

©2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Pre-print of article that will appear in T-IP, vol.27, no.12, pp.6109-6123, Dec. 2018.

The published article is available on <https://doi.org/10.1109/TIP.2018.2865674>

# Image Provenance Analysis at Scale

Daniel Moreira, Aparna Bharati, *Student Member, IEEE*, Joel Brogan, *Student Member, IEEE*, Allan Pinto, *Student Member, IEEE*, Michael Parowski, Kevin W. Bowyer, *Fellow, IEEE*, Patrick J. Flynn, *Fellow, IEEE*, Anderson Rocha, *Senior Member, IEEE*, and Walter J. Scheirer, *Senior Member, IEEE*

**Abstract**—Prior art has shown it is possible to estimate, through image processing and computer vision techniques, the types and parameters of transformations that have been applied to the content of individual images to obtain new images. Given a large corpus of images and a query image, an interesting further step is to retrieve the set of original images whose content is present in the query image, as well as the detailed sequences of transformations that yield the query image given the original images. This is a problem that recently has received the name of image provenance analysis. In these times of public media manipulation (*e.g.*, fake news and meme sharing), obtaining the history of image transformations is relevant for fact checking and authorship verification, among many other applications. This article presents an end-to-end processing pipeline for image provenance analysis, which works at real-world scale. It employs a cutting-edge image filtering solution that is custom-tailored for the problem at hand, as well as novel techniques for obtaining the provenance graph that expresses how the images, as nodes, are ancestrally connected. A comprehensive set of experiments for each stage of the pipeline is provided, comparing the proposed solution with state-of-the-art results, employing previously published datasets. In addition, this work introduces a new dataset of real-world provenance cases from the social media site *Reddit*, along with baseline results.

**Index Terms**—Digital Image Forensics, Digital Humanities, Image Retrieval, Graphs, Image Provenance, Image Phylogeny

## I. INTRODUCTION

Algorithms for the detection of manipulated content in digital images have reached a stage of maturity that is sufficient for understanding the transformations that were applied to individual images in many cases [1]–[3]. A logical next step is to develop an approach that allows us to ask more complicated questions about the relationships between related images after sequences of transformations have been applied — a problem that is not well studied in the image processing literature. In this article, we consider the *Provenance Analysis* task [4], [5], in which the objective is to recover the graph of relationships between plausibly connected images. These relationships may be expressed as undirected edges (*i.e.*, neighboring transformations are identified) or directed edges (*i.e.*, the order of neighboring transformations is expressed). The development of techniques to recover such graphs combines ideas from the areas of image retrieval, digital image forensics, and graph

D. Moreira, A. Bharati, J. Brogan, M. Parowski, K. Bowyer, P. Flynn, and W. Scheirer are with the Department of Computer Science and Engineering, University of Notre Dame, IN, USA. A. Pinto and A. Rocha are with the Institute of Computing, University of Campinas, SP, Brazil.

This paper has supplementary material available at <http://ieeexplore.ieee.org>, provided by the authors. The material includes an explanatory video. Corresponding author: W. J. Scheirer ([walter.scheirer@nd.edu](mailto:walter.scheirer@nd.edu)).

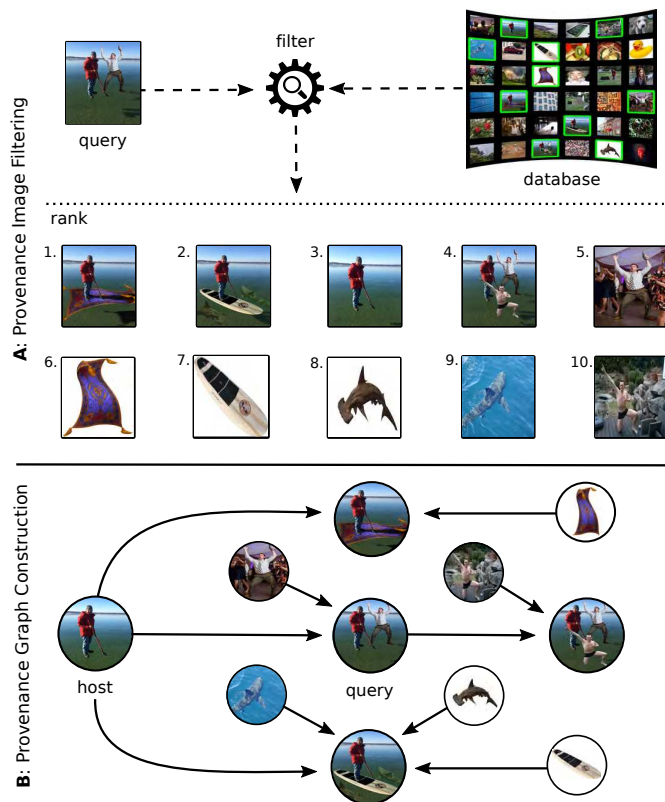


Fig. 1. Image Provenance Analysis workflow. Panel A depicts the first step of Image Provenance Analysis, namely Provenance Image Filtering, in which filters are applied to a large image database to retrieve those images that are related to a given query image. Panel B depicts the second step, namely Provenance Graph Construction, in which the filtered images are linked to each other in a way that expresses the sequences of manipulation and/or compositions (*i.e.*, the provenance history of the images).

theory, making this an interesting interdisciplinary endeavour within image processing and computer vision.

To illustrate the provenance analysis task, consider the set of example images in Panel A of Fig. 1, which were collected from the popular “Photoshop battles” forum on the social media site *Reddit* [6]. On this forum, amateur artists begin with source images and employ image manipulation tools to generate results for humorous effect. The first step in provenance analysis is *Provenance Image Filtering*, which consists of searching a potentially large pool of images for those that are most closely related to a given query image. Related images might be *semantically similar* (*i.e.*, the same scene may be present from slightly different view points or at nearby points in time), or they might be *near duplicates* related by minor transformations such as exposure and saturation

adjustments, or cropping and re-sizing, or they might be *image compositions*, which contain elements of two or more different source images. In most cases, the query will be an image that has been manipulated in some way.

The second step is *Provenance Graph Construction*, where the objective is to understand the relationships between images yielded by provenance image filtering. A *Host Image* provides the source of background content for subsequent manipulations. In Fig. 1, the host is the photo of the man holding a shovel in the leftmost part of Panel B. A *Donor Image* provides some amount of content that will be inserted into a host image. In Fig. 1, three donor images are the original images of the sharks and the paddle board in the bottom half of Panel B. They provide image content that has been inserted into the image they are linked to. Sequences of manipulations are common, and they can be expressed as a directed graph representing the order in which they were applied. This can be seen in the graph of Panel B, where the depth of the central path containing the host and the query leads to three different levels of manipulations. Our goal is to develop an algorithm that can generate such graphs in an automated fashion. We do not make strong assumptions that either the original host or donor images are available during analysis. For instance, the paddle board, flying carpet, and extra people might not necessarily be harvested at the image filtering step.

Provenance analysis is important to image processing and computer vision. It has direct applications in a number of different fields. The most immediate application is forensics, where the detection of manipulated images spans traditional policing to analysis for strategic intelligence. The question of the origins of suspect images has taken a prominent role recently, with the rise of so-called “fake news” on the Internet. While not a new problem<sup>1</sup>, concern about fake news reached new heights on the heels of the 2016 American presidential election. The rapid evolution of the online social media landscape has provided new, free media channels with which even amateur bloggers and news outlets can reach massive audiences with little effort, and even less regulation. Recent instances of fake news often involve questionable images propagating through social media. For example, in early 2017, the New York Times reported on the creation of a false story about the discovery of pre-marked ballots in Ohio that appeared a couple of months before the election [8]. The image accompanying the story was the product of a mirrored image that was selectively blacked-out in local regions [9]. This is a real-life case with multiple manipulations where provenance analysis could be applied to trace the origins of the fabrication.

Beyond the important application domain of forensics, image provenance analysis can form a powerful framework for academic research in other fields. Cultural analytics has emerged as a distinct sub-discipline within the digital humanities [10], [11] that is concerned with combining quantitative methods from social science and computer science to answer humanistic questions about cultural trends. An example of this (which we have already touched upon in Fig. 1) is the study of

Internet *memes* — cultural artifacts meant to be widely transmitted and evolve over time. Memes are an interesting object of cultural study, in that they encapsulate facets of popular entertainment, political moods, and novel elements of humor. Meme aggregators like the website *knowyourmeme.com* have done a good job at archiving such content, but a more exhaustive quantitative study of the provenance of individual memes has yet to emerge. Tracing the source(s) of modified meme images helps us unpack the underlying cultural trends that can tell us something meaningful about the community that generated the content.

Both of the application domains mentioned also motivate the need for any developed techniques to be *scalable*. Specialized algorithmic components are necessary to solve the problem at hand. First, one needs an accurate and scalable image retrieval algorithm that is able to operate over very large collections of images (realistically, on the order of millions of images) to find related candidates. Such an algorithm also has to address the particularities of the provenance image filtering task: it must perform well at retrieving the near-duplicate host images that are highly related to the query (a well-known problem in the image retrieval literature), but also perform well at retrieving donors (images that potentially donated small portions to the query) and the donors’ respective near duplicates (which might not be directly related to the query). Second, the identification of likely image transformations that explain how each retrieved image might have been used to generate the others is required, as it is used to create the ordering of the images in the provenance graph. And third, methods from graph theory are necessary to organize the relationships between images, yielding a directed graph that is human-interpretable. All of these components must be integrated as a coherent and scalable processing pipeline.

In this context, this work introduces, for the first time, a fully automated large-scale end-to-end pipeline that starts with the step of provenance image filtering (over millions of images) and ends up with the provenance graphs.

## II. RELATED WORK

*Content-based image retrieval (CBIR)*. In recent years, research advances in the domain of CBIR have included optimizing the memory footprint of indexing techniques and employing graphical processing units (GPU) for parallel search. A recent technique proposed by Johnson et al. [12] utilizes state-of-the-art image indexing (Optimized Product Quantization (OPQ) [13]) and runtime optimization to perform similarity search on the order of a billion images. Such approaches can be directly applied to perform image filtering for provenance analysis. However, as they follow the traditional CBIR inverted-file index pipeline [14], they will not generalize to all cases due to the nature of the problem. While regular CBIR will probably retrieve good host candidates to the query, in the face of compositions (which are fairly common in provenance analysis), small donors will not be highly ranked (or will not even be retrieved) without adaptations to the base approach.

The work of Pinto et al. [15] improves the retrieval of donors related to a query in the scope of provenance analysis.

<sup>1</sup>The computer hacker group Cult of the Dead Cow warned of the devastating potential of widespread online media manipulation as early as 1999 [7].

The paper introduces a two-tiered search approach. The first tier constitutes a typical CBIR pipeline, while the second tier provides a context-aware query-masking technique, which selects the regions from the query that make it divergent from hosts previously obtained in the first tier. With such regions as evidence, a second search is performed, this time avoiding hosts and retrieving additional potential donor images. Although such an approach does improve the retrieval of donors, it adopts a very “query-centric” point of view with respect to the problem of provenance analysis. It only finds the hosts and donors that directly share content with the query, ignoring the other descendants and the ancestors of such hosts and donors, which are transitively related to the query.

*Image processing for image associations.* In our proposed workflow, the filtering step yields relevant images, and then provenance graph construction is performed. The provenance graph construction step involves finding diverse types of associations among images based on their similarities and/or dissimilarities. For that reason, it is related to tasks such as visual object recognition [16], scene recognition [17], place recognition [18], object tracking [19], near-duplicate detection [20], and image phylogeny [4], since they all rely on the comparison of two or more images.

Some visual association tasks may be general, as they relate images based on the common characteristics that optimally make them related. This is the case, for instance, for object recognition. For example, a query image that implicitly requests “retrieve all the images containing dogs” may also be assumed to be generalized (any breed, color, or size). Scene recognition (e.g., “retrieve all the images depicting bedrooms”) may also include generalized queries. In such situations, a high content diversity among the related images is usually desired [21]. By contrast, some image association tasks may be specialized, in the sense that they aim at extracting the specific characteristics that aid in the visual identification of a sample in a particular setting. That is the case of place recognition (e.g., retrieve all the images of Times Square), and object tracking (e.g., segment the target vehicle plate across the frames of a street surveillance video).

Techniques for associating images in a general way include comparing global image representations [22], [23], employing bags of visual features [24], [25] and using convolutional neural networks (CNNs) [26]–[29]. Techniques for associating images in a specialized way include assessing local feature matching [30]–[34], image patch matching [35], and evaluating the quality of image registration, color matching, and mutual information [36], [37]. Particularly, provenance analysis is by definition closer to the specialized tasks; for that reason, in this work, we benefit more from techniques mentioned in the latter group.

Although one can adapt deep CNNs to provenance analysis by optimizing them for specialization rather than generalization at training time [38], [39], such a procedure is — at the present time — only accomplishable at the expense of prohibitive training times, the need for a reasonably large cluster of GPUs for model screening via hyperparameter optimization, and a sufficiently large amount of available training data [40]. In addition, making such a solution perform at scale at infer-

ence time is also challenging. Zagoruyko and Komodakis [38] report that their CNN approach performs feature extraction on high-end GPUs twice as slow as SIFT [41] running on a regular CPU. In this work, we have intentionally chosen to pursue faster alternatives.

*Image phylogeny trees.* Provenance analysis is related to the simpler task of image phylogeny, which seeks to recover a tree of relationships. Kennedy and Chang [42] were the first to point out the possibility of relying on the color information of pixels and local features for gathering clues about plausible parent-child relationships among images. Based upon the pixel colors and local features, they suggest detecting a closed set of directed manipulations between pairs of content-related images (namely copy, scaling, color change, cropping, content insertion, and overlay detection).

Rather than exhaustively modeling all of the possible manipulations between near-duplicate images, Dias et al. [43] suggest having a good dissimilarity function that can be used for building a pairwise image dissimilarity matrix  $D$ . Accordingly, they introduce *oriented Kruskal*, an algorithm that processes  $D$  to output an *image phylogeny tree*, a data structure that expresses the probable evolution of the near duplicates at hand. In subsequent work, Dias et al. [4] formally present the dissimilarity-calculation protocol that is widely used in the related literature for computing  $D$ . They then go on to conduct a large set of experiments with this methodology, considering a family of six possible transformations, namely scaling, cropping, affine warping, brightness, contrast, and lossy content compression [44]. Finally, in [5], Dias et al. replace oriented Kruskal with other phylogeny tree building methods: best Prim, oriented Prim, and Edmonds’ optimum branching [45], with the last solution consistently yielding improved results.

*Image phylogeny forests.* The image phylogeny solutions mentioned up to this point were conceived to handle near duplicates; they do not work in the presence of semantically similar images. Aware of such limitations, Dias et al. [46] extend the oriented Kruskal solution to *automatic oriented Kruskal*, an algorithm that finds a family of disjoint phylogeny trees (a phylogeny forest) from a given set of near duplicates and semantically similar images, such that each tree describes the relationships of a particular group of near duplicates. Analogously, Costa et al. [47] provide two extensions to the optimum branching algorithm, namely *automatic optimum branching* and *extended automatic optimum branching*, both based on automatically calculated cut-off points. Alternatively, Oikawa et al. [48] propose the use of clustering techniques for finding the various phylogeny trees; the idea is to group images coming from the same source, while placing semantically similar images in different clusters. Finally, Costa et al. [36] improve the creation of the dissimilarity matrices, regardless of the graph algorithm used for constructing the trees.

*Multiple parenting phylogeny trees.* Although previous phylogeny work established preliminary analysis strategies and algorithms to understand the evolution of images, the key scenario of image composition, in which objects from one image are spliced into another, was not addressed. Compositions were first addressed within the phylogeny context by

Oliveira et al. [49]. The solution presented by these authors assumes two parents (one host and one donor) per composite. Extended automatic optimum branching is thus applied for the construction of ideally three phylogeny trees: one for the near duplicates of the host, one for the near duplicates of the donor, and one for the near duplicates of the composite. Even though this work is very relevant to ours herein, it has a couple of limitations. First, it does not consider the possibility of more than two images donating content towards one composite image (such as the composite with sharks in Panel B of Fig. 1). Second, Oliveira et al. require all images to be in JPEG format.

*Provenance graphs.* To date, the entire image phylogeny literature has made use of metrics that focus on finding the root of the tree, rather than evaluating the phylogeny tree as a whole, considering every image transformation path in the case of provenance. Aware of such limitations and aiming to foster more research on the topic, the American National Institute of Standards and Technology (NIST) has recently introduced new terminology, metrics, and datasets, coining the term *image provenance* to express a broader notion of image phylogeny, and suggesting directed acyclic *provenance graphs*, instead of trees, as the data structure that describes the provenance of images [50]. They also suggest the use of a *query* as the starting point for provenance analysis.

Following this, Bharati et al. [37] introduced a more generalized method of provenance graph construction, which does not assume anything about the images and transformations. A content-based method for the construction of undirected provenance graphs is proposed, which relies upon the extraction and geometrically-consistent matching of interest points. Utilizing this information to build the dissimilarity matrix, the method uses Kruskal’s algorithm to obtain the provenance graph. The approach performs well over small cases, even in the presence of distractors (*i.e.*, images that are not related to the query).

Extending the field of provenance analysis, this paper provides the following contributions:

- 1) *Distributed interest point selection*: a novel interest point selection strategy that aims at spatially diversifying the image regions used for indexing within the provenance image filtering task.
- 2) *Iterative filtering*: a novel querying strategy that iteratively retrieves images that are directly or transitively related to the query, considering all possible hosts, donors, composites, and their respective near duplicates.
- 3) *Clustered provenance graph construction*: a novel graph construction algorithm that clusters images according to their content (joining near duplicates into the same clusters), prior to establishing their intra- and inter-cluster directed relationship maps.
- 4) State-of-the-art results on the provenance analysis benchmark released by NIST [51].
- 5) A new dataset of real-world scenarios containing composite images from Photoshop battles held on the Reddit website [6]. Experiments performed over this dataset highlight the real-world applicability of the approach.

### III. PROVENANCE ANALYSIS METHODOLOGY

As described in Sec. I, the task of image provenance analysis is divided into two major steps, namely *Provenance Image Filtering* and *Provenance Graph Construction*. Fig. 2 depicts an overview of the proposed solution in this context.

#### A. Provenance Image Filtering

The problem of image filtering for the provenance task is different from the typical image retrieval task: a given query image may fulfill one or both of the following conditions:

- The query may have a relationship to various *near duplicates*. The near duplicates may be *hosts* of the query (in the case of the query being a composite that inherits the background from a near duplicate) or the query itself may be a host, as in the case of the query donating a background to the near duplicates.
- The query may be a composite with a relationship to one or more donors, whose content may be entirely disjoint. Donors can even be composites themselves, with their own hosts and donors.

In such scenarios, the retrieval method must return as many of the directly and transitively related images as possible. These aspects define a unique image retrieval and filtering problem, known as *Provenance Image Filtering* [15], [50], which is different from more typical *near-duplicate* or *semantically similar* image retrieval. In this work, we assume that a ground-up system must be deployed for search, retrieval, and filtering, instead of relying on currently available resources such as *Google* [52] or *TinEye* [53].

1) *Distributed Interest Point Selection*: Due to the nature of the manipulations seen in tampered images, it is important to build a filtering system that is tolerant to a wide range of image transformations. Hence, we adopt a low-level image representation that is based on interest points and local features, since they are reportedly tolerant to transformations such as scaling, rotation, and contrast adjustment [54]. Nevertheless, while regular interest points are mostly designed to identify corners and blobs on the image, we also want to describe and further index homogeneous areas with low response and consequently a sparse amount of detected interest points, for retrieving images with the same type of content. Although one can use a dense sampling approach to extract interest points within those regions, this is computationally prohibitive in the context of searching millions of images [15].

Therefore, we introduce a new method called *distributed interest point selection* that aims at keeping a sparse approach while being able to provide interest points inside low-response areas. For that, we extend Hessian-based detectors (such as Speeded-Up Robust Features (SURF) [54]) in the following way. Instead of employing a threshold  $t$  to collect interest points whose local Hessian values are greater than  $t$ , we define a parameter  $p$  that expresses the fixed amount of interest points we want to extract from each target image. Within these  $p$  interest points,  $m < p$  interest points are extracted for the reason of being the top- $m$  regions with the  $m$  strongest Hessian values. The remaining  $n = p - m$  are extracted

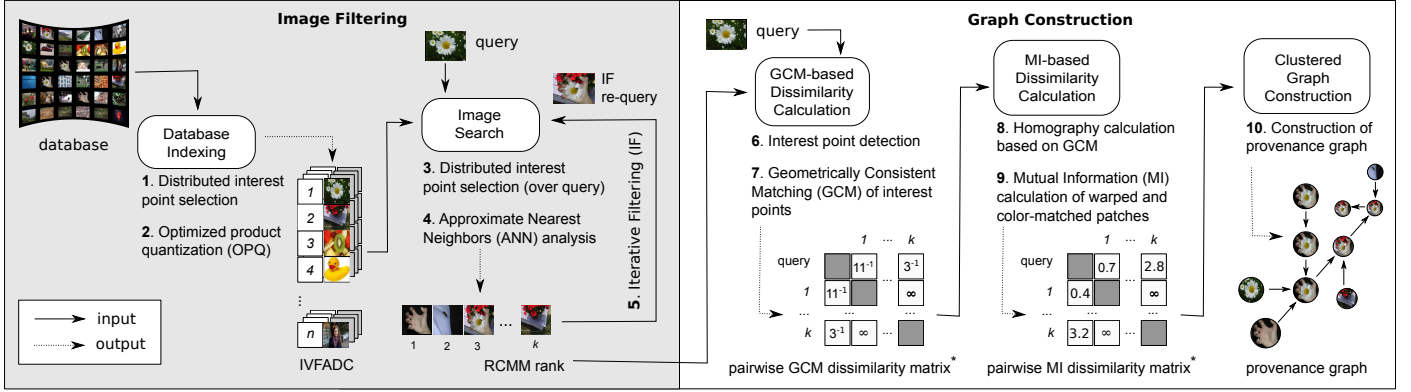


Fig. 2. Proposed pipeline for end-to-end provenance analysis. The sequence of activities is divided into two parts, which address the tasks of *image filtering* (left panel) and of *graph construction* (right panel). IVFADC stands for *Inverted File System with Asymmetric Distance Computation*. RCMM stands for *Reciprocal Condition Matching Measure*. The value of  $n$ , within IVFADC, is the number of images in the database. The value of  $k$  is a parameter of the solution and is related to the size of the RCMM rank used for building the provenance graph. \*Reported values are merely illustrative.

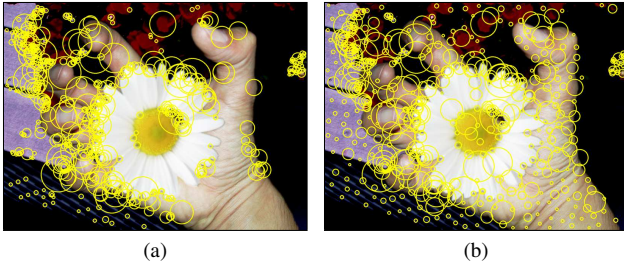


Fig. 3. Effects of using the approach of distributed interest point selection. In (a), the result of a regular SURF interest point detection. In (b), the result of the distributed approach over the same image, with many more points over homogeneous regions, such as the skin of the wrist.

from the set containing the post-top- $m$  interest points, which is also sorted according to the Hessian response. Starting from the  $(m + 1)$ -th strongest interest point, we only add the current interest point if it does not *overlap* with another already selected interest point; otherwise, we try to add the next strongest interest point, up to the point of obtaining  $n$  interest points.

Fig. 3 depicts the effect of using the distributed approach along with SURF. Fig. 3 (a) depicts a regular SURF detection, while Fig. 3 (b) depicts the distributed version, over the same image. Fig. 3 (b) presents more points over the skin of the wrist and background (which are more homogeneous regions) than Fig. 3 (a).

2) *Database Indexing*: The next step is to build the image index structure. After interest point detection and feature extraction, we are left with  $p$  description vectors per image. For an image collection  $C$ :

$$C_i \in C, \quad s.t. \quad i \in \mathcal{I}_{img} = \{0, 1, \dots, |C|\}, \quad (1)$$

our subsequent feature collection  $F$  is:

$$F_i \in F, \quad s.t. \quad i \in \mathcal{I}_{ind} = \{0, 1, \dots, |C| \times p\}, \quad (2)$$

where  $\mathcal{I}_{img}$  denotes the numbered index set of full images within  $C$ , and  $\mathcal{I}_{ind}$  indicates the subsequent numbered index

set assigned to individual features in  $F$ . We transform  $F$  to a new space using *Optimized Product Quantization* (OPQ) [13] to make the feature space well-posed for coarse *Product Quantization* (PQ). We refer to this new rotated feature set as  $F_r$ . From a random sample of  $F_r$ , a coarse set of representative centroids  $O$  is generated using PQ. A subsequent *Inverted File System with Asymmetric Distance Computation* (IVFADC) [55]) is generated from  $O$ , allowing for fast and efficient search.

3) *Image Search*: Once the database images are indexed, a search procedure can be performed via feature-wise queries. For a query image  $Q$ , a set of  $p$  distributed SURF features  $F_d$  is extracted, rotated according to  $F_r$ , and submitted to the system. Each image  $Q$  leads to a matrix of indices of *Approximate Nearest Neighbors* (ANN)  $R$  of size  $|F_d| \times K$ , where  $K$  is the parameter of the  $K$ -nearest neighbors for the system to return. Each  $R_{ij}$  value of  $R$  is computed using *Asymmetric Distance Computation* (ADC) [55]:

$$R_{ij} = \mathcal{S}(F_{d_i})_j, \quad s.t. \quad i \in \{1, \dots, |F_d|\} \text{ and } j \in \{1, \dots, K\} \quad (3)$$

where  $\mathcal{S}$  signifies a feature-wise query search on the IVFADC filtering system,  $i$  denotes the  $i$ -th query feature of  $Q$ , and  $j$  denotes the  $j$ -th ANN index of the  $j$ -th nearest feature belonging to  $F_r$ . As a consequence, each row of  $R$  corresponds to a feature in  $F_d$ , containing the  $K$ -nearest neighbor features, which we may consider as matched features. We then map  $R$  from the  $\mathcal{I}_{ind}$  space to the  $\mathcal{I}_{img}$  space. The set of unique related image indices is computed as:

$$R_{img} = \mathcal{M}(R, \mathcal{I}_{ind}, \mathcal{I}_{img}). \quad (4)$$

where function  $\mathcal{M}(R, \mathcal{I}_1, \mathcal{I}_2)$  maps index values in  $R$  from the  $\mathcal{I}_{ind}$  domain to the  $\mathcal{I}_{img}$  index domain, allowing each  $R_{ij}$  to represent the image it belongs to. Once  $R_{img}$  is obtained, a result set of images sorted by the number of matched features is calculated for giving the final global query results of  $Q$ .

The score for each image indexed within  $R_{img}$  is therefore denoted by  $freq(x, R_{img})$ , where  $x \in \{R_{img}\}$  and  $freq(x, \cdot)$  is a function that counts the number of occurrences of  $x$ .

4) *Iterative Filtering*: Once a first rank of images is retrieved through the search algorithm, we iteratively refine the results to add images that are not directly related to the query, but are still related in some way to its provenance.

In contrast to the approach described by Pinto et al. [15], which employs a two-tiered search to retrieve the small donors of the query after masking the regions that diverge between the query and the first images of the retrieved rank, in this work we employ the *reciprocal condition matching measure* (RCMM) proposed in [56] to identify and suppress the near duplicates of the query. Given that a large RCMM value between two arbitrary images indicates that they are probably near duplicates, we suppress the retrieved images whose RCMM values with the query are large. The non-suppressed (and therefore non-near-duplicate) images of the current rank are then provided as new queries to the next search iteration, which is performed using the same method explained in Sec. III-A3.

By applying the above process for a number of iterations, we search various sets of non-near-duplicate queries (which are potentially donors) and end up with a set of ranks, which are then flattened and re-ranked using RCMM. In the end, we obtain a less query-centric rank of images, which contains not only images directly related to the query, but also transitively related (e.g., ancestors of the donors of the query). As will be demonstrated in Sec. V, such a strategy improves the recall of the provenance image filtering task.

## B. Provenance Graph Construction

As one can observe in Fig. 2, the provenance graph construction task builds upon the image rank that is obtained by the provenance image filtering task, and ends up with the provenance graph. Therefore, at this point, we can assume that (in the best scenario) all images directly and transitively related to the query are available for constructing the provenance graph, as well as some *distractors* (images that should not be present in the provenance graph, because they are not related to any of the images within it).

The presence of distractors at this step is more of a matter of design. Taking into consideration that, in [37], experiments show distractors not impacting the provenance graph construction too much, and aiming to keep the provenance image filtering part as simple as possible, we give the subsequent dissimilarity matrix calculation task the duty of removing distractors. Therefore, the input is a set containing the  $k$  top-retrieved images and the query, which are then used for building dissimilarity matrices.

1) *Calculation of Dissimilarity Matrices*: Similar to [4], given the set  $I$  containing the  $k$  top-retrieved images and the query, a dissimilarity matrix  $D$  is a  $(k+1) \times (k+1)$  matrix whose elements  $d_{ij}$  describe the dissimilarity between images  $I_i$  and  $I_j$ , respectively the  $i$ -th and  $j$ -th images of  $I$ . Depending on how the values  $d_{ij}$  are calculated,  $D$  can be either symmetric or asymmetric.

In this work, following the solution proposed in [37], we neither make any strong assumptions with respect to the transformations that might have been used to generate the elements of  $I$ , nor impose limitations on the presence of

near duplicates, semantically similar images, or multi-donor composites. Instead, we focus on analyzing the shared visual content between every pair of images  $(I_i, I_j)$  through two ways of calculating  $d_{ij}$ . In the first one, we set  $d_{ij}$  as the inverse of the number of geometrically-consistent interest-point matches (GCM) between images  $I_i$  and  $I_j$ ; in this particular case, the matrix  $D$  is symmetric. In the second one, we set  $d_{ij}$  as the *mutual information* (MI) between a color transformation  $T_j(I_i)$  of image  $I_i$  towards image  $I_j$ ; in this case, the matrix  $D$  is asymmetric. Both methods are described below.

*GCM-based dissimilarity*: Provenance graph construction starts with the detection of interest points over each one of the  $k+1$  images that belong to  $I$ . At this step, different interest point detectors can be applied, such as SURF [54] or Maximally Stable Extremal Regions (MSER) [57], with each one yielding a particular dissimilarity matrix. Once the interest points are available and properly described through feature vectors (e.g., SURF features [54]), we find correspondences among them for every pair of images  $(I_i, I_j)$ . Let  $P_i$  be the set of feature vectors obtained from the interest points of image  $I_i$ , and  $P_j$  be the set of features obtained from  $I_j$ . For each feature belonging to  $P_i$ , the two best matching features are found inside  $P_j$  using Euclidean distance (the closer the features, the better the match). Inspired by Nearest-Neighbor-Distance-Ratio (NNDR) matching quality [41], we ignore all the features whose ratio of the distances to the first and to the second best matching features is smaller than a threshold  $t$ , since they might present a poor distinctive quality. The remaining features are then kept and finally matched to their closest pair.

Even with the use of NNDR, it is not uncommon to gather *geometrically inconsistent matches*, i.e., contradictory interest-point matches that, if together, cannot represent plausible content transformations of image  $I_i$  towards image  $I_j$ , and vice-versa. To get rid of these matches, we adopt a solution that is able to build a geometrically-consistent model of expected interest-point positions from any pair of matches between images  $I_i$  and  $I_j$ . For example, consider two arbitrary matches  $m_1$  and  $m_2$ , which respectively connect points  $p_1 \in I_i$  and  $q_1 \in I_j$ , and points  $p_2 \in I_i$  and  $q_2 \in I_j$ . Based upon the positions, the distance  $l_p$ , and the angle  $\alpha_p$  between points  $p_1$  and  $p_2$  (both from image  $I_i$ ), as well as upon the positions, the distance  $l_q$ , and angle  $\alpha_q$  between points  $q_1$  and  $q_2$  (both from image  $I_j$ ), we estimate the scale, translation, and rotation matrices that make  $p_1$  and  $p_2$  respectively coincide with  $q_1$  and  $q_2$ . With these matrices, we transform every matched interest point of  $I_i$  onto the space of  $I_j$ . As one might expect, points that do not coincide with their respective peers after the transformations have their matches removed from the set of geometrically consistent matches.

Finally, we compute the dissimilarity matrix  $D$  by setting every one of its  $d_{ij}$  elements as the inverse of the number of found geometrically-consistent matches between images  $I_i$  and  $I_j$ . In this case, the dissimilarity matrix is symmetric.

*MI-based dissimilarity*: The mutual-information (MI)-based dissimilarity matrix is an extension of the GCM-based alter-

native (see Fig. 2). After finding the geometrically consistent interest-point matches for each pair of images  $(I_i, I_j)$ , the obtained interest points are used for estimating the homography  $H_{ij}$  that guides the registration of image  $I_i$  onto image  $I_j$ , as well as the homography  $H_{ji}$  that analogously guides the registration of image  $I_j$  onto image  $I_i$ .

In the particular case of  $H_{ij}$ , for calculating  $d_{ij}$ , after obtaining the transformation  $T_j(I_i)$  of image  $I_i$  towards  $I_j$ ,  $T_j(I_i)$  and  $I_j$  are properly registered, with  $T_j(I_i)$  presenting the same size of  $I_j$ , and the matched interest points relying on the same position. We thus compute the bounding boxes that enclose all the matched interest points, within each image, obtaining two correspondent patches  $R_1$ , within  $T_j(I_i)$ , and  $R_2$ , within  $I_j$ . As in [37], the distribution of the pixel values of  $R_1$  is matched to the distribution of  $R_2$ , prior to calculating the pixel-wise amount of residual between them with MI.

From the point of view of information theory, MI is the amount of information that one random variable contains about another. From the point of view of probability theory, it measures the statistical dependence of two random variables. In practical terms, assuming each random variable as respectively the aligned and color-corrected patches  $R_1$  and  $R_2$ , the value of MI is given by the entropy of discrete random variables:

$$MI(R_1, R_2) = \sum_{x \in R_1} \sum_{y \in R_2} p(x, y) \log \left( \frac{p(x, y)}{\sum_x p(x, y) \sum_y p(x, y)} \right), \quad (5)$$

where  $x \in [0, \dots, 255]$  refers to the pixel values of  $R_1$ , and  $y \in [0, \dots, 255]$  refers to the pixel values of  $R_2$ . The  $p(x, y)$  value regards the joint probability distribution function of  $R_1$  and  $R_2$ . As explained in [36], it can be approximated by:

$$p(x, y) = \frac{h(x, y)}{\sum_{x, y} h(x, y)}, \quad (6)$$

where  $h(x, y)$  is the joint histogram that counts the number of occurrences for each possible value of the pair  $(x, y)$ , evaluated on the corresponding pixels for both patches  $R_1$  and  $R_2$ . As a consequence, MI is directly proportional to the similarity of the two patches.

Back to  $H_{ji}$ , it is calculated in an analogous way of  $H_{ij}$ . However, instead of  $T_j(I_i)$ ,  $T_i(I_j)$  is manipulated for transforming  $I_j$  towards  $I_i$ . Further, the size of the registered images, the format of the matched patches, and the matched color distributions are different, leading to a different value of MI for setting  $d_{ji}$ . As a consequence, the resulting dissimilarity matrix  $D$  is asymmetric, since  $d_{ji} \neq d_{ij}$ .

*Avoiding distractors:* As we have mentioned before, the image rank given to the provenance graph construction step may contain distractors, which need to be removed during the dissimilarity matrix calculation step. When computing the dissimilarity matrix  $D$ , the solution proposed by Bharati et al. [37] establishes matches between every pair of available images, including distractors. By interpreting  $D$  as the adjacency matrix of a multi-graph whose nodes are the images, they identify distractors as the nodes weakly connected (*i.e.*, that present a small number of matches, down to none) to

the minimum spanning tree that contains the query. Assuming  $(k+1)$  as the number of image nodes, they perform  $(k^2+k)/2$  operations to populate  $D$ .

In this work, we improve that process by the means of an iterative approach, which starts from the node of the query and then computes the geometrically consistent matches with the remaining  $k$  images. A set with only the strongly connected nodes is thus saved for the next iteration. In the following iterations, the algorithm keeps trying to establish matches starting from the last set of strongly matched images, up to the point where no more strong matches are found.

Although simple, this solution may provide a significant improvement in the runtime of the dissimilarity matrix calculation. Let  $d \leq k$  be the amount of distractors inside the image rank. We avoid  $(d^2 - d)/2$  operations by applying the iterative solution. In the case of a rank with 50 images ( $k = 50$ ), for instance, and 40 distractors ( $d = 40$ ) (indicating that the provenance graph contains only ten images), the number of operations is reduced from 1,275 to 795, significantly speeding up the runtime in case of small graphs.

*2) Clustered Provenance Graph Construction:* Once the GCM- and MI-based dissimilarity matrices are available, we rely on both for constructing the final provenance graph, by the means of a novel algorithm, named *clustered provenance graph expansion*. This algorithm, differently from Kruskal's algorithm, delivers directed provenance graphs rather than undirected minimum spanning trees.

The main idea behind such solution is to group the available images in a way that only near duplicates of a common image are added to the same cluster. Starting from the image query  $I_q$ , the remaining images are sorted according to the number of geometrically consistent matches shared with  $I_q$ , from the largest to the smallest. The solution then clusters probable near duplicates around  $I_q$ , as long as they share *enough content*, which is decided based upon the number of matches. After automatically adding the first image of the sorted set to the cluster of  $I_q$ , the solution iteratively analyzes the remaining available images. For deciding if the  $i$ -th candidate image  $I_i$  (where  $i > 1$ ) is a near duplicate, the algorithm keeps track of the number of matches  $m_i$  between  $I_i$  and the last image  $I_{i-1}$  added to the cluster. Let  $\mu_{i-1}$  be the average number of matches of the cluster, and  $\sigma_{i-1}$  be the standard deviation.  $I_i$  is connected to  $I_{i-1}$  in the final provenance graph, if  $m_i \in [\mu_{i-1} - \sigma_{i-1}; \mu_{i-1} + \sigma_{i-1}]$ ; in such a case,  $I_i$  is added to the cluster by affinity, and novel values of  $\mu_i$  and  $\sigma_i$  are calculated, for evaluating the next candidate  $I_{i+1}$ . Otherwise, the current cluster is considered finished up to  $I_{i-1}$ .

As a consequence, the obtained clusters have their images sequentially connected into a single path, without branches. That makes sense in scenarios involving sequential image edits where one near duplicate is obtained on top of the other, as in [51]. To determine the direction of a single edge, we rely on the mutual information. Let  $D$  be the MI-based dissimilarity matrix, and consider two images  $I_i$  and  $I_j$ , whose respective  $D$  elements are  $d_{ij}$  and  $d_{ji}$ . As explained in [49], an observation of  $d_{ij} > d_{ji}$  means that  $I_i$  probably generated  $I_j$ .

Finally, whenever a cluster is finished and there are still



disconnected available images, we find the image already added to the provenance graph whose number of matches with the remaining ones is largest. This image is then assumed as the new query  $I'_q$ , over which the aforementioned clustering algorithm is executed, considering only the yet disconnected images. As a result, the final provenance graph sees a branch rising from  $I'_q$  as an orthogonal path containing new images.

#### IV. EXPERIMENTAL SETUP

Here we describe the experimental setup, including the datasets (Sec. IV-A), metrics (Sec. IV-B), large-scale infrastructure (Sec. IV-C), and the parametric values employed for image filtering (Sec. IV-D) and graph construction (Sec. IV-E).

##### A. Datasets

1) *NIST Dataset*: As a part of the *Nimble Challenge 2017* [51], NIST released a dataset specifically curated for the tasks of provenance image filtering and graph construction. The dataset is divided into *development* and *evaluation* partitions, of which, at the current time, only the development partition contains fully released ground-truth. For that reason, we rely on that development partition, named *NC2017-Dev1-Beta4*, to evaluate the performance of the proposed solution. Nonetheless, the reader may be interested in learning about our performance on the official challenge. In the released results [58], the herein proposed solution is labeled as *ND-PURDUE* and yields the best official results reported to date.

*NC2017-Dev1-Beta4* contains 65 queries and 11,040 images that comprise samples related to the queries and distractors. As a consequence, the dataset makes available a complete groundtruth that is composed of the 65 expected image ranks as well as the 65 expected provenance graphs related to each query. The provenance graphs were manually created and include images resulting from a wide range of transformations, such as splicing, removal, cropping, scaling, rotation, translation and color correction.

Aiming to enlarge *NC2017-Dev1-Beta4* towards a more realistic scenario, we extend its set of distractors by adding nearly one million images randomly sampled from the *Nimble NC2017-Eval-Ver1* evaluation dataset [51]. The *NC2017-Eval-Ver1* dataset is the latest NIST evaluation set for measuring the performance of diverse image-manipulation detection tasks. However, as aforementioned, no complete provenance ground truth is available for this set, leading us to use *NC2017-Dev1-Beta4* in conjunction with *NC2017-Eval-Ver1*. As a result, we end up with what we call the *NIST dataset*, which comprises the 65 provenance graphs from *NC2017-Dev1-Beta4* and more than one million distractors from both datasets.

Following NIST suggestions in [51], we perform both *end-to-end* and *oracle-filter* provenance analysis over the NIST dataset. On the one hand, the end-to-end analysis includes performing the provenance image filtering task first, and then submitting the obtained image rank to the provenance graph construction step. On the other hand, the oracle-filter analysis focuses on the provenance graph construction task; it assumes that a perfect image filtering solution is available. Therefore, only the graph construction step is evaluated.

2) *Professional Dataset*: Oliveira et. al [49] introduced a multiple-parent phylogeny dataset, which comprises composite forgeries that always have two direct ancestors, namely the *host* (which is used for defining the background of the composite) and the *donor* (which they call *alien* and that donates a local portion, such as an object or person, to define the foreground of the composite). Each phylogeny case comprises 75 images, of which three represent the composite, the host, and the donor, and the remaining 72 represent transformations (*e.g.*, cropping, rotation, scale, and color transformations) over those three images. As a consequence, each case is a provenance graph composed of three independent phylogeny trees (one for the host, one for the donor, and one for the composite) that are connected through the composite and its direct parents (the host and the donor, as expected).

Although our approach is not directly comparable to the one of Oliveira et al. [49] (since they used different metrics and addressed a different problem of finding the correct original images — the graph sources — rather than the quality of the coverage of the complete provenance graph) we make use of their dataset for the reason of the composites being the work of a professional artist that tried to make the images as credible as possible. Therefore, we are assessing the metrics defined in Sec. IV-B and reporting the results over the 80 test cases found within the dataset. In order to adapt it to our provenance graph building pipeline, however, we are choosing a random image inside the provenance graph as a query for each one of the 80 experimental cases. Finally, we do not extend the professional dataset with distractors; hence we perform only oracle-filter analysis over it.

3) *Reddit Dataset*: To supplement the experimental data with even more realistic examples, we have collected a new provenance dataset from image content posted to the online Reddit community known as *Photoshop battles* [6]. This community provides a medium for professional and amateur image manipulators to experiment with image doctoring in an environment of friendly competition. Each “battle” begins with a single root image submitted by a user. Subsequent users then post different modifications, usually humorous, of the image as comments to the original post. Due to the competitive nature of the community, many image manipulations build off one another, as users try to outdo each other for comic effect. This results in manipulation provenance trees with both wide and deep chains. We use the underlying comment structure of these battles to automatically infer the ground truth provenance graph structure, as shown in Fig. 4.

Because these images are real examples of incremental manipulations, the Reddit dataset accurately represents manipulations and operations performed on images in the wild. To assure the quality of the samples, each graph was manually reviewed. Cases with missing images within the comments or disconnected nodes were automatically discarded. In total, the Reddit dataset contains 184 provenance graphs, which together sum up to 10,421 original and composite images. It will be made available to the public upon the publication of this work. Similar to the Professional dataset, we are not extending the Reddit dataset with distractors; we perform only oracle-filter

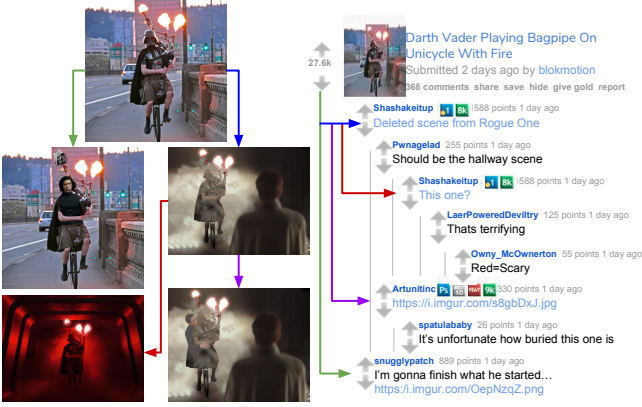


Fig. 4. A visualization of how provenance graphs are automatically inferred from a Reddit Photoshop battle instance. The parent-child behavior of comments (right) can be leveraged to infer the structure of the ground truth provenance graph (left). The colors of each comment correspond to their respective edge in the graph.

analysis over it.

### B. Evaluation Metrics

In this work, we adopt the metrics proposed by NIST in [51] for both the provenance image filtering and graph construction tasks. In the case of provenance image filtering, we report (for each image query) the CBIR recall of the expected images at three particular cut-off ranks:  $R@50$  (the recall considering the top-50 images of the retrieved image rank),  $R@100$  (recall for the top-100 images), and  $R@200$  (recall for the top-200 images). Given that recall expresses the percentage of relevant images that are being effectively retrieved, the solution delivering higher recall is considered preferable.

In the case of provenance graph construction, we assess, for each provenance graph that is computed for each query, the  $F_1$ -measure (*i.e.*, the harmonic mean of precision and recall) of the *retrieved nodes* and of the *retrieved edges* (called *vertex overlap (VO)* and *edge overlap (EO)*, respectively). Additionally, we report the *vertex and edge overlap (VEO)*, which is the  $F_1$ -measure of retrieving both nodes and edges, simultaneously [59]. The aim of using such metrics is to assess the overlap between the groundtruth and the constructed provenance graph. The higher the values of  $VO$ ,  $EO$ , and  $VEO$ , the better the quality of the solution.

Finally, in the particular case of  $EO$  (and consequently  $VEO$ ), we report the overlap both for directed edges (which are assumed to be the regular situation, and therefore kept for  $EO$  and  $VEO$ ), and for undirected edges (when an edge is considered to overlap another one if they connect analogous pairs of nodes, in spite of their orientations). All aforementioned metrics are assessed through the NIST *MediScore* tool [50].

### C. Large-Scale Infrastructure

Fig. 5 shows the proposed full pipeline for index *training* and *construction* (previously explained in Sec. III-A2). Index training refers to the process of learning the OPQ rotations

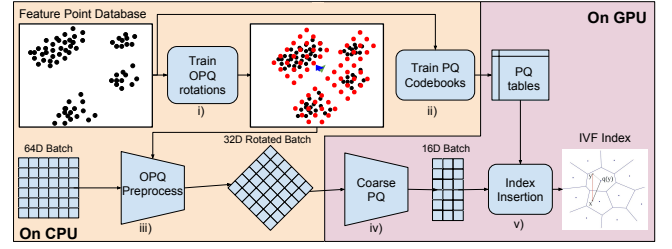


Fig. 5. Filtering pipeline infrastructure. The orange area (left) shows computations that are performed on a CPU. The purple area (right) shows the index ingestion steps that are performed on a GPU.

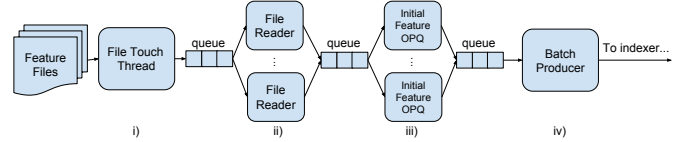


Fig. 6. Producer-consumer index ingestion. Each file contains features for an image. These file locations are pre-loaded into cache via a rate-limited “touch” thread, and are read on a producer-consumer multi-threaded basis.

and PQ codebooks from a sampling of the local features that are extracted from the target dataset. We propose to perform such step ahead of time with typical central processing units (CPU). Index construction, in turn, refers to the computation of the inverted file indices, after properly rotating the previously extracted local features. The learning of OPQ rotations and PQ codebooks can be done in advance on a CPU, but the construction of indices is well suited to the capabilities of graphical processing units (GPU), allowing for faster computation.

Besides employing GPUs to efficiently build and search an index of over 1 million high-resolution images, additional steps must be taken to increase the pipeline speed. To date, most indexing algorithms require single large files containing all features to be ingested at once [60], [61], either due to implementation choices or algorithm limitations. The operation of concatenating all features from a set of images into a single file is prohibitively time consuming when dealing with more than a few million interest points. Because our scenarios require the ingestion of multiple billions of interest points, a different solution must be adopted, in order to avoid the need for file concatenation. For that, we propose a multi-threaded producer-consumer setup, as shown in Fig. 6. In our pipeline, we provide a single feature file per image. The pipeline begins with the “touch” thread, which systematically loads image feature file locations into the computer’s file system cache, for faster retrieval in later stages. Then, a reading thread takes touched files and loads them into memory. A third thread takes sets of loaded feature files and produces feature batches of size  $B$  that are optimized in size for GPU ingestion. The fourth thread applies the initial OPQ pre-processing rotations to the feature set, before sending the final batch to the GPU. Using this method, we are able to process billions of features from high-resolution image datasets orders of magnitudes faster than previous methods.

#### D. Filtering Setup

In all provenance filtering experiments, we either start describing the images with regular 64-dimensional SURF [54] interest points, or the distributed approach explained in Sec. III-A1 combined with the SURF detector (namely *DSURF*). For the regular SURF detector, depending on the experiment, we either extract the top-2,000 most responsive interest points (namely *SURF2k*), or the top-5,000 most responsive ones (*SURF5k*). *DSURF*, in turn, is always described with 5,000 64-dimensional interest points, of which 2,500 regard the top-2,500 most responsive ones, and the remaining 2,500 are obtained avoiding overlap, as explained in Sec. III-A1.

For the sake of comparison, besides reporting results of the IVFADC system (explained in Sec. III-A2), we also report results of the KD-Forest system discussed by Pinto et al. [15] over the same set of images. Because the work in [15] is not easily scalable beyond 2,000 interest points, with respect to memory footprint, we combine it with SURF2k only (namely *KDF-SURF2k*).

Focusing on the IVFADC approach, we provide combinations of it with all the available low-level descriptor approaches, hence obtaining *IVFADC-SURF2k* (for comparison with *KDF-SURF2k*), *IVFADC-SURF5k*, and *IVFADC-DSURF*. Regardless of the descriptors, we are always performing IVFADC with a codebook set size of 32 codes and sub-codebook set size of 96; both values were learned from preliminary experiments as revealing an acceptable trade-off between index building time and size, and final system recall. Finally, aiming at evaluating the impact of using iterative filtering (explained in Sec. III-A4), we evaluate variations of the two most robust filtering solutions (namely *IVFADC-SURF5k* and *IVFADC-DSURF*) by adding iterative filtering (IF), hence obtaining the *IVFADC-SURF5k-IF* and *IVFADC-DSURF-IF* variations. All filtering methods are tested over the NIST dataset, for each one of its 65 queries.

#### E. Graph Construction Setup

As explained in Sec. III-B, the graph construction task always starts with a given query and its respective rank of potentially related images. For computing both the GCM-based and MI-based dissimilarity matrices (all explained in Sec. III-B1), we either detect and match the top-5,000 most responsive SURF interest points per image, for each image pair, or the top-5,000 largest MSER regions per image, again for each image pair. As a consequence, we have available four types of dissimilarity matrices, namely GCM-SURF and GCM-MSER (both symmetric), and MI-SURF and MI-MSER (both asymmetric).

The reason for choosing SURF and MSER is related to their potential complementarity: while SURF detects blobs of interest [54], MSER detects the stable complex image regions that are tolerant to various perspective transformations [57]. Thus, the two methods end up delivering very different sets of interest points. For extracting feature vectors from both SURF and MSER detected interest points, we compute the 64-dimensional SURF features proposed in [54]. In the particular

case of MSER, we compute the SURF features over the minimum enclosing circles that contain each one of the detected MSER image regions. During the GCM feature matching, we match only interest points of the same type (*i.e.*, we match SURF blobs with only SURF blobs, as well as MSER regions with only MSER regions).

In the end, we construct the provenance graphs from the four types of dissimilarity matrices using either Kruskal’s algorithm over the symmetric GCM-based instances (therefore obtaining undirected graphs), or the herein proposed clustered provenance graph expansion approach, which relies on both GCM- and MI-based instances (obtaining directed graphs). As a result, we have available, for performing experiments, the following options: *Kruskal-SURF*, obtained from GCM-SURF adjacency matrices, *Kruskal-MSER*, obtained from GCM-MSER adjacency matrices, *Cluster-SURF*, obtained from both GCM- and MI-SURF matrices, and *Cluster-MSER*, obtained from both GCM- and MI-MSER matrices.

All graph construction methods are tested over the NIST (65 queries), Professional (80 queries), and Reddit (184 queries) datasets. In the particular case of the NIST dataset, we report both end-to-end and oracle-filter analyses. Regarding end-to-end analysis, we start with the best top-100 image ranks that were obtained in the former set of provenance image filtering experiments. As expected, these ranks contain distractors, as well as miss some images related to the query that should be part of the final provenance graph. With respect to the oracle-filter analysis, the ranks will only contain images related to the query.

## V. RESULTS

In this section, we report the experimental results concerning the tasks of provenance image filtering (in Sec. V-A) and of provenance graph construction (in Sec. V-B).

### A. Image Filtering

Table I contains the results of provenance image filtering over the 65 queries of the NIST dataset, following the setup detailed in Sec. IV-D. The best solution is *IVFADC-DSURF-IF*, which reaches an R@50 value of 0.907, meaning that, if we use the respective top-50 rank as input to the provenance task, an average of 90.7% of the images directly and indirectly related to the query will be available for graph construction.

As one might observe, the IVFADC-based solutions presented better recall values when compared to *KDF-SURF2k*, even when the same number of interest points was used for describing the images of the dataset. That is the case, for instance, of the use of *IVFADC-SURF2k*, which provided an increase of approximately 17% in R@50 over its *KDF*-based counterpart (*KDF-SURF2k*), with an insignificant increase in the average query time (from 0.15 to 0.17 minutes). *IVFADC* makes use of CBIR state-of-the-art OPQ, which appears to be more effective than KD-trees for indexing image content.

In addition, the GPU-amenable scalability provided by *IVFADC* allowed us to increase the number of 64-dimensional SURF interest points from 2,000 to 5,000 features per image (reaching around five billion feature vectors for the entire

TABLE I  
RESULTS OF PROVENANCE IMAGE FILTERING OVER THE NIST DATASET.  
WE REPORT THE AVERAGE VALUES ON THE PROVIDED 65 QUERIES.

Solution	R@50	R@100	R@200	Query time (min)
KDF-SURF2k [15]	0.609	0.633	0.649	0.15
IVFADC-SURF2k	0.713	0.722	0.738	0.17
IVFADC-SURF5k	0.876	0.881	0.883	0.55
IVFADC-DSURF	0.882	0.895	0.899	0.54
IVFADC-SURF5k-IF	0.895	0.901	0.919	2.53
<b>IVFADC-DSURF-IF</b>	<b>0.907</b>	<b>0.912</b>	<b>0.923</b>	<b>2.20</b>

In bold, the solution with highest recall values.

dataset). With more interest points, the dataset is better described, leading, for example, to an increase of nearly 23% in R@50 for IVFADC-SURF2k over IVFADC-SURF5k. That, however, happens at the expense of slowing down the solution more than three times, from 0.17 to 0.55 minutes of average query time.

The use of DSURF also increased the recall values. Its application was responsible for an improvement of almost 7% in R@50, when we compare IVFADC-SURF5k and IVFADC-DSURF, at the expense of adding one more hour to the time required to construct the index for the entire dataset. This extra hour is related to the additional step of avoiding interest point overlaps, which is part of the DSURF detection solution. Nevertheless, the average query time remains practically unchanged (0.55 and 0.54 minutes, respectively), indicating that the cost of avoiding overlaps is balanced by a description space that generates a dataset index of better quality.

Finally, the use of IF made the recall values approach 0.9, even considering R@50. For example, the use of IVFADC-DSURF-IF yielded an improvement of nearly 3% in R@50 over IVFADC-DSURF. That happened, however, at the expense of a significant increase of search time, due to the iterative re-querying nature of IF; IVFADC-DSURF-IF lasts four times longer than IVFADC-DSURF (respectively, 0.54 and 2.20 minutes). However, in certain scenarios where time is not a constraint, the increase of 3% in recall may justify the deployment of such an approach.

All results reported in this article were obtained with 24 CPU cores operating at 2.4 GHz and eight Nvidia TITAN Xp GPUs.

### B. Graph Construction

We organize the results of graph construction according to the adopted dataset (either NIST, Professional, or Reddit). Table II shows the performance of the proposed approach over the NIST dataset. Results are grouped into end-to-end and oracle-filter analysis. In the particular case of end-to-end analysis, top-100 rank lists were obtained with IVFADC-DSURF-IF filtering, the best approach reported in Table I. As a consequence, the respective provenance graphs are built,

TABLE II  
RESULTS OF PROVENANCE GRAPH CONSTRUCTION OVER THE NIST DATASET. WE REPORT THE AVERAGE VALUES ON THE PROVIDED 65 QUERIES.

	Solution	VO	EO	VEO
End-to-end analysis	Kruskal-SURF [37]	0.638	0.429 <sup>†</sup>	0.537 <sup>†</sup>
	Kruskal-MSER	0.257	0.140 <sup>†</sup>	0.199 <sup>†</sup>
	<b>Cluster-SURF</b>	<b>0.853</b>	<b>0.353</b>	<b>0.613</b>
	Cluster-MSER	0.835	0.312	0.585
Oracle-filter analysis	<b>Kruskal-SURF [37]</b>	<b>0.933</b>	<b>0.256<sup>†</sup></b>	<b>0.609<sup>†</sup></b>
	Kruskal-MSER	0.902	0.239 <sup>†</sup>	0.585 <sup>†</sup>
	Cluster-SURF	0.931	0.124	0.546
	Cluster-MSER	0.892	0.123	0.525

<sup>†</sup>: Values for undirected edges. In bold, the solutions with the best VEO.

TABLE III  
RESULTS OF PROVENANCE GRAPH CONSTRUCTION OVER THE PROFESSIONAL DATASET. WE REPORT THE AVERAGE VALUES ON THE 80 QUERIES BELONGING TO THE TEST SET.

Solution	VO	EO	VEO
<b>Kruskal-SURF [37]</b>	<b>0.985</b>	<b>0.218<sup>†</sup></b>	<b>0.604<sup>†</sup></b>
Kruskal-MSER	0.663	0.087 <sup>†</sup>	0.377 <sup>†</sup>
Cluster-SURF	0.975	0.102	0.541
Cluster-MSER	0.604	0.043	0.326

<sup>†</sup>: Values for undirected edges. In bold, the solution with the best VEO.

on average, without almost 9% of the image nodes, which are not retrieved in the filtering step ( $R@100 = 0.912$ , in the case of IVFADC-DSURF-IF). Oracle-filter analysis, in turn, starts from a perfect rank of images, containing all and only the graph image nodes. That explains the higher values of VO in such group, at the expense of reducing EO. The reduction of EO is explained by the availability of more related images in the step of graph construction, which increases the number of possible edges and misconnections. It means that the present solutions are good at removing distractors, but there is still room to improve the effective connection of sharing-content images. The best end-to-end solution is Cluster-SURF, retrieving, on average, directed provenance graphs with 0.613 ground truth-graph coverage (VEO). The best oracle-filter solution, in turn, is Kruskal-SURF, with 0.609 undirected graph coverage.

In Table III, we present results of the proposed approaches on the Professional dataset. In comparison to the NIST dataset, the same solutions recognize fewer of the correct provenance graph edges. This happens due to the larger 75-node provenance graphs, which contain a number of near duplicates that were created through reversible operations. As a consequence, altered image nodes can be achieved using different sequences

TABLE IV  
RESULTS OF PROVENANCE GRAPH CONSTRUCTION OVER THE REDDIT DATASET. WE REPORT THE AVERAGE VALUES ON THE PROVIDED 184 QUERIES.

Solution	VO	EO	VEO
Kruskal-SURF [37]	0.884	0.156 <sup>†</sup>	0.523 <sup>†</sup>
<b>Kruskal-MSER</b>	<b>0.924</b>	<b>0.121<sup>†</sup></b>	<b>0.526<sup>†</sup></b>
Cluster-SURF	0.757	0.037	0.401
Cluster-MSER	0.509	0.027	0.271

<sup>†</sup>: Values for undirected edges. In bold, the solution with the best VEO.

of image transformations, leading to ambiguous dissimilarity values, and multiple plausible paths, within the provenance graph. The methods herein discussed are solely based on image content and do not consider any extra information, thus operating with data from only the pixel domain. Indeed, in previous image phylogeny work reporting results on the Professional dataset, the solutions made use of data from the JPEG compression tables of the images. We speculate that if information regarding the compression factor is included in the present approaches, some confusion regarding the edges can be eliminated. That would not impact the NIST dataset, though, since only a small fraction of its images are available in JPEG format. Here, the best solution is Kruskal-SURF, retrieving, on average, undirected provenance graphs with 0.604 ground truth-graph coverage (VEO).

Table IV reports results on the Reddit dataset. As one might observe, this dataset is the most challenging one, with low directed edge coverage (namely *EO*, in the case of Cluster-SURF and Cluster-MSER solutions). Since its whimsical content is the product of a diverse community, the Reddit dataset presents realistic, yet frustratingly complex, cases. As a consequence, it is not uncommon to find among the 184 collected provenance graphs suppressed ancestral images, as well as descendant images whose parental connections are defined by very particular and contextual semantic reasons (for instance, an arbitrary person resembling another in the parent image), than by strictly shared visual content. That ends up impacting our results. The best solution is Kruskal-MSER, which retrieves, on average, undirected provenance graphs with 0.526 ground truth-graph coverage (VEO).

With respect to runtime, the SURF-based graph construction solutions (namely Kruskal-SURF and Cluster-SURF) individually spent, on average, 12 minutes to process  $100 \times 100$  image adjacency matrices on 16 CPU cores operating at 2.4 GHz. The MSER-based solutions (namely Kruskal-MSER and Cluster-MSER), in turn, spent around 15 minutes to process the same amount of data on the same hardware. This was expected, since the MSER detection process is reportedly slower than SURF. Moreover, taking Kruskal’s algorithm in perspective with the proposed clustered solution, the obtained differences in runtime were always below one millisecond, in favor of Kruskal, for the same data and hardware.

## VI. CONCLUSIONS

The determination of image provenance is a difficult task to solve. The complexity increases significantly when considering an end-to-end, fully-automatic provenance pipeline that performs at scale. This is the first work, to our knowledge, to have proposed such a technique, and we consider these experiments an important demonstration of the feasibility of large-scale provenance systems.

Our pipeline included an image indexing scheme that utilizes a novel iterative filtering and distributed interest point selection to provide results that outperform the current state-of-the-art found in [15]. We also proposed methods for provenance graph building that improve upon the methods of previous work in the field, and provided a novel clustering algorithm for further graph improvement.

To analyze these methods, we utilized the NIST Nimble Challenge [62] and the multiple-parent phylogeny Professional dataset [49] to generate detailed performance results. Beyond utilizing these datasets, we committed to real-world provenance analysis by building our own dataset from Reddit [6], consisting of unique manipulation scenarios that were generated in an unconstrained environment. This is the first work of its kind to analyze fully in-the-wild provenance cases.

Upon scrutinizing the results from the three differently sourced datasets, we observed that the proposed approaches perform decently well in connecting the correct set of images (with reported vertex overlaps of nearly 0.8), but still struggle when inferring edge directions — a result that highlights the difficulty of this problem. Directed edges are dependent on whether the transformations are reversible or can be inferred from pixel information. In this attempt to perform provenance analysis, we found that although image content is the most reliable source of information connecting related images, other external information may be required to supplement the knowledge obtained from pixels. This external information can be obtained from file metadata, object detectors and compression factors, whenever available.

Work in this field is far from complete. The problem of unconstrained, fully-automatic image provenance analysis is not solved. For instance, this work does not currently utilize previous work found in the Blind Digital Image Forensics (BDIF) field. Significant improvements in region localization, provenance edge calculation, and even edge direction estimation could be performed by using systems already created in the BDIF field. We plan to explore the benefits of integrating splicing and copy-move detectors, along with Photo Response Non-Uniformity (PRNU), and Color Filter Array (CFA) models into our pipeline for detecting image inconsistencies and building higher accuracy dissimilarity matrices.

While this work is a significant first step, we hope to spur others on to further investigate fully-automatic image forensics systems. As the landscapes of social and journalistic media change, so must the field of image forensics adapt with them. News stories, cultural trends, and social sentiments flow at a fast pace, often fueled by unchecked viral images and videos. There is a pressing need to find new solutions and approaches to combat forgery and misinformation. Further, the dual-use

nature of such systems makes them useful for other applications, such as cultural analytics, where image provenance can be a primary object of study. We encourage researchers to think broadly when it comes to image provenance analysis.

#### ACKNOWLEDGMENT

This material is based on research sponsored by DARPA and Air Force Research Laboratory (AFRL) under agreement number FA8750-16-2-0173. Hardware support was generously provided by the NVIDIA Corporation. We also thank the financial support of FAPESP (Grant 2017/12646-3, DéjàVu Project), CAPES (DeepEyes Grant) and CNPq (Grant 304472/2015-8).

#### REFERENCES

- [1] H. Farid, "Image Forgery Detection: A Survey," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 16–25, 2009.
- [2] 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*, vol. 43, pp. 1–42, 2011.
- [3] H. Farid, "How to Detect Faked Photos," *Sigma XI American Scientist*, vol. 105, no. 2, pp. 77–85, 2017.
- [4] Z. Dias, A. Rocha, and S. Goldenstein, "Image Phylogeny by Minimal Spanning Trees," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 774–788, 2012.
- [5] Z. Dias, S. Goldenstein, and A. Rocha, "Exploring heuristic and optimum branching algorithms for image phylogeny," *Elsevier Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 1124–1134, 2013.
- [6] Reddit.com, "Photoshobbattles," <https://www.reddit.com/r/photoshobbattles/> (accessed August 11, 2017), 2017.
- [7] J. Backer, "Disinformation," <https://www.youtube.com/watch?v=SdIxzJXNsNA> (accessed August 11, 2017), 1999.
- [8] S. Shane, "From Headline to Photograph, a Fake News Masterpiece," *The New York Times*, <https://www.nytimes.com/2017/01/18/us/fake-news-hillary-clinton-cameron-harris.html> (accessed August 11, 2017), 2017.
- [9] A. Murabayashi, "The Problem of Fake Photos in Fake News," PetaPixel, <https://petapixel.com/2017/01/19/problem-fake-photos-fake-news/> (accessed August 11, 2017), 2017.
- [10] L. Manovich, "Cultural Analytics: Visualizing Cultural Patterns in the Era of 'More Media'," *Domus*, March 2009.
- [11] S. Yamaoka, L. Manovich, J. Douglass, and F. Kuester, "Cultural Analytics in Large-Scale Visualization Environments," *IEEE Computer Society*, vol. 44, no. 12, pp. 39–48, 2011.
- [12] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *ArXiv e-prints*, vol. abs/1702.08734, 2017.
- [13] T. Ge, K. He, Q. Ke, and J. Sun, "Optimized Product Quantization for Approximate Nearest Neighbor Search," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [14] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.
- [15] A. Pinto, D. Moreira, A. Bharati, J. Brogan, K. Bowyer, P. Flynn, W. Scheirer, and A. Rocha, "Provenance Filtering for Multimedia Phylogeny," in *IEEE International Conference on Image Processing (ICIP)*, 2017.
- [16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Springer International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [17] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million Image Database for Scene Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–14, 2017.
- [18] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual Place Recognition: A Survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.
- [19] L. Čehovin, A. Leonardis, and M. Kristan, "Visual Object Tracking Performance Measures Revisited," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1261–1274, 2016.
- [20] A. Jinda-Apiraksa, V. Vonikakis, and S. Winkler, "California-ND: An annotated dataset for near-duplicate detection in personal photo collections," in *IEEE International Workshop on Quality of Multimedia Experience (QoMEX)*, 2013.
- [21] T. Deselaers, T. Gass, P. Dreu, and H. Ney, "Jointly Optimising Relevance and Diversity in Image Retrieval," in *ACM International Conference on Image and Video Retrieval (CIVR)*, 2009.
- [22] T. Deselaers and V. Ferrari, "Global and Efficient Self-Similarity for Object Classification and Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [23] A. Oliva and A. Torralba, "Building the gist of a scene: The role of global image features in recognition," *Elsevier Progress in Brain Research*, vol. 155, pp. 23–36, 2006.
- [24] S. Avila, N. Thome, M. Cord, E. Valle, and A. D. A. Araújo, "Pooling in image representation: The visual codeword point of view," *Elsevier Computer Vision and Image Understanding*, vol. 117, no. 5, pp. 453–465, 2013.
- [25] L. Nanni and A. Lumini, "Heterogeneous bag-of-features for object/scene recognition," *Elsevier Applied Soft Computing*, vol. 13, no. 4, pp. 2171–2178, 2013.
- [26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *ACM Communications*, vol. 60, no. 6, pp. 84–90, 2017.
- [28] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *International Conference on Neural Information Processing Systems (NIPS)*, 2014.
- [29] L. Wang, S. Guo, W. Huang, Y. Xiong, and Y. Qiao, "Knowledge Guided Disambiguation for Large-Scale Scene Classification With Multi-Resolution CNNs," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 2055–2068, 2017.
- [30] M. E. Maresca and A. Petrosino, "MATRIOSKA: A Multi-level Approach to Fast Tracking by Learning," in *Springer International Conference on Image Analysis and Processing (ICIAP)*, 2013.
- [31] G.-Z. Yang and E. Johns, "RANSAC with 2D geometric cliques for image retrieval and place recognition," in *Workshop on Visual Place Recognition in Changing Environments*, 2015.
- [32] A. Joly, O. Buisson, and C. Frelicot, "Content-Based Copy Retrieval Using Distortion-Based Probabilistic Similarity Search," *IEEE Transactions on Multimedia*, vol. 9, no. 2, pp. 293–306, 2007.
- [33] H. Huang, W. Guo, and Y. Zhang, "Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm," in *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, 2008.
- [34] E. Silva, T. Carvalho, A. Ferreira, and A. Rocha, "Going deeper into copy-move forgery detection: Exploring image telltales via multi-scale analysis and voting processes," *Elsevier Journal of Visual Communication and Image Representation*, vol. 29, pp. 16–32, 2015.
- [35] M. Milford, W. Scheirer, E. Vig, A. Glover, O. Baumann, J. Mattingley, and D. Cox, "Condition-invariant, top-down visual place recognition," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5571–5577.
- [36] F. Costa, A. Oliveira, P. Ferrara, Z. Dias, S. Goldenstein, and A. Rocha, "New dissimilarity measures for image phylogeny reconstruction," *Springer Pattern Analysis and Applications*, vol. 20, no. 4, pp. 1289–1305, 2017.
- [37] A. Bharati, D. Moreira, A. Pinto, J. Brogan, K. Bowyer, P. Flynn, W. Scheirer, and A. Rocha, "U-phylogeny: Undirected provenance graph construction in the wild," in *IEEE International Conference on Image Processing (ICIP)*, 2017.
- [38] S. Zagoruyko and N. Komodakis, "Learning to Compare Image Patches via Convolutional Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4353–4361.
- [39] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative Learning of Deep Convolutional Feature Point Descriptors," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 118–126.
- [40] F. Chollet, *Deep Learning with Python*. Manning, 2017.
- [41] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Springer International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [42] L. Kennedy and S.-F. Chang, "Internet image archaeology: Automatically tracing the manipulation history of photographs on the web," in *ACM International Conference on Multimedia*, 2008.

- [43] Z. Dias, A. Rocha, and S. Goldenstein, "Video phylogeny: Recovering near-duplicate video relationships," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2011.
- [44] Z. Dias, S. Goldenstein, and A. Rocha, "Large-Scale Image Phylogeny: Tracing Image Ancestral Relationships," *IEEE Multimedia*, vol. 20, no. 3, pp. 58–70, 2013.
- [45] J. Edmonds, "Optimum Branchings," *Journal of Research of the National Bureau of Standards*, vol. 71, no. 4, pp. 233–240, 1967.
- [46] Z. Dias, S. Goldenstein, and A. Rocha, "Toward image phylogeny forests: Automatically recovering semantically similar image relationships," *Elsevier Forensic Science International*, vol. 231, no. 1, pp. 178–189, 2013.
- [47] F. Costa, M. Oikawa, Z. Dias, S. Goldenstein, and A. de Rocha, "Image phylogeny forests reconstruction," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 10, pp. 1533–1546, 2014.
- [48] M. A. Oikawa, Z. Dias, A. de Rezende Rocha, and S. Goldenstein, "Manifold Learning and Spectral Clustering for Image Phylogeny Forests," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 5–18, 2016.
- [49] A. de Oliveira, P. Ferrara, A. De Rosa, A. Piva, M. Barni, S. Goldenstein, Z. Dias, and A. Rocha, "Multiple parenting phylogeny relationships in digital images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 2, pp. 328–343, 2016.
- [50] National Institute of Standards and Technology, "Nimble Challenge 2017 Evaluation Plan," [https://w3auth.nist.gov/sites/default/files/documents/2017/09/07/nc2017evaluationplan\\_20170804.pdf](https://w3auth.nist.gov/sites/default/files/documents/2017/09/07/nc2017evaluationplan_20170804.pdf) (accessed October 11, 2017), 2017.
- [51] —, "Nimble Challenge 2017 Evaluation," <https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation> (accessed October 5, 2017), 2017.
- [52] L. A. Barroso, J. Dean, and U. Holzle, "Web search for a planet: The google cluster architecture," *IEEE Micro*, vol. 23, no. 2, pp. 22–28, 2003.
- [53] J. Cheng, "TinEye image search helps ferret out copyright ripoffs," <https://arstechnica.com/uncategorized/2008/08/tineye-image-search-helps-ferret-out-copyright-ripos/> (accessed June 14, 2018), 2008.
- [54] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Elsevier Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [55] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.
- [56] J. Brogan, P. Bestagini, A. Bharati, A. Pinto, D. Moreira, K. Bowyer, P. Flynn, A. Rocha, and W. Scheirer, "Spotting the difference: Context retrieval and analysis for improved forgery detection and localization," in *IEEE International Conference on Image Processing (ICIP)*, 2017.
- [57] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions," *Elsevier Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.
- [58] J. Fiscus, H. Guan, Y. Lee, A. Yates, A. Delgado, D. Zhou, D. Joy, and A. Pereira, "The 2017 Nimble Challenge Evaluation: Results and Future Directions," [https://www.nist.gov/sites/default/files/documents/2017/07/31/nist2017mediaforensicsworkshop\\_20170726.pdf](https://www.nist.gov/sites/default/files/documents/2017/07/31/nist2017mediaforensicsworkshop_20170726.pdf) (accessed May 21, 2018), 2017.
- [59] P. Papadimitriou, A. Dasdan, and H. Garcia-Molina, "Web graph similarity for anomaly detection," *Springer Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 19–30, 2010.
- [60] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2227–2240, 2014.
- [61] —, "Fast matching of binary features," in *IEEE Conference on Computer and Robot Vision (CRV)*, 2012, pp. 404–410.
- [62] National Institute of Standards and Technology, "The 2017 Nimble Challenge Evaluation Datasets," <https://www.nist.gov/itl/iad/mig/nimble-challenge> (accessed October 11, 2017), 2017.