Discovering Image Manipulation History by Pairwise Relation and Forensics Tools

Xu Zhang, Zhaohui H. Sun, *Senior Member, IEEE*, Svebor Karaman, and Shih-Fu Chang, *Fellow, IEEE*

***Abstract*—Given a potentially manipulated probe image, provenance analysis aims to find all images derived from the probe (offsprings) and all images from which the probe is derived (ancestors) in a large dataset (provenance filtering), and reconstruct the manipulation history with the retrieved images (provenance graph building). In this paper, we address two major challenges in provenance analysis, retrieving the source image of the small regions that are spliced into the probe image, and, detecting source images within the search results. For the former challenge, we propose to detect spliced regions by pairwise image comparison and only use local features extracted from the spliced region to perform an additional search. This removes the influence of the background and greatly improves the recall. For the latter, we propose to learn a pairwise ancestor-offspring detector and use it jointly with a holistic image manipulation detector to identify the source image. The proposed provenance analysis system has performed remarkably in evaluations using comprehensive provenance datasets. It's the winning solution for NIST Media Forensics Challenge (MFC) in 2018, 2019 and 2020. In MFC 2019, our provenance results achieved a 12% improvement in filtering and a 20% gain in oracle provenance graphs building over the alternative methods. In the real-world Reddit dataset, the edge overlap between our reconstructed provenance graphs and the ground-truth graphs is 5 times better than the state-of-the-art system.**

***Index Terms*—Image provenance, image forensics, image retrieval, graph reconstruction.**

I. Introduction

IMAGE manipulation, often used in changing image content and semantic meaning, is becoming increasingly easy with the prevalence of image editing tools and computer vision algorithms. It brings enhanced capabilities to art, photography and entertainment industries. However it also causes great concerns in security and ethics, since the traditional perception of treating visual media as trustworthy content is no longer valid.

Manuscript received December 1, 2019; revised March 16, 2020; accepted May 4, 2020. Date of publication June 3, 2020; date of current version August 24, 2020. This work was supported by the United States Air Force Research Laboratory (AFRL) and the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-16-C-0166. Any opinions, findings and conclusions or recommendations expressed in this material are solely the responsibility of the authors and does not necessarily represent the official views of AFRL, DARPA, or the U.S. Government. The guest editor coordinating the review of this manuscript and approving it for publication was Matt Turek. *(Corresponding author: Xu Zhang.)*

Xu Zhang, Svebor Karaman, and Shih-Fu Chang are with the Department of Electrical Engineering, Columbia University in the City of New York, New York, NY 10027 USA (e-mail: spongezhang@gmail.com; svebor.karaman@gmail.com; sc250@columbia.edu).

Zhaohui H. Sun is with the Kitware, Inc., Clifton Park, NY 12065 USA (e-mail: harry.sun@kitware.com).

Digital Object Identifier 10.1109/JSTSP.2020.2999827

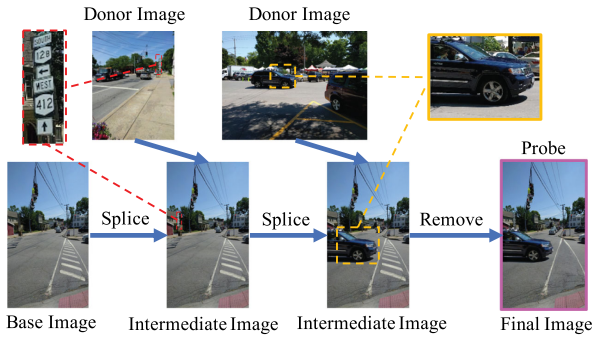


Fig. 1. Example provenance graph. Given a probe image (in magenta), our system uses a filtering step to find relevant images and uses a graph reconstruction step to reconstruct a provenance graph revealing the possible manipulation history. In this example, the street sign (in red) and the car (in yellow) were cropped and spliced into the image. The white crosswalk lines were removed.

Over the past few decades, the image forensics community has developed a large number of tools to detect and localize potential manipulations of images [1] for security reasons. However, simply finding whether an image has been manipulated or not is insufficient in some situations. Indeed, almost every image published on the Internet has undergone "friendly" manipulations to provide a better viewing experience. Such modifications should clearly be distinguished from image manipulations with malicious intent. Having the ability to tell the intent of the manipulation is of great importance. However, it is still an open problem, and very few forensics studies have looked into this direction.

Provenance analysis research assumes that an image does not exist in a vacuum. Each image has a history and context that grows over time [2]. The image editor generally starts with some versions of the image that has already been manipulated. The manipulation history and the life-cycle of the image often provide important forensics information. To reveal the information, starting with a potentially manipulated image as a probe, the goal of provenance analysis is to search for images from which the probe was derived (ancestors), and also images that were derived from the probe (offspring), then reconstruct the manipulation history using those retrieved images. Provenance analysis is very different from image manipulation detection. Given a probe image, as highlighted in the magenta box in Fig. 1, image manipulation detection aims at determining whether the image has been manipulated or not. However, provenance analysis seeks to recover the provenance graph (Fig. 1). It also tells us the history of the image manipulation, including the original information of the image, the spliced information (the foreground black car

and the road sign) and the removed information (the crosswalk). The ability of provenance analysis and manipulation graph reconstruction is very important for image forensics, social media analysis, copyright protection and many other applications. It can be used to support downstream analysis such as attribution of the sources and reasoning about the possible intent of the manipulation based on the history of the manipulations and paths of dissemination among sites and players involved in the manipulation graph.

Provenance analysis contains two main steps:

- provenance filtering: given a probe image, find all ancestor and offspring images related to the probe, and
- provenance graph building: reconstruct the manipulation history from the provenance filtering result to capture the source-manipulation relation between any pair of images in the provenance filtering set.

Provenance analysis involves three major challenges: 1) the probe image may be a composition of multiple source images, and some source images may only contribute small objects to the probe (for example, swapping a face in a party photo with another face from a different image). Retrieving the source image of the small object could be quite challenging; 2) Determining the source images (original images) among a number of similar images in the search result is critical and challenging; And, 3) constructing the provenance graph is an NP-hard problem. Some manipulations could be quite subtle (for example, removing or replacing a number on a ticket). It is challenging to accurately reconstruct the manipulation history with such subtle operations.

To address these challenges in provenance analysis, we make unique contributions in the following aspects:

1) we propose to learn a pairwise ancestor-offspring classifier and integrate it with the image manipulation detector to determine source images;
2) we propose to detect potential spliced regions in the probe image, and only use features within the spliced regions for provenance filtering. We show the effectiveness of the proposed method in retrieving the source images of small spliced regions;
3) we improve the graph building algorithm by combining both the local feature matching score and the pixel similarity score.

The proposed algorithm outperforms the state-of-the-art provenance analysis algorithms by a very large margin. In the NIST Media Forensics 2019 dataset, our provenance results achieve the top performance, with a 12% improvement in filtering and a 20% gain in oracle provenance graphs building over the alternative methods. In the real-world Reddit dataset, the edge overlap between our reconstructed provenance graphs and the ground-truth graphs is 5 times better than the state-of-the-art system.

II. Related Work

A. Provenance Analysis

Research on provenance analysis dates back to 2008 when Kennedy and Chang [2] introduced the first work "Internet Image Archaeology" showing that we can reconstruct a plausible manipulation history by detecting directional manipulations (such as cropping and resizing) and checking the consistency of the manipulation operations. The paper also shows an interesting finding that manipulation operations often change the semantic meaning and the public viewpoint of the original image throughout its life cycle. Instead of detecting directional manipulations and checking consistency, Dias *et al.* [3] propose to construct a dense graph, by calculating pairwise dissimilarity scores between all possible image pairs and cutting down the dense graph to a minimum spanning tree with the oriented Kruskal algorithm. The tree is then considered as the provenance graph (a.k.a. phylogeny tree). Compared to the manipulation detection based method [2], this optimization based algorithm can deal with unknown types of manipulation operations. Different tree reconstruction methods and dissimilarity scores have been explored in [4] and [5], respectively.

The above methods assume there is only one source image in the provenance graph or tree. Dias *et al.* extend the single provenance tree to the provenance forest, which contains multiple provenance trees from different source images [6]. Oliveira *et al*. further consider the relation between different provenance trees in the forest [7], [8]. They reconstruct the link between splice operations by assuming there are three kinds of trees: host/background, alien/foreground and composition trees. Bharati *et al*. [9], [10] and Moreira *et al*. [11] address the provenance analysis problem with a large-scale image dataset with distractors. The setting is proposed in the Media Forensics Challenge (MFC) hosted by NIST [12]–[15], which is very close to the real-world application. Besides images, there is also research for provenance analysis on video [16], [17] and text [18].

B. Content-Based Image Retrieval

The provenance filtering task in provenance analysis is closely related to Content-Based Image Retrieval (CBIR). Both provenance filtering and CBIR aim at retrieving a subset of "related" images from a large gallery image set. The main difference is that provenance filtering focuses on retrieving images from the same source, while CBIR aims at retrieving images containing the same or similar content (*e.g.* same landmark or similar products). Nevertheless, technologies used for these two tasks are very similar.

Features of different levels are widely used in provenance filtering and CBIR, including: 1) key-point feature, such as SIFT [19] and deep learning based local feature [20], 2) region level feature [21], and 3) image based feature [22], [23]. To enable fast search in a large image set, multiple indexing and quantization methods have been proposed [24], [25]. Recently, with the help of the GPU, indexing and searching billion-scale datasets becomes feasible, even with a single personal computer [26], [27]. Post-processing methods in CBIR, such as geometric verification [28] and query expansion [29], have also been proved to be effective in provenance filtering [30].

In spite of all these similarities, an important and unique challenge exists in provenance filtering. That is retrieving the source images of small spliced regions in a probe image.

C. Other Image Forensics Research

Image forensics has long been studied [31]. Many methods have been proposed to automatically detect manipulations at

digital, physical, and semantic levels. At the digital level, the studied manipulations include metadata change, double compression [32], contrast enhancement [33], intensity and color adjustment, image sharpening, blurring, median filtering, image recapture [34], image cropping, face swapping, seam carving [35], object copy-paste [36], content-aware fill, JPEG dimple [37], and image splicing [38]. Image forensics research at the physical level includes reflection and shadow authentications [39], and the detection of computer generated images [40]. Semantic-level manipulation (e.g. event re-purposing) has been studied in [11] and [41].

The rest of the paper is organized as the following. The provenance filtering algorithm is presented in Sec. III. The provenance graph building algorithm is presented in Sec. IV. The experimental results are shown in Sec. V, followed by conclusions in Sec. VI.

III. Provenance Filtering

Given a probe image I_q, which may have been manipulated, and a set of N world/gallery images $\{I_1, \ldots, I_N\}$, the goal of provenance filtering is to identify a ranked list of K images $\mathcal{I}^{(q)} = \{I_1^{(q)}, \ldots, I_K^{(q)}\}$. Each image $I_i^{(q)}$ is associated with a score indicating how likely the image belongs to the same provenance graph as the probe image.

Compared to the conventional image retrieval methods, which have been explored for decades, the major challenge in provenance filtering is searching for the sources of small spliced regions. Image splicing is one of the most commonly used image manipulation techniques. It produces a composite image by splicing part of an image (donor image) into another image (base image). The operation is to remove the original content in the base image, and/or add new content from the donor image to the base image. In real-world image manipulations, some spliced regions can be very small (less than 0.1% of the size of the base image), but may induce a dramatic change of the meaning of the image (*e.g.* changing the face of a person or the text in the image). Searching for the sources of small spliced regions, a.k.a. donor images, within a large-scale world image set is critical and challenging.

The proposed provenance filtering system contains two steps, first a conventional local feature-based search with query expansion which is followed by an additional search based on splice detection. By limiting the search to local features extracted from the suspected spliced regions, we can improve the recall performance of the donor images of the small spliced regions. Our provenance filtering pipeline is illustrated in Fig. 2.

A. Local Feature-Based Provenance Filtering

We adopt a local feature-based image retrieval method for provenance filtering. Local key-point features are effective for image retrieval with small spliced regions from the donor images, while the image-level features are likely to fail. Given the world image set, with up to millions of images or more, local feature detector (SIFT [19]) is used to extract key points and associated descriptors from each image. Due to memory constraint, each of large images is down-scaled to around 7

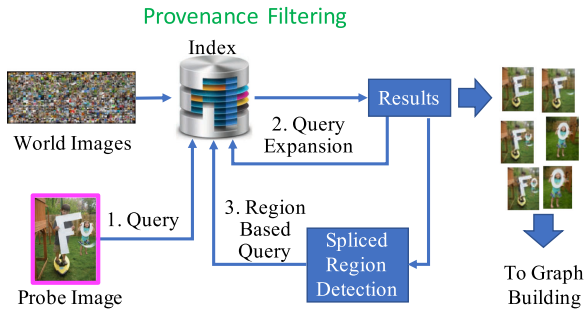


Fig. 2. Provenance filtering pipeline. The proposed algorithm has three search steps. The query image is used for the first step search to get the initial result. From the initial result, we further apply query expansion and region-based search to retrieve more images. The final result is obtained by score-fusion of the three search results.

megapixels. The SIFT descriptor is ℓ_2 normalized. We empirically find out that SIFT is rather robust to image manipulation such as image blurring and compression. With more than a million world images, a typical scale for a real-world problem, the number of descriptors can easily go up to a few billions.

To efficiently perform a search over such a large database, we adopt the open-source FAISS library [26]. Specifically, a two-step indexing method is utilized to index the billion-scale local descriptor set. The first step is a k-means based coarse quantization, which decomposes the overall dataset into multiple exclusive clusters. It helps quickly reduce the scale of the problem. Within each cluster, a second step of fine quantization utilizes Optimized Product Quantization (OPQ) [25] to index the residual of each descriptor with respect to its cluster center. For details of the indexing method, we refer readers to [26].

A probe image may contain thousands of descriptors. For each descriptor, the search algorithm first finds a few cluster centers that are closest to the descriptor. Within each cluster, it searches for a fixed number of descriptors that are the closest to the query descriptor. The approximated distance $d(p_k, q_{ij})$ between the ℓ_2 normalized descriptor p_k in query image $I^{(q)}$ and each ℓ_2 normalized returned descriptor q_{ij} (the reconstruction of the j^{th} descriptor in image I_i by the quantization method) is converted to a similarity score by the following equation:

$$s_{\text{local}}(p_k, q_{ij}) = \max(0, (1 - d(p_k, q_{ij})/0.15)^3). \quad (1)$$

The reason why we choose 0.15 here is that we notice the matching becomes unreliable when the distance is larger than 0.15. The cubic function is to encourage smaller distance. The descriptor-level score above is then aggregated to an image-level score by,

$$s_{\text{image}}(I^{(q)}, I_i) = \sum_{j,k} s_{local}(p_k, q_{ij}) \quad (2)$$

The gallery images are ranked based on the image-level similarity score.

Query expansion (QE) [28] is also applied to help retrieve more related images. Initial returned images with image-level similarity score higher than a threshold are queried again with their local features. Lastly, the top K images after multiple query

expansion steps are returned and referred to as the filter set in the following.

B. Searching for Small Donors

One of the major challenges of provenance filtering is searching for the sources of the small spliced regions. The reason is that the number of features in an image may be dominated by the number of local features in the background region, causing false retrieval of a large number of distractors, which rank higher than the real donor images. To address this issue, we propose to remove local features in the background and only use the local features in the spliced region for image search.

The spliced region in a probe image can be possibly detected by a splice detector. However, existing off-the-shelf image splice detectors are still limited in terms of detection and localization accuracies. Localizing the spliced region in a single manipulated image remains challenging to date. Inspired by the famous game "Spot the Difference", we notice that if the manipulated image and the original image are put side-by-side, spotting the differences between these two images becomes much easier. We propose to search for small changes through comparison of similar image pairs within the filter set of images obtained from provenance filtering.

Specifically, for each image in the filter set with a similarity score (with respect to the probe image) higher than a threshold, its most similar image in the filter set is selected and compared to the image. We use this pair of similar images in the filter set to spot the potential spliced region. Various change detection methods can be adopted to spot the potential spliced area. Here, as a proof of concept, we use a simple method - the number of changed pixels is less than a certain percentage of the image. An appropriate value of the parameter can be determined through experiments on separate validation sets. For example, the percentage of changed pixels is set to 10% in our implementation. Note here we focus on spotting of small spliced regions and using them as additional queries. Our rationale is that source images of large spliced regions are expected to be covered by the image-level search method described in the previous subsection. Because of the sufficient number of local features in the spliced region, the impact of distractor features from the non spliced area is lower and the risk of missing the source images of the spliced region is thus expected to be lower.

Local features are extracted from the local regions and used to search the index for additional source images. To extract local features from the local regions, the original image with highest resolution is used instead of a down-scaled image. Features from all potential spliced regions are aggregated for the region-level search over the index built in Sec. III-A. The image-level search results and the region-level search results are fused by re-ranking the similarity score. The K images with the highest score overall are returned as the final filter set. Note, if one image appears in both image-level and region-level results, the highest score of the image is regarded the final score for fusing.

IV. Provenance Graph Building

Given the probe image I_q and the final filter set $\mathcal{I}^{(q)}$ from the provenance filtering step, the goal of the provenance graph building is to recover the manipulation history by building a graph $\mathcal{G}$ with a vertex set $\mathcal{V} \subseteq \mathcal{I}^{(q)} \cup I_q$ and a directed edge set $\mathcal{E} = (V_i, V_j)$, where $V_i \in \mathcal{V}$, $V_j \in \mathcal{V}$, $V_i \neq V_j$, and (V_i, V_j) means V_j is directly derived from V_i. The reconstructed graph $\mathcal{G}$ should be as close as possible to the ground-truth provenance graph. Note that recent provenance research only focuses on the direction of the edge, not the operation.

We first remove all images in the filter set with a similarity score less than a threshold (20 in implementation) with respect to the probe image. It helps remove the false retrieved images. Remaining images in the filter set may be derived from multiple source images. In order to solve the problem, we make a few assumptions: 1) images sharing the same background are derived from the same source image, 2) images from the same source form a provenance arborescence (directed rooted tree), and 3) different trees are connected by splicing operations/edges. These assumptions cover many of the typical image manipulation processes. To reconstruct the graph, we first run hierarchical clustering over all the images in the filter set. Given two images I_1, I_2, the distance between two images is defined as

$$d(I_1, I_2) = max\left(0, 1 - \frac{s_{\text{image}}(I_1, I_2)}{\max(\parallel I_1 \parallel, \parallel I_2 \parallel)}\right) \tag{3}$$

where $s_{\text{image}}(I_1, I_2)$ is the image-level similarity score defined in Eq. (2) and $\parallel I \parallel$ is the number of the local feature in I. The distance between two image clusters is the minimum distance of any two images in these two clusters. The hierarchical clustering stops when the minimum distance between clusters is larger than 0.7. This helps split all the images into different groups. We assume images in the same group come from the same source image. In each cluster, the source image (the least manipulated image) is determined by using a pair-wise relation classifier and an image manipulation detector. Given the cluster and the source image, a directed tree is built as the manipulation history within the cluster. Finally, we link different trees by finding the splicing links between them. The overall graph building pipeline is shown in Fig. 3. In the following, we provide details of each step.

A. Similarity Score

The graph construction is based on a simple intuition that if two images are similar, they are more likely to be directly connected in the graph. Given a cluster of images, we compute the similarity/affinity scores between all $n(n-1)$ image pairs. Specifically, the similarity score $s_{sim}(I_i, I_j)$ between an image pair (I_i, I_j) is defined as:

$$s_{sim}(I_i, I_j) = s_{lf}(I_i, I_j) + \lambda s_{pixel}(I_i, I_j), \tag{4}$$

where $s_{lf}(I_i, I_j)$ is the number of the matched SIFT points between two images [19], $s_{pixel}(I_i, I_j)$ is a pixel level similarity score and λ is the weighting parameter. To calculate $s_{pixel}(I_i, I_j)$, we first align two images with matched local features. If two images can be aligned, $s_{pixel}(I_i, I_j)$ is the number of pixels with the same color at the same location of the two aligned images. Otherwise, $s_{pixel}(I_i, I_j) = 0$. The pixel level score is very useful with minor local manipulation, which may not be captured by the local feature.

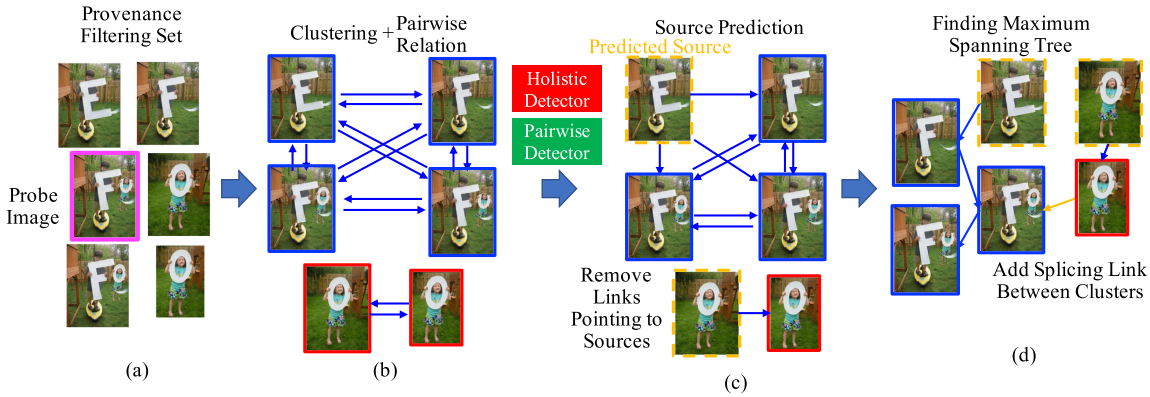


Fig. 3. Provenance graph building pipeline. (a) Shows an example filter set from provenance filtering step, the probe image is highlighted in the magenta box. In (b), we compute the similarity score between images and cluster the images based on the similarity score. After that, the source image in each cluster is predicted and the links pointing to the sources are removed in (c). The final graph is derived by finding the maximum spanning tree within each cluster and splicing links between clusters (d).

After calculating the similarity score, we have a full directed graph with $n(n-1)$ edges and each edge has a similarity score as its weight. However, since the similarity score is symmetric, $s_{sim}(I_i, I_j) = s_{sim}(I_j, I_i)$, directly optimizing over such a graph will result in non-unique result. To address this issue, we detect a source image in each cluster and remove all the directed edges that point to the source. It removes the ambiguity in graph reconstruction.

B. Source Identification

Two pieces of information are combined to identify the source within each image cluster, including a integrity score and pairwise ancestor/offspring relations.

1) Likelihood of being Manipulated: Source images in the provenance graph are original images which have not been manipulated. An intuitive idea to detect the source image among a cluster of images is to estimate the integrity score for each image, and choose the one that is the least likely to have been manipulated as the source image.

In the past decades, a large number of image manipulation detectors have been developed to detect whether an image has been manipulated or not. Following the trends of deep neural network, deep learning based image manipulation detectors have become very popular in the image forensics community. Recently, the holistic image manipulation detector [42] has attracted a lot of interests, due to its competitive detection performance and robustness to various manipulation types. The holistic image manipulation detector assesses the integrity of a probe image, and determines if the image has been modified by classifying the median-filtered residual image using a convolutional neural network. A probe image is uniformly partitioned into non-overlapped tiles. The filtered image tiles pass through a CNN classifier to get tile-level integrity scores. The tile-level integrity score is sorted in descending order. And the i^{th} tile-level score is chosen as the image-level integrity score, where $i = \text{int}(\alpha * N)$, N is the total number of tiles, and α is a hyper-parameter. The holistic detector is agnostic to the specific manipulation types and is applicable to a wide variety of manipulations and their combinations. Since the provenance graph may contain many different manipulation types, the holistic detector suits the objective of the source image detection task well.

In source image detection, each image in an image cluster is passed through the holistic detector and get a normalized integrity score ($s_{integrity} \in [0, 1]$), indicating the likelihood of whether the image is original or not. The image with the highest integrity score is considered as the source image.

The holistic detector checks image one by one individually. However, in many cases, it's hard to assign an accurate integrity score to a single individual image. For example, given two images, one is the original and the other is slightly blurred. It's hard to tell which image is more original until comparing these two images side-by-side. Inspired by this, we propose to estimate the pairwise ancestor-offspring relationship between two input images.

2) Pairwise Ancestor-Offspring Relation: For any pair of images within a cluster, the two images are first aligned spatially by local feature matching. If the image alignment fails, we conclude that one image cannot be derived from the other. For image pairs that can be aligned, a deep neural network is learned to determine whether one image is derived from the other. Many image manipulations are local manipulations (only a small portion of the image is changed). Instead of taking the whole image as input, the network takes image patches sampled from the changed regions (the gray-scale level changes more than 5) of the two images as input. Sampling patches from the changed regions helps the network focus on the local manipulations. The training pipeline is shown in Fig. 4. We choose L2-Net [43] as the base network, which takes two 32×32 image patches as input. It's a lightweight CNN which is specifically designed for patch-level tasks.

Given n images in an image cluster $\{I_1, \ldots, I_n\}$, in order to calculate the pairwise ancestor score of image I_i with respect to all the other images in the group, we consider $n-1$ image pairs (I_i, I_j), $i \neq j$. For image pair (I_i, I_j), we sample M aligned image patches $(P_m^{(ij)}, Q_m^{(ij)}), 1 \leq m \leq M, P_m^{(ij)} \in I_i$ and $Q_m^{(ij)} \in I_j$ from the changed region of I_i and I_j. The patch pairs are fed to the pairwise ancestor-offspring detection network, $f(P_m^{(ij)}, Q_m^{(ij)}) : \mathbb{R}^{d \times d} \times \mathbb{R}^{d \times d} \rightarrow \{0, 1\}$, where d is

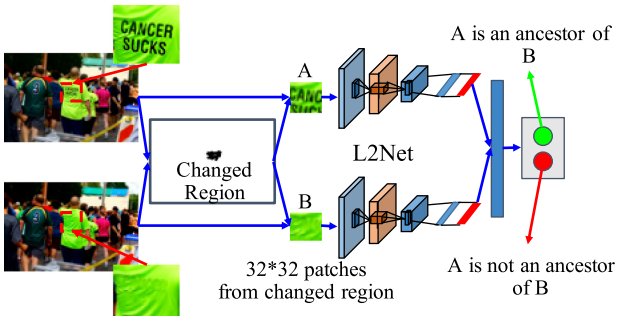


Fig. 4. Training of the pairwise ancestor-offspring relation recognition network. We sample patch pairs from the changed region of two images and predict an ancestor-offspring score for each patch pair.

the width and height of the image patch. The output of the network indicates whether $P_m^{(ij)}$ is an ancestor of $Q_m^{(ij)}$ or not. For image I_j, image I_i gets a pairwise ancestor score of:

$$s_p(I_i, I_j) = \sum_{m=1}^{M} f(P_m^{(ij)}, Q_m^{(ij)}). \tag{5}$$

For all $n-1$ possible image pairs, image I_i gets an unnormalized ancestor score of:

$$s_a(I_i) = \sum_{j \neq i} s_p(I_i, I_j). \tag{6}$$

We normalize the pairwise ancestor-offspring by the sum of the unnormalized score over all the images:

$$\hat{s}_a(I_i) = \frac{s_a(I_i)}{\sum_{j=1}^{n} s_a(I_j)}. \tag{7}$$

The final source score of image I_i is defined as the average of the integrity score $s_{integrity}$ and the normalized ancestor score $\hat{s}_a$. The source of the image cluster is chosen as the image with the highest final score. Once the source in a cluster is identified, all the incoming links to the source (indicating the source as the descendent node) are removed. This is to ensure the final constructed graph will have the source as the root and to remove the ambiguity caused by the symmetric similarity score in graph building.

C. Graph Construction

We applied Chu−Liu/Edmonds' algorithm [44] to the graph with edges pointing to the source removed to find the maximum spanning arborescence. The Chu−Liu/Edmonds' algorithm takes a directed graph as input, with a distinguished vertex as root and a weight associating with each edge. It returns a spanning arborescence rooted at the given root vertex with the maximum/minimum sum of weights of all edges. This spanning arborescence is regarded as the provenance graph for this cluster. Technically, we could combine both the similarity score and the pairwise ancestor score ($s_p(I_i, I_j)$) as the weight of the edge. However, since we've already applied the ancestor score to find the source, applying the ancestor score here is redundant. In addition, it requires an additional step of normalizing and fusing the scores. In practice, we also don't observe that combining both scores helps improve the final performance.

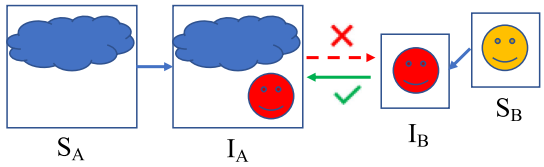


Fig. 5. We determine the direction of splicing by checking the matching result. Since there are matches between S_B and I_A and no match between S_A and I_B, the direction is from I_B to I_A.

To link different arborescences, consider two arborescences A and B with source images S_A and S_B, respectively. We identify a pair of images (I_A, I_B) with $I_A \in A$ and $I_B \in B$ which have the maximum number of matched SIFT points among all image pairs from these two arborescences. If the number of the matched local feature is larger than a threshold (20 in implementation), it indicates there may exist an splicing operation between image I_A and image I_B. To figure out which arborescences is the host/background, we check the number of matched local features between source image S_A and I_B. If arborescences A is the host, we shouldn't be able to find any matches between S_A and I_B, since the content of I_B is spliced into offsprings of S_A, but not S_A itself. If there is no match between S_A and I_B, we make a link from I_B to I_A indicating that content in I_B is spliced to I_A (Fig. 5). We also check S_B and I_A likewise.

D. Time Complexity Analysis

Assume there are a total number of N images related to the query image. The complexity of calculating the holistic score is $O(N)$. Considering the worse case in which all the images belong to the same cluster, the complexity of calculating the pairwise score is $O(N^2)$. The complexity for hierarchical clustering is $O(N^2 logN)$. And the complexity of the Chu−Liu/Edmonds' algorithm is $O(N^2)$. Overall, the time complexity of the proposed graph building method is $O(N^2 logN)$.

V. Experiment

A. Data

We conduct experiments with three provenance evaluation datasets developed by NIST for the annual Medifor (Media Forensics) challenge from 2017 to 2019 and one real world dataset collected by Moreira *et al.* [11] from the Photoshop battle community on Reddit. We here provide details on each dataset:

- NC2017EP1 [12] is a dataset released by NIST in 2017 which contains a total of 1,008,681 world images with 151 probe images.
- MFC18EP1 [13] is a dataset released by NIST in 2018 which contains a total of 1,020,343 world images with 897 probe images.
- MFC19EP1 [14] is a dataset released by NIST in 2018 which contains a total of 2,013,732 world images with 1,027 probe images.

- Reddit is a dataset that contains 184 provenance graphs[1] with 10,421 images in total. Since this dataset does not contain a distractor world image set, we only use it to evaluate the oracle provenance graph building performance (Sec. V-E).

All Medifor datasets contain the ground-truth provenance graph recorded when the manipulated images were created. The datasets contain around 50 different manipulation operations. The number of nodes in the provenance graph varies from 2-200+. More details of the datasets can be found in [15]. The Reddit dataset does not record the real world manipulation history. A noisy "ground-truth" provenance graph is inferred from the corresponding Reddit posts. Since the Reddit dataset is sampled from real-world Reddit photoshop battle posts, a lot of intermediate manipulated images may be missing and the posted image may be re-compressed. These make the Reddit dataset a very challenging real-world testset for provenance analysis.

B. Tasks and Evaluation Metrics

We evaluate our system and the state-of-the-art methods for the Provenance Filtering and the Provenance Graph Building tasks, as defined in the NIST MediFor challenge [12].

1) Provenance Filtering: Given a probe image, the task of Provenance Filtering is to search for images in the world image set which are in the same provenance graph as the probe and to return the retrieved result as a ranked list. The performance is evaluated by the recall at different cut-off positions of the returned ranked list. Specifically, we apply the recall values of top $\{200, 300\}$ images, namely R@200, and R@300 [15]. A higher recall means a better performance.

2) Provenance Graph Building: Given a probe image and a list of images, the task of Provenance Graph Building is to reconstruct the provenance graph. Depending on how the image list is obtained, the task can have two different settings: 1) end-to-end, the image list is the ranked list returned by provenance filtering, and 2) oracle, the image list is a clean and complete list of images which are known to be in the provenance graph. In the end-to-end setting, the input image list may contain distractors mistakenly retrieved by the filtering algorithm or miss some of the "ground-truth" images. This setting is designed to test the performance of the overall provenance system. The oracle setting, with its clean and complete image list, is designed to test the performance of the provenance graph building algorithm only.

We use the overlap between the ground-truth graph and the reconstructed graph to evaluate the performance of the graph building algorithm. Assuming the ground-truth graph G_{gt} contains a vertex set V_{gt}, a source image vertex set S_{gt} (with $S_{gt} \subseteq V_{gt}$) and an edges set E_{gt}, while the reconstructed graph G_{rec} contains a vertex set V_{rec}, a source image vertex set S_{rec} (with $S_{rec} \subseteq V_{rec}$), and an edges set E_{rec}. The source vertex recall (SR), source vertex overlap (SO), vertex recall (VR), vertex overlap (VO), edge overlap (EO), and full graph overlap, *i.e.*, vertex and edge overlap (VEO), measuring between the two graphs are defined as:

$$
\begin{aligned}
SR &= \frac{|S_{gt} \cap S_{rec}|}{|S_{gt}|}, \quad SO = 2\frac{|S_{gt} \cap S_{rec}|}{|S_{gt}| + |S_{rec}|}, \\
VR &= \frac{|V_{gt} \cap V_{rec}|}{|V_{gt}|}, \quad VO = 2\frac{|V_{gt} \cap V_{rec}|}{|V_{gt}| + |V_{rec}|}, \\
EO &= 2\frac{|E_{gt} \cap E_{rec}|}{|E_{gt}| + |E_{rec}|}, \\
VEO &= 2\frac{|V_{gt} \cap V_{rec}| + |E_{gt} \cap E_{rec}|}{|V_{gt}| + |V_{rec}| + |E_{gt}| + |E_{rec}|},
\end{aligned} \tag{8}
$$

where $|\cdot|$ is the number of element (vertex or edge) in the set.

C. Implementation Details

Our implementation is available at https://github.com/Columbia-Provenance-Analysis/Provenance_Analysis

1) Provenance Filtering: For indexing of the world image set, we extract SIFT features for all the images and ℓ_2 normalize the descriptors. We run FAISS [26] to build a two-step index. The first step contains 262,144 centroids in k-means. In the second step, a 128 dimensional residual feature is transformed to a 32 dimensional feature and quantized to an 8-byte code by OPQ. The image similarity score threshold used in query expansion is set to 50 based on empirical validation.

2) Provenance Graph Building: For provenance graph building, during the training stage, two deep neural network models are learned. The holistic image manipulation detector [42] is trained with 800,000 labeled images. Following the implementation in [42], high-pass filtering is applied to remove scene content and retain residues corresponding to camera fingerprints and manipulation artifacts. A VGG-16 [45] is trained to predict manipulation labels based on the residual images. All the models are trained and all the hyperparameters are tuned with the NC2017 and MFC18 development datasets provided by NIST [15]. The development datasets and the evaluation datasets have no overlap.

The pairwise ancestor-offspring relation network is trained with 100k patch pairs. When training the network, stochastic gradient descent with a momentum of 0.9 and a starting learning rate of 0.01 are used. The network is trained for 20 epochs.

During graph construction, we first apply the hierarchical clustering to input images. Within each image cluster, the integrity score of each image is computed by the holistic image integrity detector and the two-branch ancestor-offspring relation network is applied to get the pairwise score. The percentile value α is set to 0.023 and the parameter λ in Eq. (4) is set to 0.01.

We compare the proposed algorithm with the state-of-the-art provenance analysis system [11], which applies distributed interest point selection and iterative filtering in provenance filtering, and, geometrically consistent matching and mutual information based dissimilarity score in graph building. We also compare with [10], which further takes advantage of the metadata.

[1]Due to the copyright issue, the Reddit dataset only provides links to the images instead of images themselves. Since the download links are broken for the source images of 13 graphs, we use 171 graphs for evaluation.

TABLE I
PROVENANCE FILTERING RESULT (%)

	R@200	R@300	BR	DR	IR
	NC2017EP1				
[11]	69.8	69.8	N/A	N/A	N/A
Baseline	66.6	66.9	66.2	40.1	67.3
QE	70.0	70.1	70.1	49.5	70.5
Ours	**71.9**	**72.2**	**70.7**	**56.8**	**71.6**
	MFC18EP1				
[11]	89.2	89.5	N/A	N/A	N/A
Baseline	90.2	90.3	92.0	49.0	89.9
QE	91.6	91.8	93.8	58.7	91.4
Ours	**92.9**	**93.1**	**94.0**	**71.7**	**92.6**
	MFC19EP1				
[11]	81.4	81.9	N/A	N/A	N/A
Baseline	90.4	90.4	94.9	45.9	89.6
QE	91.8	92.0	**95.4**	63.9	91.0
Ours	**92.9**	**93.1**	**95.4**	**71.1**	**91.9**

TABLE II
PROVENANCE FILTERING RESULT (%) FOR DONORS OF DIFFERENT SIZES

	#SD	SDR	#MD	MDR	#LD	LDR
	NC2017EP1					
Baseline	73	29.7	113	43.0	91	44.9
QE	73	34.1	113	59.5	91	**49.5**
Ours	73	**58.0**	113	**60.8**	91	**56.8**
	MFC18EP1					
Baseline	261	23.0	250	52.4	228	75.0
QE	261	35.1	250	64.4	228	79.5
Ours	261	**57.9**	250	**78.3**	228	**80.1**
	MFC19EP1					
Baseline	195	28.3	265	56.8	286	67.0
QE	195	33.3	265	72.6	286	76.8
Ours	195	**55.3**	265	**75.2**	286	**78.0**

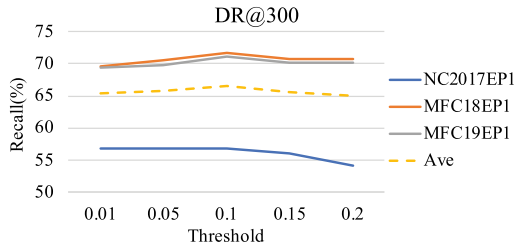


Fig. 6. DR@300 when using different percentages of changed pixels to determine the potential spliced region.

D. Provenance Filtering

The performance of the provenance filtering algorithms in terms of R@200 and R@300 are shown in Table I. We also report performance for different image types in the provenance graph. For each query image, the base image is the source image of the background, the donor images are the source images of the spliced regions and the intermediate images are other images generated throughout the image manipulation process. The recall of base, donor and intermediate images of the top 300 retrieved images are reported as Recall of Base Images (BR), Recall of Donor Images (DR), and Recall of Intermediate Images (IR), respectively. See Fig. 1 as examples of base, donor, and intermediate nodes.

The proposed method clearly outperforms the previous state-of-the-art method [11]. The query expansion helps improve the SIFT local feature baseline, while our newly proposed splice detection further improves the performance. As discussed earlier, retrieving the donor image is harder than retrieving the base and intermediate images, as the donor image may only contribute a small region to the probe image. The results also support this observation, confirmed by BR and IR are much higher than DR, especially on the MFC18EP1 and MFC19EP1 datasets. Our method achieves significant performance gain in terms of Donor Recall (DR), thanks to the spliced region detection based search which is designed to solve this problem specifically.

To further analyze the performance for retrieving the donor image, the recall for donor image is further broken down based on the size of the spliced region in the probe image. If the spliced region in the probe image is smaller than 1% of the total area of the probe image, the source image of the spliced region is called a Small Donor (SD). If the ratio is larger than 10%, the source is called a Large Donor (LD). Otherwise, it is called a Medium Donor (MD). The number of small, medium and large donor images (#SD, #MD and #LD, respectively) as well as the recall in the top 300 images (SDR, MDR and LDR, respectively) are shown in Table II.

In general, the smaller the spliced region, the harder it is to retrieve the source. One can expect to have SDR<MDR<LDR, which is observed for all methods on all datasets with the only exception of NC2017EP1 dataset, where MDR>LDR. The reason may be the data bias in this small dataset. The proposed method shows excellent performance for retrieving the source of small spliced region, which is one of the most challenging problems in provenance filtering. Compared to query expansion, the proposed method improves SDR by 23.9% in absolute values (a relative improvement of 70%) in NC2017EP1 dataset, 22.8% (65% relative) in MFC18EP1 dataset and 22.0% (66% relative) in MFC19EP1 dataset. The performance gain is clearly due to the proposed spliced region based search.

We further investigate how the final performance is changed when using different percentages of changed pixels to determine the potential spliced region. We use $[1\%, 5\%, 10\%, 15\%, 20\%]$ for the experiment and the result is shown in Fig. 6. Using different thresholds gives us very similar performance, which means the method is rather robust to the choice of this value. Choosing the value to 10% performs slightly better than others.

E. Oracle Provenance Graph Building

To test the performance of the proposed graph building algorithm, we first run an oracle graph building experiment, in which the input is a clean and complete set of images which compose a provenance graph. Since the input is a clean and complete set, we only evaluate the source recall (SR), the source overlap (SO) and the edge overlap (EO), *i.e.* we evaluate the source identification and edge construction.

The graph building performance of all baselines and the proposed method are shown in Table III. The proposed algorithm is a clear winner over the other state-of-the-art methods. In the NC2017EP1 and the MFC18EP1 datasets, the performance of the proposed algorithm is about twice the performance of [11] in terms of edge overlap. In the MFC19EP1 dataset, the

TABLE III
ORACLE GRAPH BUILDING RESULT (%)

	SR	SO	EO
	NC2017EP1		
[11]	N/A	N/A	25.3
Holistic	50.4	50.3	43.1
Pairwise	63.0	63.4	49.8
Ours	**72.0**	**72.7**	**51.0**
	MFC18EP1		
[11]	N/A	N/A	25.9
Holistic	54.7	55.2	45.8
Pairwise	58.4	59.5	46.8
Ours	**65.8**	**67.0**	**49.1**
	MFC19EP1		
[11]	N/A	N/A	37.0
Holistic	56.1	56.2	50.1
Pairwise	69.8	69.9	53.7
Ours	**77.6**	**77.8**	**56.8**
	Reddit		
[11]	N/A	N/A	3.7
[10] Meta*	N/A	N/A	3.4
[10] Meta+Visual*	N/A	N/A	8.5
Holistic	8.19	4.46	**21.6**
Pairwise	10.5	5.58	21.1
Ours	**12.9**	**7.09**	21.3

*Using metadata.

TABLE IV
END-TO-END GRAPH BUILDING RESULT (%)

	SR	SO	VR	VO	EO	VEO
	NC2017EP1					
[11]	N/A	N/A	N/A	**70.3**	24.2	46.8
Holistic	19.7	24.4	59.6	68.0	26.9	47.8
Pairwise	23.4	29.7	59.6	68.0	28.9	48.7
Ours	**26.7**	**33.3**	59.6	68.0	**29.6**	**49.0**
	MFC18EP1					
[11]	N/A	N/A	N/A	79.8	27.2	54.3
Holistic	30.0	31.7	82.4	**82.2**	37.2	60.0
Pairwise	36.4	38.2	82.4	**82.2**	37.9	60.4
Ours	**42.4**	**45.2**	82.4	**82.2**	**39.9**	**61.3**
	MFC19EP1					
[11]	N/A	N/A	N/A	70.3	29.5	51.9
Holistic	39.7	41.1	85.5	**85.2**	**42.5**	**65.2**
Pairwise	44.2	46.2	85.5	**85.2**	42.0	65.0
Ours	**46.8**	**49.5**	85.5	**85.2**	42.0	65.0

relative performance gain is 54%. In the Reddit dataset, our performance is around 5 times better than [11], which is similar to our method only using visual information for graph building. Even compared to methods which incorporate metadata [10], our method still shows a tremendous relative improvement of 150%. These numbers consistently show the very strong performance of the proposed graph building algorithm.

In terms of source detection, although the holistic detector is one of the state-of-the-art image manipulation detectors, its performance is lower than the pairwise method. The reasons are that the integrity score in the holistic detector may not be very reliable, and the holistic detector may be not sensitive enough to detect small and local image manipulations. The proposed pairwise ancestor-offspring detector is able to consider two images and focus on the changed area. We conjecture that's the reason why the pairwise method achieves a better performance than the holistic detector. The integration of these two scores yields the best result, indicating that the two detectors are complementary.

The performance on the Reddit dataset is lower than that seen on the Medifor datasets. The reasons are the genealogy graphs in the Reddit dataset are much larger (generally more than 40 images) than those in the Medifor datasets, and the holistic detector and the pairwise detector are trained on the Medifor development dataset. In addition, the Reddit dataset is a real-world dataset with lots of missing intermediate nodes.

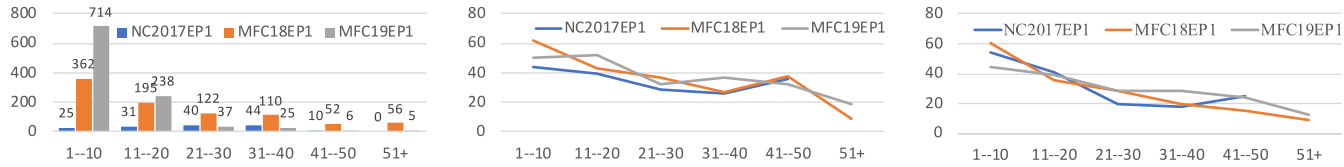
F. End-to-End Provenance Graph Building

The end-to-end provenance graph building evaluates the performance of the entire provenance analysis system. Various performance measurements of the final reconstructed graph are given in Table IV. As shown in Sec. V-B2, the evaluation metrics are the recall and overlap of source vertices (SR and SO), all vertices (VR and VO), overlap of the edges (EO) and both vertices and edges (VEO). The overall conclusion is very similar to what we got in Sec. V-E. The proposed method is again a clear winner over the previous state-of-the-art method. In terms of source recall and overlap (SR and SO), our fused method achieves significant gain over the holistic or pairwise method alone, showing one more time that the two approaches are complementary. For the edge overlap (EO), the improvements in NC2017EP1, MFC18EP1 and MFC19EP1 are 5.4% (22.3% relative), 12.7% (46.7% relative) and 12.5% (42.4% relative). Since images in the provenance graph may not be correctly retrieved and distractors may be incorrectly retrieved by the filtering algorithm, the end-to-end graph building problem is undoubtedly a more challenging problem than the oracle one. There is a clear performance gap between the oracle graph building (Table III) and the end-to-end graph building (Table IV).

Fig. 7 shows performance variation with respect to the size of the provenance graph. Intuitively, if there are more images in the provenance graph, it is harder to find out which image is the source (SO) and how the images are connected to each other (EO). We break down the graph building performance by the number of images in the ground-truth provenance graph, at every 10 images as an interval. The histogram of the provenance graph size is shown in Fig. 7a. The average SO and EO for different sizes of graph are shown in Fig. 7b and Fig. 7c, respectively. Overall, Fig. 7 shows the proposed algorithm performs very well with small graphs (≤ 10 images). However, with the increasing of the number of nodes in the graph, the performance drops significantly.

G. Qualitative Result

An automatically reconstructed provenance graph under the end-to-end setting is shown in Fig. 8. The green color means true positive, red means false positive, grey means false negative, the rectangle shape means vertex, circle means source, and an arrow corresponds to an edge. The proposed algorithm reconstructs the middle and bottom parts correctly. In the region highlighted by the blue dashed box, the proposed algorithm is able to identify the spliced car and the removed crosswalk. In the last two images in the highlighted region, a white car is duplicated on the road curb, as highlighted in the yellow boxes. The proposed algorithm



(a) Number of Graph (b) Source Overlap Score (c) Edge Overlap Score

Fig. 7. End-to-end provenance analysis performance break-down by the size of the provenance graph (number of images in the graph). (a)–(c) show the number of graph, source overlap score and edge overlap score with different sizes of graph respectively.

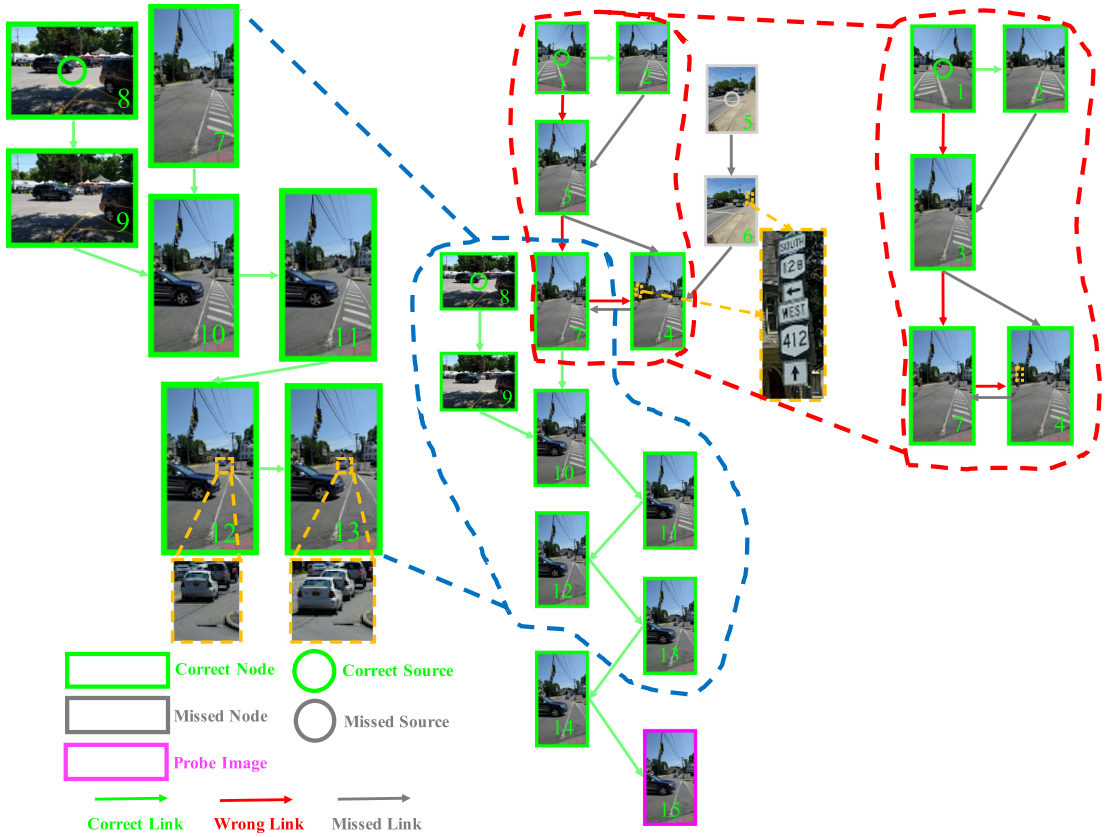


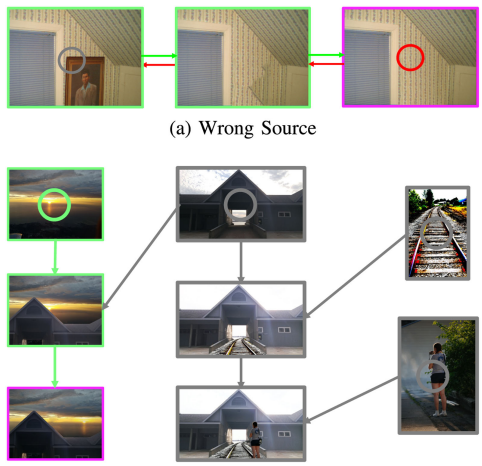
Fig. 8. Example of a reconstructed provenance graph. See details in Sec. V-G.

is able to capture such a tiny change and correctly recover the order of the manipulation.

The proposed algorithm fails to retrieve the source of the inserted road sign as shown in Fig. 8. This introduces some errors at the top of the graph (the red dashed box). We also show 2 more failed cases in Fig. 9. For Fig. 9a, the algorithm gets the source wrong, thinking the image without the portrait is the source and the portrait is added. However, the history shows the image with the portrait is the original one and the portrait is removed by the editor. For Fig. 9b, since the spliced house is too dark, the algorithm failed to find the house image and all other images related to it, including the spliced rail track and the spliced tourist.

H. Computation Time

The experiments are carried out on a computer with 32 CPU processors, 384GB RAM and 4 Nvidia 1080Ti GPUs. For



(a) Wrong Source

(b) Missing Donors and Related Images

Fig. 9. Two failed graph building results. See legend in Fig. 3. See details in Sec. V-G. Results are simplified to show the error.

MFC19EP1 dataset, with 2,000,000 world images and about 1,000 probe images, extracting local features takes about two days. The total number for local descriptors is about four billion. Building search index takes a day. Searching all probe images takes about a day, on average 1.5 minutes per probe. The search time for one probe includes the search of the original probe image, all expanded images, and the region-based search. The search of one single image takes around 10-20 seconds. The graph building step takes about two days. We apply 32 CPUs for graph building in parallel. Based on the size of the provenance graph, the graph building time on a single CPU processor can differ from a few minutes (when dealing with graphs of less than 10 images) to a few hours (for graphs of more than 50 images). In particular, calculating the similarity matrix (Eq. (4)) takes around 60% of the time. Pairwise relation calculation takes around 30%. Holistic detector and final graph construction take 5% each.

VI. Conclusion

Provenance analysis is a critical step in understanding the history and potential purposes of manipulations applied to a set of images. In this paper, we have proposed algorithms and a system to address the major challenges in provenance analysis, including retrieving the source images of small spliced regions and detecting the original/source images from a set of retrieved images. In provenance filtering, we have studied the similar image pairs for spliced area detection and used local features from the potential spliced region for image retrieval. In provenance graph building, we have further explored the pairwise relation by learning a pairwise ancestor-offspring detector. By combining the pairwise relation with the state-of-the-art image integrity detector, the proposed algorithm has achieved remarkable performance gain in source detection. The proposed provenance analysis system outperforms the state-of-the-art provenance analysis system by a very large margin. In the real-world Reddit dataset, the edge overlap of the proposed system is five times better than the state-of-the-art system. However, as shown in the experiment part, 1) the recall for the small donor image is low, and, 2) the source recall and overlap are still very low, especially for the real-world Reddit dataset. Our future work will focus on developing robust local feature and accurate search algorithm for retrieving small donors, and improving the accuracy for detecting the source image. To use the reconstructed graphs to understand the intent of the image manipulations applied to various real-world application settings such as news and social media is also an important future research direction.

Acknowledgment

Any opinions, findings and conclusions or recommendations expressed in this material are solely the responsibility of the authors and do not necessarily represent the official views of AFRL, DARPA, or the U.S. Government.

We also thank Daniel Moreira from the University of Notre Dame for sharing the performance data of their algorithms on the Medifor datasets.

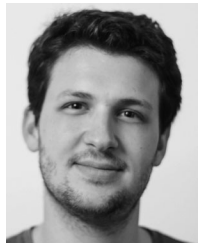
References

[1] H. Farid, *Photo Forensics*. Cambridge, MA, USA: The MIT Press, 2016.
[2] L. Kennedy and S.-F. Chang, "Internet image archaeology: Automatically tracing the manipulation history of photographs on the web," in *Proc. ACM Int. Conf. Multimedia*, 2008, pp. 349–358.
[3] Z. Dias, A. Rocha, and S. Goldenstein, "Image phylogeny by minimal spanning trees," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 2, pp. 774–788, Apr. 2012.
[4] Z. Dias, S. Goldenstein, and A. Rocha, "Exploring heuristic and optimum branching algorithms for image phylogeny," *J. Visual Commun. Image Representation*, vol. 24, no. 7, pp. 1124–1134, Oct. 2013.
[5] F. Costa, A. Oliveira, P. Ferrara, Z. Dias, S. Goldenstein, and A. Rocha, "New dissimilarity measures for image phylogeny reconstruction," *Pattern Anal. Appl.*, vol. 20, no. 4, pp. 1289–1305, 2017.
[6] Z. Dias, S. Goldenstein, and A. Rocha, "Toward image phylogeny forests: Automatically recovering semantically similar image relationships," *Forensic Sci. Int.*, vol. 231, no. 1-3, pp. 178–189, Sep. 2013.
[7] A. Oliveira *et al.*, "Multiple parenting identification in image phylogeny," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 5347–5351.
[8] A. A. d. Oliveira *et al.*, "Multiple parenting phylogeny relationships in digital images," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 2, pp. 328–343, Feb. 2016.
[9] A. Bharati *et al.*, "U-Phylogeny: Undirected provenance graph construction in the wild," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2017, pp. 1517–1521.
[10] A. Bharati *et al.*, "Beyond pixels: Image provenance analysis leveraging metadata," in *Proc. IEEE Winter Conf. Appl. Comput. Vision*, 2019, pp. 1692–1702.
[11] D. Moreira *et al.*, "Image provenance analysis at scale," *IEEE Trans. Image Process.*, vol. 27, no. 12, pp. 6109–6123, Dec. 2018.
[12] NIST, "Nimble Challenge 2017 Evaluation," Sep. 2016. [Online]. Available: https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation
[13] J. G. Fiscus *et al.*, "2018 MediFor Challenge," Jul. 2019. [Online]. Available: https://www.nist.gov/publications/2018-medifor-challenge
[14] NIST, " Media Forensics Challenge 2019," Nov. 2018. [Online]. Available: https://www.nist.gov/itl/iad/mig/media-forensics-challenge-2019-0
[15] H. Guan *et al.*, "MFC datasets: Large-scale benchmark datasets for media forensic challenge evaluation," in *Proc. IEEE Winter Appl. Comput. Vision Workshops*, Jan. 2019, pp. 63–72.
[16] Z. Dias, A. Rocha, and S. Goldenstein, "Video Phylogeny: Recovering near-duplicate video relationships," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2011, pp. 1–6.
[17] L. Xie, A. Natsev, J. R. Kender, M. Hill, and J. R. Smith, "Visual memes in social media: Tracking real-world news in YouTube videos," in *Proc. ACM Int. Conf. Multimedia*, 2011, pp. 53–62.
[18] B. Shen, C. W. Forstall, A. D. R. Rocha, and W. J. Scheirer, "Practical text phylogeny for real-world settings," *IEEE Access*, vol. 6, pp. 41 002–41 012, 2018.
[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[20] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks," in *Proc. Brit. Mach. Vision Conf.*, 2016, pp. 119.1–119.11.
[21] E. Mohedano, K. McGuinness, N. E. O'Connor, A. Salvador, F. Marques, and X. Giro-i Nieto, "Bags of local convolutional features for scalable instance search," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2016, pp. 327–331.
[22] A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson, "A baseline for visual instance retrieval with deep convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 251–258.
[23] F. Radenović, G. Tolias, and O. Chum, "Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 3–20.
[24] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2010.
[25] T. Ge, K. He, Q. Ke, and J. Sun, "Optimized product quantization for approximate nearest neighbor search," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2013, pp. 2946–2953.
[26] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8733051

[27] A. Babenko and V. Lempitsky, "Efficient indexing of billion-scale datasets of deep descriptors," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 2055–2063.
[28] G. Tolias and H. Jégou, "Visual query expansion with or without geometry: Refining local descriptors by feature aggregation," *Pattern Recognit.*, vol. 47, no. 10, pp. 3466–3476, 2014.
[29] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *Proc. IEEE Int. Conf. Comput. Vision*, 2007, pp. 1–8.
[30] A. Pinto *et al.*, "Provenance filtering for multimedia phylogeny," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2017, pp. 1502–1506.
[31] H. Farid, "Image forgery detection," *IEEE Signal Process. Mag.*, vol. 26, no. 2, pp. 16–25, Mar. 2009.
[32] I. Amerini, T. Uricchio, L. Ballan, and R. Caldelli, "Localization of jpeg double compression through multi-domain convolutional neural networks," in *Proc. IEEE CVPR Workshop Media Forensics*, 2017, vol. 3, pp. 1865–1871.
[33] L. Wen, H. Qi, and S. Lyu, "Contrast enhancement estimation for digital image forensics," *ACM Trans. Multimedia Comput., Commun. Appl.*, vol. 14, no. 2, pp. 1–21, 2018.
[34] W. Fan, S. Agarwal, and H. Farid, "Rebroadcast attacks: Defenses, reattacks, and redefenses," in *Proc. Eur. Signal Process. Conf.*, 2018, pp. 942–946.
[35] A. Sarkar, L. Nataraj, and B. S. Manjunath, "Detection of seam carving and localization of seam insertions in digital images," in *Proc. 11th ACM Workshop Multimedia Secur.*, 2009, pp. 107–116.
[36] Y. Wu, W. Abd-Almageed, and P. Natarajan, "BusterNet: Detecting copy-move image forgery with source/target localization," in *Proc. Eur. Conf. Comput. Vision*, 2018, pp. 168–184.
[37] S. Agarwal and H. Farid, "Photo forensics from JPEG dimples," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2017, pp. 1–6.
[38] D. Cozzolino, G. Poggi, and L. Verdoliva, "Splicebuster: A new blind image splicing detector," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2015, pp. 1–6.
[39] Z. H. Sun and A. Hoogs, "Object insertion and removal in images with mirror reflection," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2017, pp. 1–6.
[40] F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva, "Detection of GAN-generated fake images over social networks," in *Proc. IEEE Conf. Multimedia Inf. Process. Retrieval*, 2018, pp. 384–389.
[41] E. Sabir, W. AbdAlmageed, Y. Wu, and P. Natarajan, "Deep multimodal image-repurposing detection," in *Proc. ACM Multimedia Conf.*, 2018, pp. 1337–1345.
[42] Z. Sun and A. Hoogs, "Image manipulation detection and localization by residual classification," *IEEE Trans. Inf. Forensics Secur.*, under review.
[43] Y. Tian, B. Fan, and F. Wu, "L2-Net: Deep learning of discriminative patch descriptor in Euclidean space," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 6128–6136.
[44] J. Edmonds, "Optimum branchings," *J. Res. Nat. Bureau Standards B*, vol. 71, no. 4, pp. 233–240, 1967.
[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–14.

Xu Zhang received the Ph.D. degree from Tsinghua University, Beijing, China, in 2016. He is currently a Postdoctoral Research Scientist with Digital Video and MultiMedia Lab, Columbia University in the city of New York, New York, NY, USA, supervised by Prof. S.-F. Chang. His current research interests include local feature detection and description, large scale image retrieval, and image forensics.

Zhaohui H. Sun (Senior Member, IEEE) received the B.E. and M.E. degrees in electrical engineering and information science from the University of Science and Technology of China, Hefei, China, in 1992 and 1995, respectively, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Rochester, Rochester, NY, USA, in 1998 and 2000, respectively. He has been with Kitware, Inc., Clifton Park, NY, USA, since 2010. He was a Lead Computer Scientist with the Visualization and Computer Vision Laboratory, GE Global Research, Niskayuna, NY, USA, and a Principal Research Scientist with Eastman Kodak, Rochester, NY, USA. His current research interests include vision technologies, digital video/image analysis, and multimedia computing.



Svebor Karaman received the Ph.D. degree in computer science from the University of Bordeaux, Bordeaux, France. He is currently an Associate Research Scientist with the DVMM Lab, Columbia University, New York, NY, USA. His current research interests include computer vision and machine learning, more specifically video indexing, person re-identification, hashing and large-scale content based indexing.

Shih-Fu Chang (Fellow, IEEE) received the bachelor's degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1985, and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California at Berkeley, Berkeley, CA, USA, in 1991 and 1993, respectively. He is the Richard Dicker Professor in Electrical Engineering and Computer Science with Columbia University, New York, NY, USA. His research interests include computer vision, machine learning, and multimodal knowledge extraction. He was the recipient of the IEEE Kiyo Tomiyasu Award and Technical Achievement Awards from IEEE Signal Processing Society and ACM SIGMM. He is a Fellow of AAAS and ACM, and an elected Academician of Academia Sinica.