs of images is rather limiting. With a VMM there is no measure of confidence for a plausible parent-child relationship or even candidate parameters for the family of transformations that lead a parent image to its resulting offspring. Without such information, Kennedy et al.'s approach is useful to point out possible candidates of either *highly-manipulated* or *original images*, but not appropriate to infer the phylogeny of a given set of near-duplicates.

Different from the Visual Migration Map proposed by Kennedy et al. [5], recently two research groups have concurrently focused on the discovery of the structure of dependencies within a set of images and on the underlying conditions of such dependencies [4], [3]. De Rosa et al. [4] aim at exploring image relationships based on the conditional probability of dependency between two images. They use two separate components: the images' content and their content-independent counterparts.

After considering a real forensic case [6], our group [3] has been investigating the use of graph theory and computer vision approaches to identify the generating structure of modifications to images. Please refer to Sections III and IV for more details about our proposed approach for this problem.

De Rosa et al. [4] propose to detect the image dependencies within a set of images $\mathcal{I}$, hypothesizing that any image $\mathcal{I}_i \in \mathcal{I}$ can be univocally described as the composition of two separable and independent parts: $[\mathcal{I}_i^c]$ (the content of the real scene) and $[\mathcal{I}_i^r]$ (content-independent or random part of the image) such as $\mathcal{I}_i \leftrightarrow [[\mathcal{I}_i^c], [\mathcal{I}_i^r]]$, $\forall \mathcal{I}_i \in \mathcal{I}$. Therefore, to verify the dependency between two images $\mathcal{I}_A$ and $\mathcal{I}_B$, the authors consider that the images' mutual information can be expressed as the sum of the mutual information between the content-based components and the content-independent components.

To simplify the hypothesis testing for determining the dependency between two images, the authors consider a family of image processing functions and assume that if there is a dependency between two images $\mathcal{I}_A$ and $\mathcal{I}_B \in \mathcal{I}$, one of the two images can be obtained approximately by applying some of the available image processing functions to the other image. The authors rely on the content part of the images to instantiate the family of functions. After the transformation of one of the images to the other using the instantiated image processing functions, the authors calculate the correlation coefficient between the two images based on their content-independent parts. As the method is asymmetric, the authors choose the direction and correlation value of highest response.

After analyzing all possible pairs of images in $\mathcal{I}$, the authors are left with a dependency graph whose nodes represent images and edges denote the dependency correlation value of the images. As a further processing of the graph, the authors eliminate loops and also edges lower than a specified threshold. In their initial work, De Rosa et al. [4] use a family of image processing functions comprising scaling, rotation, cropping, color transfer, and JPEG compression.

As De Rosa et al.'s approach [4] is related to ours, it is important to point out some of their differences. First, our image phylogeny approach does not rely on any independent component of the image. Relying on such information makes it difficult to differentiate near duplicates acquired by the same device under similar, but different, conditions. Second, our approach uses the complete matrix of dissimilarities to make decisions globally instead of locally eliminating edges. Section VI-C compares the approaches we introduce in this paper to De Rosa et al's. Finally, we introduce a series of quantitative evaluation metrics, and explore the behavior of the algorithms with missing links and roots.

## III. Image Phylogeny

An *Image Phylogeny Tree* describes the structure of transformations and the evolution of near-duplicate images. In this paper, the algorithm builds upon an initial set of near-duplicates and a *dissimilarity function* $d$ that yields small values for ordered pairs that are likely father and son on the tree, and large values for ordered pairs that are unlikely so.

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

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

for all possible values of $\vec{\beta}$ that parameterizes $\mathcal{T}$. Equation 1 measures the amount of residual between the best transformation of $\mathcal{I}_A$ to $\mathcal{I}_B$, according to the family of operations $\mathcal{T}$, and $\mathcal{I}_B$ itself. The comparison is made possible using any point-wise comparison method $\mathcal{L}$ in the end.

The dissimilarity function $d_{\mathcal{I}_A,\mathcal{I}_B}$ is not a distance, and does not form a metric space [22] along the images – it does not satisfy the symmetry and triangle-inequality properties. For example, for the symmetry property case, if the family of transformations $\mathcal{T}$ represents only the spatial rescaling operation in images, reducing the spatial resolution (downsampling) is different than increasing the spatial resolution (upsampling). When $\mathcal{I}_A$ and $\mathcal{I}_B$ have different dimensions, $d_{\mathcal{I}_A,\mathcal{I}_B}$ would calculate the image metric of the residual on $\mathcal{I}_B$'s image dimensions, while $d_{\mathcal{I}_B,\mathcal{I}_A}$ would calculate the image metric of the residual on $\mathcal{I}_A$'s image dimensions.

## IV. Image Phylogeny Reconstruction Process

There are two important and independent factors in the process of reconstructing the image phylogeny tree from a set of near-duplicate images: the tree-building algorithm and the dissimilarity function.

An exceptional dissimilarity function can make a mediocre method shine, while not even the best technique will be able to properly work with an inadequate dissimilarity function.

In this context, our contribution in this paper is threefold: (1) We propose a principled way to build the dissimilarity matrix $M$ measuring how likely a pair of images is related in the tree; (2) We propose an algorithm for the construction of *Image Phylogeny Trees* based on a modified version of a *Minimum Spanning Tree* algorithm [23]; and (3) We formally define an evaluation methodology for assessing contributions for this relatively new problem.

### A. Overview of the Dissimilarity Matrix Construction

Given a set of $n$ near-duplicate images, our first task toward building up the image phylogeny tree is to calculate the dissimilarity between every pair of such images. For that we need to consider a good set of possible image transformations, $\mathcal{T}$, from which one image can generate a descendant. The use of families of transformations has been initially proposed in the image registration literature [24], [25]. In addition, De Rosa et. al. [4] were the first to use this idea to estimate dissimilarities in image phylogeny.

There are countless possible transformations an image can undergo to create a near-duplicate of itself. The family of image transformations $\mathcal{T}$ that we consider in this paper is:

1) **Resampling**: the image can be re-sampled (up or down).
2) **Cropping:** the image can be cropped.
3) **Affine Warping (including rotation, translation, and off-diagonal correction):** the image can be rotated, translated or even slightly distorted.
4) **Brightness and Contrast**: the image pixels' color can be adjusted through brightness and contrast operations.
5) **Lossy Compression**: the image is be compressed using the standard lossy JPEG algorithm.

Given any pair of images $\mathcal{I}_A$ and $\mathcal{I}_B$, our first task consists of estimating the possible values of $\vec{\beta}$ that parameterizes $\mathcal{T}$ that best approximates image $\mathcal{I}_A$ onto $\mathcal{I}_B$ through the dissimilarity function $|\mathcal{I}_B - T_{\vec{\beta}}(\mathcal{I}_A)|_{\mathcal{L}}$ (Eq. 1), according to the chosen point-wise comparison method $\mathcal{L}$. As the dissimilarity is not symmetric, we also calculate $\vec{\beta}$ that parameterizes $\mathcal{T}$ and best approximates image $\mathcal{I}_B$ onto $\mathcal{I}_A$.

To estimate the $\vec{\beta}$ possible values that best approximates image $\mathcal{I}_A$ onto $\mathcal{I}_B$, we

1) calculate the corresponding points between images $\mathcal{I}_A$ and $\mathcal{I}_B$ using the Speeded-Up Robust Features (SURF) algorithm [26];
2) robustly estimate the affine warping transformation parameters which includes translation, rotation, off-diagonal correction, and resampling operations for image $\mathcal{I}_A$ with respect to $\mathcal{I}_B$ taking the corresponding points into consideration and using Random Sampling Consensus (RANSAC) algorithm [27];
3) calculate the mean and variance of each $\mathcal{I}_B$'s color channel and normalize image $\mathcal{I}_A$'s color channels using such measures;
4) compress the result of steps 2 and 3 according to $\mathcal{I}_B$'s quantization table;

Step 1 above gives candidate points to estimate the resampling and cropping operations for $\mathcal{I}_A$ with respect to $\mathcal{I}_B$. However, these points are likely unstable and we need to robustly filter them in Step 2. At this point, we are able to resample and crop image $\mathcal{I}_A$ with an affine warp robustly estimated with RANSAC and the local feature descriptors calculated in Step 1. In Step 3, we perform pixel intensity normalization of image $\mathcal{I}_A$ according to the $\mathcal{I}_B$ color channels' mean and variance. Step 4 compresses image $\mathcal{I}_A$ according to $\mathcal{I}_B$'s quantization table. Finally, to calculate the dissimilarity between both images, we uncompress both of them and calculate their point-wise dissimilarity using the standard *Minimum Squared Error* (MSE) as the technique $\mathcal{L}$.

Note that the experimental procedure that generates the synthetic data (Section V-D) applies more complex transformations (for example, nonlinear gamma correction) than the family $\mathcal{T}$ implemented in the dissimilarity matrix estimation. This simulates real case situations, where we do not have any prior knowledge regarding the image relationships nor their transformations.

To avoid extra possible bias, we use *OpenCV*[2] algorithms

[2]http://sourceforge.net/projects/opencvlibrary/

for the estimation of the dissimilarity matrix while for the data sets' generation process uses uses *ImageMagick*[3] algorithms, see Section V-D for further details.

The final product of the transformation estimations for every pair of near-duplicate images is the dissimilarity matrix $M$ upon which we can calculate the image phylogeny tree as we describe next.

In this paper we do not model the overlay of text or logos, since we do not focus on the construction of the dissimilarity function and families of transformations. Nevertheless, when this type of modification is aimed at watermarking/signing the image, it is applied in small areas on the borders of the images as not to destroy the content – in this case, the error of not modeling the transformation would not dominate the value of the dissimilarity estimate.

### B. Image Phylogeny Algorithm – Oriented Kruskal

The algorithm used to create image phylogeny trees is as important as the dissimilarity matrix related to the $n$ image near-duplicates under analysis. In Section IV-A, we explained how to build a dissimilarity matrix from $n$ image near-duplicates taking into consideration a family of transformations. In this section, we introduce an algorithm for the construction of *Image Phylogeny Trees* based on a modified version of a *Minimum Spanning Tree* algorithm [23]. The original method as proposed in [23] is an exact algorithm for the problem of finding a minimum spanning tree in a non-oriented graph. However, the problem we have here consists of a graph with edges orientation.

In the literature, the problem called Optimal Branching (OB) deals with the construction of minimum spanning trees on directed graphs with known roots. Edmonds [28] and, independently, Chu and Liu [29] proposed an $O(nm)$ algorithm where $n$ is the number of vertices and $m$ the number of edges. This algorithm was further improved [30], [31], [32] and the best known implementation [32] has $O(m + n \log n)$ complexity. In the image phylogeny scenario, we have complete graphs, represented by the full dissimilarity matrix – $m = n^2 - n = O(n^2)$, so basic OB would have a complexity of $O(n^2)$. Additionally, we would need to run OB once for each possible root in the tree, leading to a final complexity of $O(n^3)$.

In this paper, we adopt a different strategy. We propose a heuristic to deal with this problem which we called Oriented Kruskal that does not assume the knowledge of the root. It finds the root and builds the oriented tree in a single execution with complexity $O(n^2 \log n)$. Additionally, the implementation of our algorithm is simpler than OB's, since it does not require a Fibonacci Heap [33].

Although our interpretation of a tree is that the transformation goes from the parent to its child representing the changes of the images, the data structure in our approach keeps the opposite relationship, the edges pointing from child to parent. Recall that in the definition of a tree, each node has exactly one parent, except the root which has no parent.

[3]http://www.imagemagick.org/script/index.php

Given a dissimilarity matrix $M$ built upon a set of $n$ near-duplicate images, the Oriented Kruskal algorithm starts with a forest in which each node (image near-duplicate) is the root of a tree with just one element. Next, the algorithm sorts all positions $(i, j)$ of $M$ from lowest to highest dissimilarity values. The algorithm then analyzes each position $(i, j)$ according to the sorted order, joining different trees and checking whether or not the endpoints of the analyzed position is a root. The algorithm stops when the number of edges in the tree is $n-k$, where $k$ is the number of trees we are interested in. If $k = 1$, we believe the $n$ images under analysis have a common source and, consequently, one IPT (Image Phylogeny Tree). If we can gather further information about the process, we can have a set of IPTs. We discuss this in Section VI-C.

Algorithm 1 presents the operations step-by-step. Lines 1-3 of Algorithm 1 initialize the `tree` vector with $n$ initial trees, each one containing a vertex representing an image. Each position `tree[i]` identifies the parent of a node with `id=i`. At the end, `tree` contains the tree(s) representations. For instance, `tree = [1, 1, 2]` represents a tree with three vertices such that vertex 1 is the root of the tree and also the parent of vertex 2, which in turn, is the parent of vertex 3.

The **for** loop in Lines 6-16 examines matrix positions in order of dissimilarity, from lowest to highest. The **for** loop checks, for each position $(i, j)$, if the endpoints $i$ and $j$ do not belong to the same tree (Test I) and if $j$ is the root of a tree (Test II). If so, we can add the oriented edge $(j \rightarrow i)$ to the forest. Otherwise, we discard such a position.

In the end, all the nodes are connected by $n_{edges} = n - k$ edges forming a set of $k$ trees.

**Algorithm 1** Oriented Kruskal

**Require:** number of near-duplicate images $n$
**Require:** an $n \times n$ dissimilarity matrix $M$ and the number of required IPTs $k$
1: **for** $i \in [1..n]$ **do** ▷ Initialization
2: $tree[i] \leftarrow i$
3: **end for**
4: $sorted \leftarrow$ sort positions $(i, j)$ of $M$ into nondecreasing order
5: $n_{edges} \leftarrow 0$ ▷ Controls stopping criterium
6: **for** each position $(i, j) \in sorted$ **do**
7: **if** ($\texttt{Root}(i) \neq \texttt{Root}(j)$) **then** ▷ **Test I:** joins different trees
8: **if** ($\texttt{Root}(j) = j$) **then** ▷ **Test II:** endpoint must be a root
9: $tree[j] \leftarrow i$
10: $n_{edges} \leftarrow n_{edges} + 1$
11: **end if**
12: **end if**
13: **if** ($n_{edges} = n - k$) **then** ▷ $n - k$ edges were already added
14: **return** $tree$ ▷ Returning the final $k$ IPTs
15: **end if**
16: **end for**

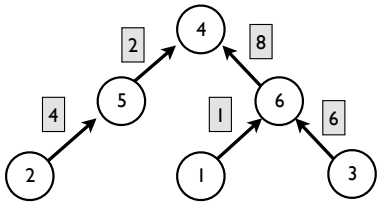
The running time depends on how we implement the `Root` function. If we use a *disjoint-set-forest* with the *union-by-rank* and *path-compression heuristics*, we can implement such a function very efficiently [34].

Lines 1-3 take $O(n)$ time to initialize the `tree` vector. Line 4 takes $O(n^2 \log n)$ to sort the matrix positions since we have $(n^2 - n)$ relevant positions when analyzing $n$ near-duplicate images. Lines 6-16 take $O(n^2 \alpha(n))$ to check any position, where $\alpha(n)$ is the inverse of *Ackermann function*. The amortized cost for finding the root using an implemen-

*Dissimilarity Matrix*

| M | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **1** | - | 31 | 57 | 37 | 45 | 49 |
| **2** | 31 | - | 33 | 23 | 29 | 32 |
| **3** | 51 | 41 | - | 42 | 37 | 38 |
| **4** | 16 | 36 | 28 | - | 15 | 27 |
| **5** | 35 | 18 | 54 | 30 | - | 54 |
| **6** | 12 | 40 | 22 | 60 | 19 | - |

*Reconstructed Tree* [6, 5, 6, 4, 4, 4]




*Algorithm Steps*

| | | | | |
|---|---|---|---|---|
| ✔ | 1 | M[6,1] = | 12 | Select Edge ( 1 → 6 ) |
| ✔ | 2 | M[4,5] = | 15 | Select Edge ( 5 → 4 ) |
| × | 3 | M[4,1] = | 16 | **Test II:** Root(1) = 6 |
| ✔ | 4 | M[5,2] = | 18 | Select Edge ( 2 → 5 ) |
| × | 5 | M[6,5] = | 19 | **Test II:** Root(5) = 4 |
| ✔ | 6 | M[6,3] = | 22 | Select Edge ( 3 → 6 ) |
| × | 7 | M[2,4] = | 23 | **Test I:** Root(2) = Root(4) |
| ✔ | 8 | M[4,6] = | 27 | Select Edge ( 6 → 4 ) |

Figure 2. Simulation of the Oriented Kruskal algorithm to construct an Image Phylogeny Tree ($k = 1$) from a Dissimilarity Matrix.

tation with *union-by-rank* and *path-compression* heuristic is $\alpha(n)$ [34]. Finally, since $\alpha(n) = O(\log n)$, the total Oriented Kruskal's running time is $O(n^2 \log n)$, similar to the well-known, and not oriented, minimum spanning tree algorithm proposed by Kruskal [23].

*Simulation of the Algorithm for one IPT*

Figure 2 depicts the execution of the proposed algorithm for a toy example with $n = 6$ near-duplicate images. The algorithm initially receives a dissimilarity matrix $M$ that captures the dissimilarities between each pair of near-duplicates and the desired number of IPTs, in this example, $k = 1$.

The first step of the algorithm initializes a forest with $n = 6$ roots, one for each duplicate and sorts all the positions $(i, j)$ in the dissimilarity matrix $M$ according to their dissimilarity value. Thereafter, the algorithm starts the construction of the Image Phylogeny Tree taking each position $(i, j)$ at a time and performing the required tests to ensure it can safely insert the tested position as an oriented edge $(j \rightarrow i)$ in the final answer. The algorithm first selects the position $(6, 1)$ with the lowest dissimilarity value $M[6, 1] = 12$. Since this position connects two disjoint trees (Test I) and the endpoint 1 is a root (Test II), it is selected. The same happens with the position (4,5).

The algorithm then tests position (4,1) with dissimilarity 16. Since the endpoint 1 is not a root (it belongs to a tree with root 6), it is discarded. The algorithm proceeds with the selection of the position (5,2) as an oriented edge and discarding the position (6,5) because the endpoint 5 is not a root of any tree. After discarding position (6,5), the algorithm selects the position (6,3) with dissimilarity 22 as an oriented edge and discards the position (2,4) because it joins two nodes belonging to the same tree. Finally, the algorithm ends in Step 8, when testing the position (4,6) with dissimilarity 27.

## V. Evaluation Methodology

In this section, we present the methodology we use to validate the techniques we discuss in this paper. Such methodology and data sets set forth a possible standardization of this relatively new problem allowing researchers to have a concrete measure of progress.

For the image phylogeny tree reconstruction problem, it is important to evaluate solutions under two complementary vantage points: with controlled scenarios in which we can devise several objective measures; and also with uncontrolled scenarios in which we have no clue about the possible transformations of a set of image near-duplicates.

In this paper, we deal with both scenarios. For *Controlled Environments* we create a realistic data set with several transformations an image can undergo to generate a near-duplicate. We analyze the results using several quantitative measures of success. We are also able to evaluate the behavior of the proposed solutions when some of the connections in a phylogeny tree are not present (e.g., when an image has a child and a grand-child but the child is not present in the analyzed near-duplicate image set).

Complementing the *Controlled Environments*, we also present results considering an *Uncontrolled Environment* in which we collect several images from the internet with the same semantical context but we have no clue whether or not they are related at first place. For this challenging task, we also present a principled way to check the reliability and performance of the proposed approach.

### A. Evaluation of Results on Controlled Environments

In this section, we introduce and describe four quantitative metrics (*Root*, *Edges*, *Leaves*, and *Ancestry*) to evaluate a reconstructed tree in scenarios where we have Ground Truth.

*1) IPT vs. Reference IPT:* In the first case we want to compare a reconstructed IPT with its ground truth tree. For this intent, we devise the metrics:

**Root**: $R(\text{IPT}_1, \text{IPT}_2) = \begin{cases} 1, & \text{If } \texttt{Root(IPT}_1\texttt{)} = \texttt{Root(IPT}_2\texttt{)} \\ 0, & \text{Otherwise} \end{cases}$

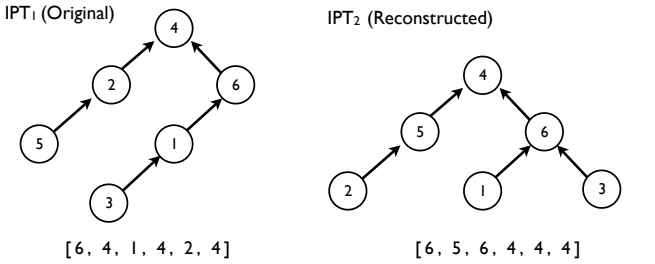
**Edges**: $E(\text{IPT}_1, \text{IPT}_2) = \frac{|E_1 \cap E_2|}{n-1}$

**Leaves**: $L(\text{IPT}_1, \text{IPT}_2) = \frac{|L_1 \cap L_2|}{|L_1 \cup L_2|}$

**Ancestry**: $A(\text{IPT}_1, \text{IPT}_2) = \frac{|A_1 \cap A_2|}{|A_1 \cup A_2|}$

Each of these metrics evaluates a different set of properties of the results, and together allow the practitioner to have a picture of the reconstruction algorithm's overall behavior. As in any designed experiment, we can only calculate them if we do have the real *Ground Truth* (Controlled Environment) to compare the estimated result. In Section V-D, we report the methodological approach we used for obtaining this set of controlled experiments.

In order to describe the above evaluation metrics, we use the illustrative example of Figure 3 of an original image phylogeny tree $\text{IPT}_1$, and its reconstructed version, $\text{IPT}_2$.




| **Trees** (node points to its parent) | |
|---|---|
| $\text{IPT}_1 = [6, 4, 1, 4, 2, 4]$ | $\text{IPT}_2 = [6, 5, 6, 4, 4, 4]$ |
| **Root** | |
| $\text{Root}(\text{IPT}_1) = 4$ | $\text{Root}(\text{IPT}_2) = 4$ |
| **Edges** (oriented edges of children to parents) | |
| $E_1 = \{(1 \to 6), (2 \to 4), (3 \to 1), (5 \to 2), (6 \to 4)\}$ | |
| $E_2 = \{(1 \to 6), (2 \to 5), (3 \to 6), (5 \to 4), (6 \to 4)\}$ | |
| **Leaves** | |
| $L_1 = \{5, 3\}$ | $L_2 = \{1, 2, 3\}$ |
| **Ancestry** | |
| $A_1 = \{(2,5), (4,5), (4,2), (1,3), (6,3), (4,3), (6,1), (4,1), (4,6)\}$ | |
| $A_2 = \{(5,2), (4,2), (4,5), (6,1), (4,1), (6,3), (4,3), (4,6)\}$ | |

Figure 3. On the left, an example of an original tree encompassing an original image and its transformations to create near-duplicates. On the right, the reconstructed phylogeny tree of transformations and near-duplications. On the bottom, a table with details about the data structures necessary to run and evaluate an IPT reconstruction algorithm.

Considering the *Root Evaluation* metric, the reconstructed tree has the correct root node. According to the *Edges Evaluation* metric, the reconstructed tree has $E = \frac{|\{(1\to6),(6\to4)\}|}{5} = \frac{2}{5} = 40\%$ correct edges. In addition, the reconstructed tree has $L = \frac{|\{3\}|}{|\{1,2,3,5\}|} = \frac{1}{4} = 25\%$ of correct leaves. Finally, the reconstructed tree has $A = \frac{7}{10} = 70\%$ correct ancestors, where common ancestry are $\{(4,5), (4,2), (6,1), (4,1), (6,3), (4,3), (4,6)\}$.

*2) IPT vs. Reference IPF:* Sometimes we might be analyzing a set of near-duplicate images without the common source of the images being present in the set. Although the near-duplicates are all related, the tree's root is not available. In this case, we want to compare a reconstructed IPT with its ground truth forest (IPF). For instance, suppose a common source $\mathcal{I}_\mathcal{A}$ has three direct children but such source is not available for analysis. In this case, we need to compare this reconstructed IPT to its original image phylogeny forest. For this intent, we need to make slight changes to the above metrics:

**Root**: $R_F(\text{IPT}, \text{IPF}) = \begin{cases} 1, & \text{If } \texttt{Root(IPT)} \in \texttt{Root(IPF)} \\ 0, & \text{Otherwise} \end{cases}$

**Edges**: $E_F(\text{IPT}, \text{IPF}) = \frac{|E_T \cap E_F|}{|E_F|}$

**Leaves**: $L_F(\text{IPT}, \text{IPF}) = \frac{|L_T \cap L_F|}{|L_T \cup L_F|}$

**Ancestry**: $A_F(\text{IPT}, \text{IPF}) = \frac{|A_T \cap A_F|}{|A_F|}$

According to the above metrics, if the IPT's root is the root of one of the trees in the reference forest, it is correct. If an edge connects two nodes in the IPT and the same is true in the reference forest, it is correct. The same holds for leaves and ancestry information. Figure 4 shows an example of an original image phylogeny tree, its structure after the removal of the original root and some other links giving rise to an image phylogeny forest (IPF), and its reconstructed version.

Considering the *Root$_F$ Evaluation* metric, the reconstructed tree has $R_F$(IPT, IPF) = 1 given that the algorithm found one correct root from the reference forest. According to the *Edges$_F$ Evaluation* metric, the reconstructed tree has $E_F$(IPT, IPF)= $\frac{|\{(5\to2),(6\to2),(8\to3),(12\to3)\}|}{6} = \frac{4}{6} = 66.6\%$ correct edges. In addition, the reconstructed tree has $L_F$(IPT, IPF)= $\frac{|\{7,8,11,12\}|}{|\{5,7,8,11,12\}|} = \frac{4}{5} = 80\%$ of correct leaves. Finally, the reconstructed tree has $A_F$(IPT, IPF)= $\frac{5}{7} = 71.4\%$ correct ancestors, where the common ancestry are $\{(2,5), (2,11), (2,6), (3,8), (3,12)\}$.

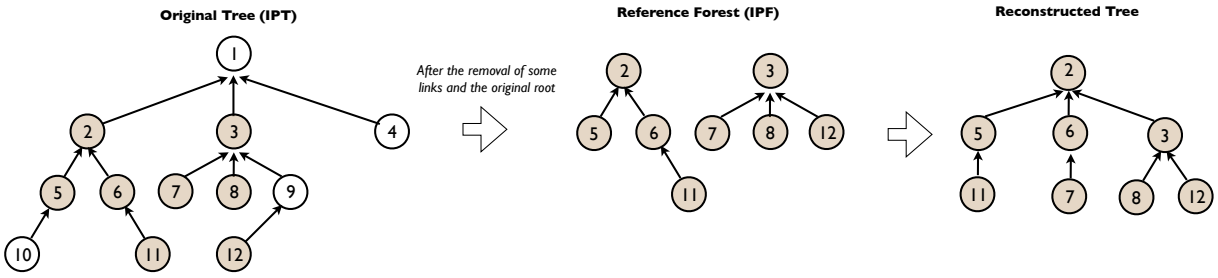
### *B. Evaluation of Results on Uncontrolled Environments*

For uncontrolled environments, we do not have any definitive proof of a parent-child relationship between any two images. Although we do not have the ground-truth for the image relationships within a set of possible near-duplicates $\mathcal{I}$, we are able to define some metrics to evaluate the robustness of an image phylogeny algorithm and its associated input dissimilarity matrix. For that, we can select one image $\mathcal{I}_A \in \mathcal{I}$ and artificially generate a near-duplicate $\mathcal{I}_B = T'_{\vec{\beta}}(\mathcal{I}_A)$ where $T'_{\vec{\beta}} \in \mathcal{T}'$ denotes a family of pre-defined operations $\mathcal{T}$ and respective parameters $\beta$.

We devise four error metrics ($\text{Er}_{i\in\{1,2,3,4\}}$) and one accuracy metric (P) that compare the tree reconstructed from the original set with the tree reconstructed from the augmented set with the new near-duplicate $\mathcal{I}_B$.

- **Er$_1$** is one if the new node $\mathcal{I}_B$ is not a child of its generating node $\mathcal{I}_A$, and zero otherwise.
- **Er$_2$** is one if the structure of the tree relating the nodes in the original set changes with the insertion of the new node $\mathcal{I}_B$ in the set, and zero otherwise.
- **Er$_3$** is one if the new node $\mathcal{I}_B$ appears as a father of another node on the original tree, and zero otherwise.
- **Er$_4$** is one if the root of the reconstructed tree of the original set is different from the root of the reconstructed tree of the set augmented with $\mathcal{I}_B$, and zero otherwise.
- **P** is one if the reconstructed tree is perfect compared to the original tree ($\text{Er}_1$ = $\text{Er}_2$ = 0), and zero otherwise.

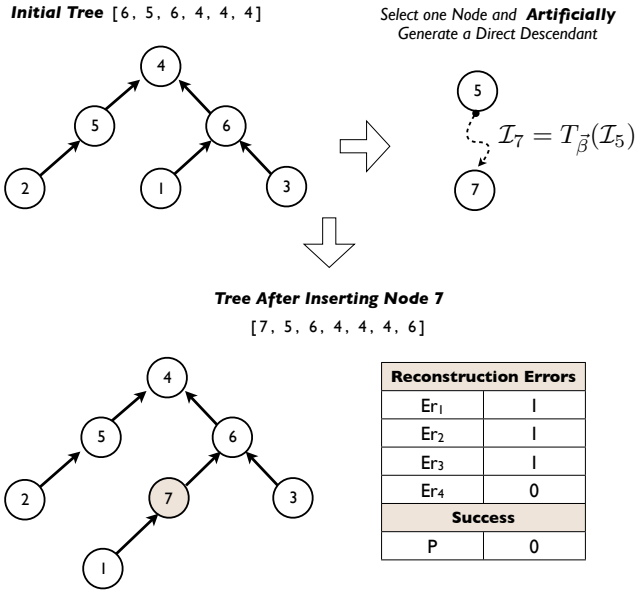
Figure 5 shows one example for $n = 6$ images. In this case, suppose we select the Node 5 to artificially generate an offspring (Node 7). After the insertion of Node 6 to the original tree, suppose it changes according to the bottom part of Figure 5. In this case, we have a hit for errors $\text{Er}_{i\in\{1,2,3\}}$ because Node 7 appears as a child of Node 6 instead of




| Reference Forest (IPF) | Reconstructed Tree (IPT) |
|---|---|
| **Root**<br>Root(IPF) $= \{2, 3\}$ | **Root**<br>Root(IPT) $= 2$ |
| **Edges**<br>$E = \{(5 \to 2), (6 \to 2), (11 \to 6), (7 \to 3), (8 \to 3), (12 \to 3)\}$ | **Edges**<br>$E = \{(5 \to 2), (11 \to 5), (6 \to 2), (7 \to 6), (3 \to 2), (8 \to 3), (12 \to 3)\}$ |
| **Leaves**<br>$L = \{5, 7, 8, 11, 12\}$ | **Leaves**<br>$L = \{7, 8, 11, 12\}$ |
| **Ancestry**<br>$A = \{(2,5), (2,6), (2,11), (6,11), (3,7), (3,8), (3,12)\}$ | **Ancestry**<br>$A = \{(2,5), (2,11), (2,6), (2,7), (2,3), (2,8), (2,12), (5,11), (6,7), (3,8), (3,12)\}$ |

Figure 4. On the top left, an example of a tree encompassing an original image and its transformations to create near-duplicates. On the middle, the image phylogeny forest resulting from the removal of the white nodes of the original tree (left). On the right, the reconstructed phylogeny tree of transformations and near-duplications. On the bottom, a table with details about the data structures necessary to run and evaluate an IPT reconstruction algorithm in the scenario with missing links in which the original root is not present.

Node 5, it changes the structure of the initial tree and it appears as a father of Node 1. However, the new node does not lead to a change in the original tree's root. Finally, since the node does not appear in the correct position and changes the tree's structure, the reconstruction is not perfect ($P = 0$).




| Reconstruction Errors | |
|---|---|
| $Er_1$ | 1 |
| $Er_2$ | 1 |
| $Er_3$ | 1 |
| $Er_4$ | 0 |
| Success | |
| P | 0 |

Figure 5. Process of insertion of an artificial near-duplicate on an IPT reconstructed with an image phylogeny algorithm.

### C. Experiments Setup

We use four experimental scenarios to study the behavior of our IPT(s) construction algorithm:

1) **Controlled Environment with Complete Trees**. In this case, we have a controlled environment in which we know a priori the ground truth of the image relationships. In addition, if we are evaluating the relationship of $n$ near-duplicate iamges, there are no missing links between two given nodes. In this experiment, we assess the quality of the results using the metrics we devised in Section V-A1 and Oriented Kruskal looks for one IPT ($k = 1$).
2) **Controlled Environment with Missing Links**. Similar to the previous experiment except now we can have trees with missing links. It is possible to a parent node $\mathcal{I}_A$ to have a child $\mathcal{I}_B$ and a grand-child $\mathcal{I}_C$ but the link from $\mathcal{I}_A$ to $\mathcal{I}_C$ is missing which means $\mathcal{I}_B$ is not present in the set of available images. With this analysis, we want to verify if the proposed approach preserves the ancestry information with missing pieces of the evolution. We evaluate two possible sub-scenarios here: (1) missing links preserving the root (common source) and (2) missing links in which the root is always missing. Here Oriented Kruskal also looks for one IPT ($k = 1$). We assess the quality of the results using the metrics we devised in Section V-A1 for the first case and the metrics devised in Section V-A2 for the second case.
3) **Behavior on Multiple Sets of Near-duplicate Images**.

The objective here is to verify if our approach is robust enough to find the correct structure of near-duplicate images when there is more than one IPT present using the test cases proposed by De Rosa et al [4]. For reference, we also show De Rosa et al's results [4]. For this experiment, Oriented Kruskal looks for two IPTs ($k = 2$).

4) **Uncontrolled Environment**. Here, do not have any prior knowledge or ground-truth about the image's relationships (they do not necessarily represent an image near-duplicate set). We evaluate the quality of the results using the metrics described in Section V-B. In this case, Oriented Kruskal looks for one IPT ($k = 1$).

### D. Data Sets

*1) Controlled Data Set:* To generate a near-duplicate an image can undergo several possible transformations, however, these transformation must not destroy the overall meaning of the image otherwise it would not be considered a near-duplicate. For the controlled scenario, we select typical transformations an image can undergo such as: JPEG recompression with a different quality factor, isotropic and anisotropic (different scales for horizontal and vertical axis) resampling, contrast adjustment, brightness adjustment, nonlinear gamma correction, rotation, translation, and cropping.

Table I shows the transformations and their operational ranges for creating the controlled data set. The cropping operation already includes translation since it is performed with respect to the image center. We tried to select realistic ranges in order to preserve image semantics while generating near-duplicates.

Table I
TRANSFORMATIONS AND THEIR OPERATIONAL RANGES FOR CREATING THE CONTROLLED DATA SET.

| Transformation | Operational Range |
|---|---|
| **Geometry** | |
| (1) Resampling (Up/Down) | |
| – Global scaling | $[90\%, 110\%]$ |
| (2) Rigid Transformation | |
| – Global scaling | $[90\%, 110\%]$ |
| – Rotation | $[-5°, 5°]$ |
| (3) Generic Affine Transformation | |
| – Scaling by axis | $[90\%, 110\%]$ |
| – Rotation | $[-5°, 5°]$ |
| – Off-diagonal correction | [0.95, 1.05] |
| **Cropping** | |
| (4) Cropping | $[0\%, 5\%]$ |
| **Color** | |
| (5) Brightness Adjustment | $[-10\%, 10\%]$ |
| (6) Contrast Adjustment | $[-10\%, 10\%]$ |
| (7) Gamma Correction | $[0.9, 1.1]$ |
| **Compression** | |
| (8) Re-compression | $[50\%, 100\%]$ |

All of these transformations can be combined in any form to create a near-duplicate. In addition, the color transformations can be performed either linearly or non-linearly across the color channels. All the near-duplicate generation process uses the algorithms implemented in the ImageMagick Library. The pixel interpolation method for resizing uses a Gaussian filter. Brightness and contrast are globally applied to the image – they are converted to offset and slope of a linear transform and then applied using a polynomial function to each channel independently. Finally, Gamma ($\gamma$) adjusts the image's channel values pixel-by-pixel according to a power law, $pixel^{1/\gamma}$ (also applied independently in each channel).

To create the data set, we selected 50 images from the *Uncompressed Color Image Database* (UCID) [35] which contains a wide variety of images with $512 \times 384$-pixel resolution, without compression artifacts. For reproducibility, we used the images with $\texttt{id} = i \times 25$ where $i \in [1..50]$.

For each image, we have created trees with 10, 20, 30, 40, and 50 nodes (near-duplicate images) from 50 different tree topologies. A tree with 50 nodes represents an original image with its forty-nine near-duplicates. For each topology, we created ten different set of parameters also randomly selected from the predefined parameter ranges. The final data set has 25,000 test cases for each tree size. As we select five possible tree sizes, the final data set has 125,000 test cases (50 images, 50 topologies, 10 sets of parameters, 5 different tree sizes).

For the experiments with missing links retaining the root, we selected all trees with 50 nodes (25,000) and randomly created subsets of them with varying size ($[5..50]$ with increments of 5). Therefore, the data set for the experiment with missing links has 250,000 test cases (25,000 trees $\times 10$ missing links setups). For each tree, we randomly select some nodes to remove (preserving the original root) generating a tree with missing links – a tree with some missing connections. Then, we reconstruct the tree with the remaining nodes and evaluate the reconstruction process using the metrics we designed in Section V-A1 to compare trees. Figure 6 illustrates the process.

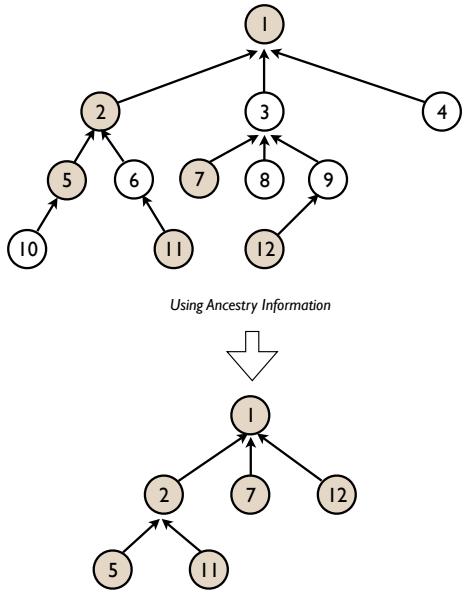



Figure 6. From an initial tree reconstructed with a given image phylogeny algorithm (top), we randomly remove some nodes and generate the tree on the bottom based on the tree's ancestry information. Missing links experienceents consider the ancestry relations on the newly generated tree.

For the experiments with missing links always removing the root, we selected all trees with 50 nodes (25,000) and

randomly created subsets of them with varying size ([5..50] with increments of 5) the same way as before (25,000 trees $\times 10$ missing links setups) except that for each tree, we randomly select some nodes to remove and always remove the original root generating a forest with missing links. The data set for the experiment with missing links without the original root also has 250,000 test cases. To evaluate the results, we use the metrics designed in Section V-A2.

*2) Uncontrolled Data Set:* For the uncontrolled scenario, we crawled 10 target image groups from the internet. An image group is a group of images with the same semantical meaning about which we can not tag as a near-duplicate set given that we have no information about the image relations. Figure 7 depicts the target groups. For instance, the image group related to *Iranian Missiles* refers to 20 images of the iranian missiles case [36]. The total number of images across the 10 target groups is 187.

Since for each target group we create a direct descendant for each image using five variation of parameters (c.f., Tab. I), we have a data set with $187 \times 5 = 935$ images[4].

## VI. Experiments and Results

### A. Controlled Environment with Complete Tree

This section evaluates the image phylogeny approach we introduce in this paper. We evaluate trees of size 10, 20, 30, 40, and 50 nodes. For each size, we have 25,000 possible trees varying the original image, tree topologies, and sets of generation transformation parameters as we discussed in Section V-D.

Figure 8 depicts the summary results for our approach in the controlled environment with complete tree. In the plot, the $x$-axis denotes the number of nodes in the tested tree while the $y$-axis denotes the percentage of correct reconstruction (score) according to the four metrics we devised in Section V-A1 (Root, Edges, Leaves, and Ancestry).

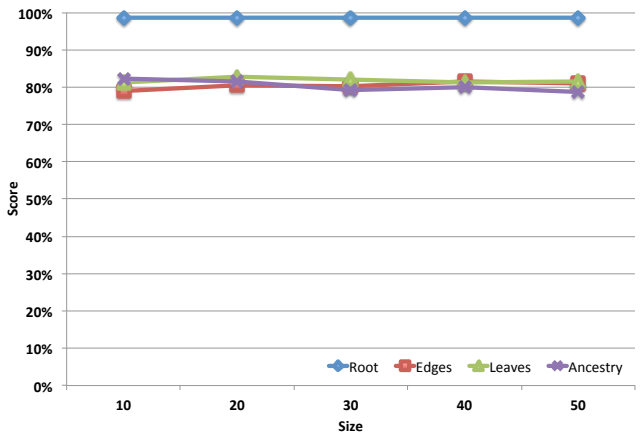



Figure 8. Summary results for the introduced approach for the scenario with no missing links.

Regardless the tree size, the proposed approach correctly finds the root with more than $98.5\%$ accuracy. The same holds for the edges, leaves, and ancestry information in which the proposed approach finds the correct edges, leaves, and ancestries in about $80\%$ of the cases regardless the tree size.

Another interesting form for evaluating the effectiveness of an IPT reconstruction approach is to assess the average depth of the tree in which it finds the correct root (the lower the average depth the better). If an algorithm finds the correct root at depth zero, it means it correctly identified the root of the tree. In this experiment, regardless of the tree size, the average depth at which our solution finds the correct root is lower than 0.03.

If we have several images to analyze, knowing the most probable images to be the root of an image phylogeny tree and having reliable ancestry information are an important result for forensics. With such knowledge we can reduce team effort during investigation, or even design cascade detectors which, at the initial levels, efficiently filter out several nodes. Nodes requiring a more in-depth analysis are then processed in later levels with more discriminative but more computationally intensive, detectors.

### B. Controlled Environment with Missing Links

In this section, we evaluate the effectiveness of the introduced approach with respect to the reconstruction of the image phylogeny tree when some of the links are missing but the the original root is always present and also when we have missing links and original root is always missing.

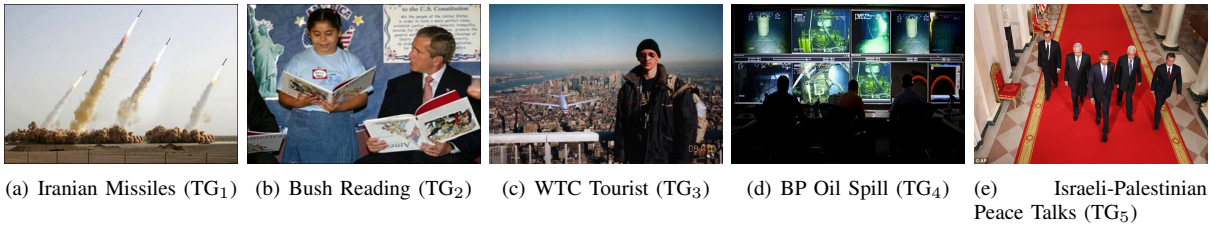
Figure 9 depicts the summary results for the proposed image phylogeny approach in the scenario with missing links retaining the root. In the plot, the $x$-axis denotes the number of nodes remaining in the tested tree after removing $50 - x$ nodes. For instance, $x = 5$ means we are evaluating a tree with 45 missing nodes in the image phylogeny tree, while $x = 50$ is the complete tree. The $y$-axis denotes the percentage of correct reconstruction (score) according to the four metrics we devised in Section V-A (Root, Edges, Leaves, and Ancestry) to compare two IPTs.

For trees with 35 missing nodes ($x = 15$ in the plot), the proposed approach finds the root correctly for $94.2\%$ of the test cases and correctly finds more than $50\%$ of the edges, leaves, and ancestry information. For trees with 20 missing links ($x = 30$ in the plot), the solution correctly finds the root in $96.9\%$ of the cases.

When we compare Figures 8 and 9, we can draw some interesting conclusions for the forensics scenario. First, Oriented Kruskal algorithm is specially good at identifying the root of the trees regardless the number of missing links. When dealing with internet cases, this feature is of particular interest since the lack of information does not hinder the identification process. Second, the algorithm performs reasonably well even in the case where $90\%$ of the tree structure is missing – the Oriented Kruskal algorithm correctly identifies $45.1\%$ of edges, $53.6\%$ of leaves, and $54.7\%$ of the ancestry information when 45 pieces of information out of 50 are missing.

Figure 10 depicts the summary results for the proposed image phylogeny solution in the scenario with missing links always removing the root. In this case, we need to compare a

[4]The necessary information with respect to the used data sets is available in http://www.ic.unicamp.br/∼rocha/pub/communications.html.



(a) Iranian Missiles ($TG_1$) (b) Bush Reading ($TG_2$) (c) WTC Tourist ($TG_3$) (d) BP Oil Spill ($TG_4$) (e) Israeli-Palestinian Peace Talks ($TG_5$)


(f) Criminal Record ($TG_6$) (g) Palin and Rifle ($TG_7$) (h) Beatles Rubber ($TG_8$) (i) Kerry and Fonda ($TG_9$) (j) OJ Simpson ($TG_{10}$)

Figure 7. Examples of the ten target image groups (TG) for the Uncontrolled Scenario. The images are slightly distorted for displaying puporses.

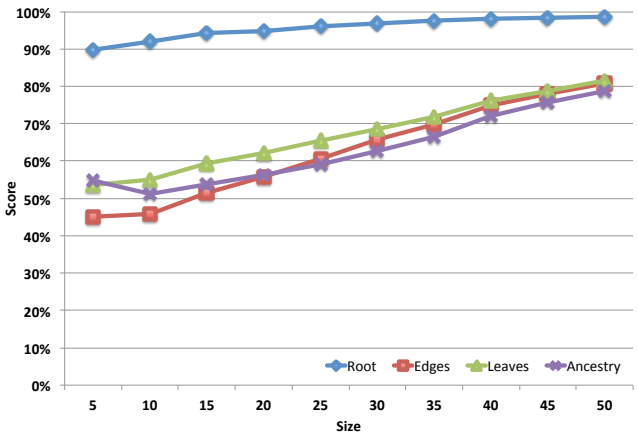



Figure 9. Summary results for the introduced approach for the scenario with missing links preserving the root.

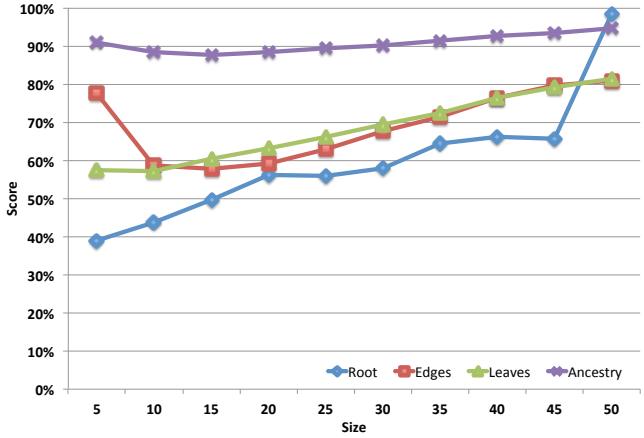



Figure 10. Summary results for introduced approach for the scenario with missing links always removing the root.

reconstructed IPT with an original forest (IPF). For that, we use the four metrics devised in Section V-A2 ($\text{Root}_F$, $\text{Edges}_F$, $\text{Leaves}_F$, and $\text{Ancestry}_F$) to compare IPTs and IPFs.

In this comparison, regardless the number of missing links, the ancestry information with respect to the reference forest correctly found is close to 90%. For instance, in the case with 40 missing nodes including the root ($x = 10$ in the plot), the algorithm correctly finds $88.5\%$ of ancestry information. This strong result shows that even in the case where the original root is missing along with several other nodes, the algorithm is robust enough to analyze which node is related to each other correctly finding their ancestry connections.

Although the root information in this difficult scenario is not as high as when the original root is present, the average depth for finding one of the roots of the original IPF is less than 1.0 regardless the number of missing links, which is still very low. Also, for 35 missing nodes or less, the algorithm finds the correct root in more than 50% of the cases.

The results above considering missing links with the original root also missing, are specially important for forensics analysis where we have to investigate a set of images with similar semantics but we are clueless about their relationships. Even without any information, the algorithm performs reasonably well. Finally, as expected the results improve with less missing information even when the original root is not present.

### C. Behavior on Multiple Sets of Near-duplicate Images

In this section, we present test results for our approach in a scenario where there is more than one distinct set of near-duplicate images. In this case, the goal here is to verify if our approach is robust enough to find the correct structure of near-duplicate images when there is more than one IPT present. In this test, we use the test cases proposed by De Rosa et al. [4] and we also show their results for reference. Such a comparison, however, must not be taken as definitive

but rather as illustrative of the two main methods in image phylogeny literature to date.

In their work [4], De Rosa et al. propose two controlled test cases. Each test case contains two original images and four near-duplicates (descendants) generated through any combination of compression, histogram stretching, rotation, and scaling operations. The first test (defined as *easy* by the authors) has two original images (roots) which depict the same scene with a slight difference in perspective and are acquired with different cameras. The second test case (deemed *hard* by the authors) has two original images depicting the very same visual content and are acquired with the same digital camera.

Figure 11(a) depicts the ground truth for the two trees (both test cases share the same tree's structure). Figures 11(b)-(c) and Table II compare De Rosa et al.'s approach to our proposed solution both visually and in terms of the metrics devised in Section V-A. The ground truth depicted in Figure 11(a) shows there are two roots to be found (Node 1 and Node 2), five leaves (Nodes {5,6,8,9,10}), eight edges ($\{(9 \rightarrow 1), (3 \rightarrow 1), (5 \rightarrow 3), (6 \rightarrow 3), (4 \rightarrow 2), (7 \rightarrow 2), (10 \rightarrow 4), (8 \rightarrow 7)\}$), and 12 ancestors (the direct ancestors given by the edges plus $\{(5,1), (6,1), (10,2), (10,4)\}$).

De Rosa et al.'s algorithm [4] finds 1/2 correct roots in both test cases or 50%. Also, for both test cases, the algorithm finds 7/9 correct edges or 77.8% (it changes edges $(7 \rightarrow 2)$ by $(2 \rightarrow 7)$), 5/5 correct leaves or 100%, and 10/15 correct ancestors or 66.7% (it mixes the ancestors (7,2) and (8,2) with (2,7), (4,7), and (10,7)).

To compare the introduced approach to the one presented by De Rosa et al. [4], we run Oriented Kruskal with $k = 2$. Oriented Kruskal finds 2/2 correct roots in both test cases, i.e., 100%. For Test Case #1, the algorithm finds 7/9 correct edges or 77.8% (it changes edges (10→ 4) by (10→8)), 4/6 leaves or 66.7%, and 11/14 correct ancestors or 78.6% (it mixes the ancestors (10,4), (10,7), and (10,8)). For Test Case #2, the algorithm achieves perfect reconstruction according to all the metrics considered.

When the family of transformations of the dissimilarity function is powerful enough to capture the difference between the different trees, finding the correct $k$ in order to perform the IPTs reconstruction is an outlier detection problem. In the simplest case, a forensics investigator may have some hint about the number of groups in the set of $n$ near-duplicate images he has to analyze and set $k$ accordingly. Another possibility is to investigate clustering algorithms over the dissimilarity matrix to gain some information about groups of images that share relations. Finally, this flexibility is also useful if we want to investigate possible groups of related images changing $k$ progressively and following the merging process of the trees.

In the first test case above, the first processed edge $(4 \rightarrow 2)$ has a dissimilarity value of 0.011, followed by $(8 \rightarrow 7)$ with the same dissimilarity. Then edge $(3 \rightarrow 1)$ is processed with dissimilarity 0.018 and so on until edge $(9 \rightarrow 1)$ is processed with dissimilarity 0.046. At this point, the algorithm has inserted $n = 8$ edges and has found two IPTs. If we wanted only one IPT, the next edge to process would be $(1 \rightarrow 7)$ with dissimilarity value 0.184, which would connect the two IPTs.

Table II
COMPARISON OF DE ROSA ET AL.'S [4] AND ORIENTED KRUSKAL ALGORITHMS FOR IMAGE PHYLOGENY TREE.

| De Rosa et al. [4] | | | | |
|---|---|---|---|---|
| | Root | Edges | Leaves | Ancestry |
| TC#1 | 50.0% | 77.8% | 100% | 66.7% |
| TC#2 | 50.0% | 77.8% | 100% | 66.7% |
| **Oriented Kruskal** | | | | |
| | Root | Edges | Leaves | Ancestry |
| TC#1 | 100.0% | 77.8% | 66.7% | 78.6% |
| TC#2 | 100.0% | 100.0% | 100.0% | 100.0% |

For Test Case #2, the eighth processed edge $(2 \rightarrow 1)$ has a dissimilarity of 0.057 and the ninth edge $(2 \rightarrow 1)$, which would connect the two trees, has a dissimilarity of 0.062. As $k = 2$ the algorithm stops before processing such an edge.

Finally, this experiment corroborates the good results obtained with our approach in the previous sections. Interestingly, our method has a perfect reconstruction for the Test Case #2 which is the most difficult according to De Rosa et al. [4]. In both test cases, the algorithm correctly finds the roots with only a few errors in the ancestry hierarchy for Test Case #1 and none for Test Case #2.

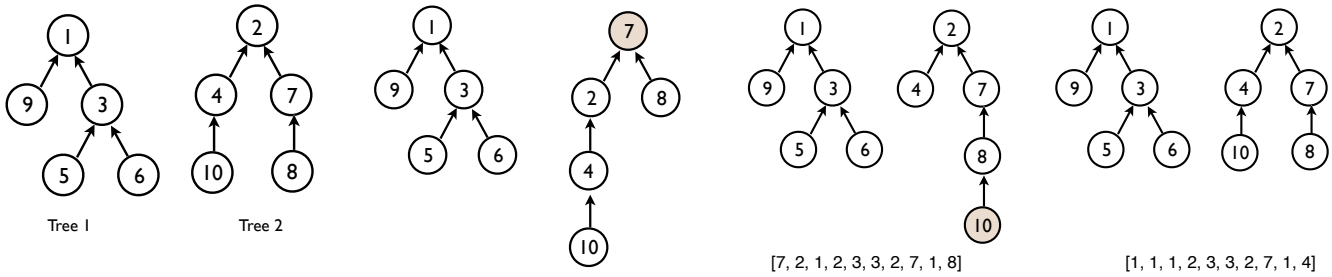
According to De Rosa et al. [4], Test Case #1 is simpler given that it contains two near-duplicate image sets coming from images of different cameras. This seems to be in line with what our algorithm has found. The high dissimilarity value between node 1 and node 7 is the connecting point between the two IPTs if we force the algorithm to yield only one IPT.

### D. Uncontrolled Environment

In this section, we present results for the uncontrolled scenario in which we do not have any prior knowledge about the relationships of a given set of images and use the data set of Section V-D. The data set has 10 target image groups with varying numbers of elements. Recall that each image group contains several images with the same semantics collected from the internet. For this analysis, we use the methodology we discussed in Section V-B in which for each possible image in a target group we artificially generate a direct duplicate and compare the tree before the duplication with the tree after the insertion of such duplicate.

Table III presents the results for the unconstrained scenario using the proposed IPT approach. The results show the method is robust to the insertion of an artificial node. It finds the correct relationship for the artificial node with a perfect reconstruction rate of $P = 72.7\%$ with no further change to the original tree. The best result is achieved with the target groups $\text{TG}_{i\in\{2,7,8\}}$ while the target groups $\text{TG}_{i\in\{1,6\}}$ presented the worse results. Even for the more difficult target groups, the solution achieves a reconstruction rate of $P \geq 54\%$.

In only 0.2% of the cases the introduced approach changes the original root after the insertion of an artificial created node – an important achievement for forensics. This means the introduced IPT algorithm is stable when analyzing sets of related images. In addition, in only $6.8\%$ of the tests, the solution changes the original tree structure ($\text{Er}_2$) or inserts the artificial created node not as a leaf of the tree ($\text{Er}_3$). Finally,




(a) Ground Truth. (b) De Rosa et al. [4]. (c) Oriented Kruskal – Test Cases 1 (left) and 2 (right).

Figure 11. Ground truth and the reconstructed trees for the approach proposed by De Rosa et al. [4] and for Oriented Kruskal. Each test case has two trees with five nodes. De Rosa et al.'s algorithm has the same error for the second tree in both test cases. Highlighted nodes represent mistakes.

Table III
ORIENTED KRUSKAL IPT ALGORITHM RESULTS FOR THE UNCONSTRAINED SCENARIO.

| | **Description** | # of Cases | $\%Er_1$ | $\%Er_2$ | $\%Er_3$ | $\%Er_4$ | %P |
|---|---|---|---|---|---|---|---|
| $TG_1$ | Iranian Missiles | 90 | 40.0% | 11.1% | 11.1% | 0.0% | 55.6% |
| $TG_2$ | Bush Reading | 95 | 17.9% | 3.2% | 3.2% | 0.0% | 81.1% |
| $TG_3$ | WTC Tourist | 95 | 25.3% | 6.3% | 6.3% | 1.1% | 71.6% |
| $TG_4$ | BP Oil Spill | 100 | 25.0% | 0.0% | 0.0% | 0.0% | 75.0% |
| $TG_5$ | Israeli-Palestinian Peace Talks | 95 | 21.1% | 7.4% | 7.4% | 0.0% | 75.8% |
| $TG_6$ | Criminal Record | 90 | 41.1% | 13.3% | 13.3% | 0.0% | 54.4% |
| $TG_7$ | Palin and Rifle | 100 | 17.0% | 2.0% | 2.0% | 0.0% | 81.0% |
| $TG_8$ | Beatles Rubber | 100 | 8.0% | 9.0% | 9.0% | 1.0% | 85.0% |
| $TG_9$ | Kerry and Fonda | 80 | 21.3% | 13.8% | 13.8% | 0.0% | 68.8% |
| $TG_{10}$ | OJ Simpson | 90 | 18.9% | 2.2% | 2.2% | 0.0% | 78.9% |
| | **Average** | 93.5 | 23.5% | 6.8% | 6.8% | 0.2% | 72.7% |

the approach finds the correct parent of the artificial node for about $76.5\%$ of the test cases ($Er_1$=23.5%).

## VII. CONCLUSIONS AND FUTURE WORK

In this paper, we tackled the problem of identifying the image relationships within a set of near-duplicate images, which we named Image Phylogeny Tree.

We presented a principled way to calculate a dissimilarity matrix which measures the relationship between every pair of images in a set of near-duplicate images. For that, we accounted for the most probable transformations an image undergoes when generating an offspring, namely compression, resampling after affine warping and crop, and channel-wise pixel normalization. We also proposed an algorithm to reconstruct image phylogeny trees named Oriented Kruskal.

We validated our approach on both controlled and uncontrolled environments with more than 625,000 test cases. In this sense, this paper presents a significant improvement over our previous work [3] not only in the experimentation but also in the proposed solutions.

There are several immediate applications for our work. In a forensic scenario in which the task is to trace back the original document of a chain of possible modifications, our approach might be of vital importance since it robustly estimate a dissimilarity matrix for a given set of $n$ near-duplicates and effective at finding the root of the near-duplicate's tree. For copyright enforcement, our solution might be interesting to find all the individuals in the tree that actually changed the document. In this task, the proposed approach might be useful since it is able to point out the ancestry information even when there are some missing pieces of information.

Another contribution of our paper is the introduction of quantitative measures for the controlled and uncontrolled validation scenarios. We release a set of scripts to generate the NDDR sets used in this paper, and scripts that calculate the different error metrics of a reconstructed tree. This will allow fair comparisons for the community's future contributions on this relatively new problem.

This paper focuses on the IPT reconstruction from the dissimilarity matrix, but there are still the question on how different families of transformations impact the quality of reconstruction, and wether we should use the image content or the sensor information. Also, a future direction of investigation includes approaches that automatically find the number of trees, $k$, in a set of $n$ images with similar semantics.

## ACKNOWLEDGMENT

We thank the São Paulo Research Foundation (FAPESP), the Brazilian National Council for Scientific and Technological Development (CNPq), and Microsoft for the financial support. We also extend our gratitude to Dinei Florêncio for the invaluable discussions regarding the validation with real world data, as well as to Dr. Michael Eckmann, Dr. Walter J. Scheirer, Valerie Miller, MD, and all the reviewers and area editor for the important feedbacks on early drafts of this work.

## REFERENCES


[1] E. Valle, "Local-descriptor matching for image identification systems," PhD Thesis, Universit de Cergy-Pontoise, Cergy-Pontoise, France, 2008.

[2] A. Joly, O. Buisson, and C. Frélicot, "Content-Based Copy Retrieval Using Distortion-Based Probabilistic Similarity Search," *IEEE Transactions on Multimedia*, vol. 9, no. 2, pp. 293–306, 2007.

[3] Z. Dias, A. Rocha, and S. Goldenstein, "First Steps Toward Image Phylogeny," in *Intl. Workshop of Information Forensics and Security (WIFS)*. IEEE, 2010, pp. 1–6.

[4] A. D. Rosa, F. Uccheddua, A. Costanzo, A. Piva, and M. Barni, "Exploring Image Dependencies: a New Challenge in Image Forensics," in *Media Forensics and Security II*. SPIE, 2010, pp. X1–X12.

[5] L. Kennedy and S.-F. Chang, "Internet Image Archaeology: Automatically Tracing the Manipulation History of Photographs on the Web," in *Intl. Conference on Multimedia*. ACM, 2008, pp. 349–358.

[6] S. Goldenstein and A. Rocha, "High-Profile Forensic Analysis of Images," in *Intl. Conf. on Crime Detection and Prevention*, 2009, pp. 1–6.

[7] B. Lewin, *Genes VI*, 6th ed. Oxford University Press, 1997.

[8] J. Reeds, *John Dee: Interdisciplinary studies in English Renaissance Thought*, 1st ed. Springer, 2006, ch. John Dee and the Magic Tables in the Book of Soyga.

[9] C. Babbage, "Notice respecting some errors common to many tables of logarithms," *Memoirs of the Astronomical Soc.*, vol. 3, pp. 65–67, 1827.

[10] P. Morrison and E. Morrison, *Charles Babbage and his calculating engines*. Dover Publications Inc., 1961.

[11] C. D. McKinsey, *The Encyclopedia of Biblical Errancy*. Prometheus Books, 1995.

[12] N. R. Lightfoot, *How We Got the Bible*. Baker Books, 2010.

[13] A. Rocha, W. Scheirer, T. E. Boult, and S. Goldenstein, "Vision of the Unseen: Current Trends and Challenges in Digital Image and Video Forensics," *ACM Computing Surveys*, In Press 2011.

[14] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganograph*, 2nd ed. Morgan Kaufmann, 2007.

[15] A. Jaimes, S. fu Chang, and A. Loui, "Duplicate detection in consumer photography and news video," in *ACM Multimedia*, 2002, pp. 423–424.

[16] F. Schaffalitzky and A. Zisserman, "Multi-view matching for un-ordered image sets, or how do I organize my holiday snaps?" in *European Conference on Computer Vision*, 2002, pp. 414–431.

[17] D.-Q. Zhang and S. fu Chang, "Detecting image near-duplicate by stochastic attributed relational graph matching with learning," in *ACM Multimedia*, 2004, pp. 877–884.

[18] C.-S. Lu and C.-Y. Hsu, "Geometric distortion-resilient image hashing scheme and its applications on copy detection and authentication," *Multimedia Systems*, vol. 11, no. 2, pp. 159–173, December 2005.

[19] Z. Liu, T. Liu, D. Gibbon, and B. Shahraray, "Effective and Scalable Video Copy Detection," in *ACM Intl. Conference on Multimedia Information Retrieval*, 2010, pp. 119–128.

[20] Z. X. H. Ling, F. Zou, Z. Lu, and P. Li, "Robust Image Copy Detection Using Multi-resolution Histogram," in *ACM Intl. Conference on Multimedia Information Retrieval*, 2010, pp. 129–136.

[21] J. Fridrich, D. Soukal, and J. Lukas, "Detection of Copy-Move Forgery in Digital Images," in *Digital Forensics Research Conf.*, 2003.

[22] M. O'Searcoid, *Metric Spaces*. Springer, 2006.

[23] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proc. of the American Mathematical Society*, vol. 7, no. 1, pp. 48–50, 1956.

[24] L. Brown, "A Survey of Image Registration Techniques." *ACM Computing Survey*, vol. 24, pp. 325–376, 1992.

[25] J. Pluim, J. Maintz, and M. Viergever, "Mutual Information Based Registration of Medical Images: A Survey," *IEEE Transactions on Medical Imaging*, vol. 22, pp. 986–1004, 2003.

[26] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[27] M. Fischler and R. Bolles, "Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography." in *Communications of the ACM*, vol. 24(6), 1981, pp. 381–395.

[28] J. Edmonds, "Optimum Branchings." *J. Research of the National Bureau of Standards*, vol. 71B, pp. 233–240, 1967.

[29] Y. J. Chu and T. H. Liu, "On the Shortest Arborescence of a Directed Graph." *Science Sinica*, vol. 14, pp. 1396–1400, 1965.

[30] R. E. Tarjan, "Finding Optimum Branchings." *Networks*, vol. 7, pp. 309–312, 1977.

[31] P. Camerini, L. Fratta, and F. Maffioli, "A note on finding optimum branchings." *Networks*, vol. 9, pp. 309–312, 1979.

[32] H. N. G. Z. Galil T. Spencer Robert Endre Tarjan, "Efficient algorithms for finding minimum spanning trees in undirected and directed graphs," *Combinatorica*, vol. 6, pp. 109–122, 1986.

[33] M. L. F. Robert Endre Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *Journal of ACM*, vol. 34, pp. 596–615, 1987.

[34] R. E. Tarjan, "Efficiency of a good but not linear set union algorithm," *Journal of the ACM*, vol. 22, no. 2, pp. 215–225, 1975.

[35] G. Schaefer and M. Stich, "UCID – An Uncompressed Colour Image Database," in *SPIE Storage and Retrieval Methods and Applications for Multimedia*, 2004, pp. 472–480, http://www.staff.lboro.ac.uk/~cogs/datasets/UCID/ucid.html.

[36] Mike Nizza and Patrick Witty, "In an iranian image, a missile too many," Available at http://thelede.blogs.nytimes.com/2008/07/10/in-an-iranian-image-a-missile-too-many, The New York Times, July $10^{th}$, 2008.


**Zanoni Dias** received his B.Sc (Computer Science) and Ph.D. (Computer Science) degrees from University of Campinas (Unicamp), Brazil, in 1997 and 2002, respectively. Since 2003, he is an assistant professor in the Institute of Computing, Unicamp, Brazil. His main interests include theoretical computer science, bioinformatics, and computational molecular biology.

**Anderson de Rezende Rocha** received his B.Sc (Computer Science) degree from Federal University of Lavras (UFLA), Brazil in 2003. He received his M.S. and Ph.D. (Computer Science) from University of Campinas (Unicamp), Brazil, in 2006 and 2009, respectively. Currently, he is an assistant professor in the Institute of Computing, Unicamp, Brazil. In 2011, Prof. Rocha was awarded the Microsoft Research Faculty Fellow. His main interests include digital image and video forensics, data hiding, pattern analysis, machine learning, and general computer vision. He is the co-general chair of the *2011 IEEE Intl. Workshop on Information Forensics and Security* (WIFS), and also an Associate Member of the *IEEE Information Forensics and Security Technical Committee* (IFS-TC).

**Siome Goldenstein** received a Ph.D. in Computer and Information Science from University of Pennsylvania in 2002, an M.Sc. in Computer Science from Pontifícia Universidade Catolica do Rio de Janeiro in 1997, and an Electronic Engineering degree from the Federal University of Rio de Janeiro in 1995. He is an Associate Professor at the Institute of Computing, University of Campinas, Unicamp, Brazil and a senior IEEE member. His interests lie in computer vision, computer graphics, forensics, and machine learning. He is an Area Editor of two journals, Computer Vision and Image Understanding (CVIU) and Graphicsl Models (GMOD), has been in the program committee of multiple conferences and workshops, was the local organizer of the 2007 IEEE Intl. Conference on Computer Vision in Brazil, and was co-chair of the 2007 IEEE Workshop on Computer Vision Applications for Developing Regions and co-chair of the 2008 IEEE Workitorial of vision of the unseen. Over the years he had funding from CNPq, CAPES, and FAPESP, and was one of the Principal Investigators of the Project Harpia, a multimillion dollar contract for the Brazilian Customs.