Multimed Tools Appl (2019) 78:6253–6275
https://doi.org/10.1007/s11042-018-6350-5

Multiple image copy detection and evolution visualisation using tree graphs

Mohand Said Allili[1] · **Nathalie Casemajor**[2] · **Aymen Talbi**[1]

Received: 8 September 2017 / Revised: 17 June 2018 / Accepted: 29 June 2018 /
Published online: 20 July 2018

Abstract Image copy detection is an important problem for several applications such as detecting forgery to enforce copyright protection and intellectual property. One of the important problems following copy detection, however, is the assessment of the type of modifications undergone by an original image to form its copies. In this work, we propose a method for quantifying some of these modifications when multiple copies of the same image are available. We also propose an algorithm to estimate temporal precedence between images (i.e., the order of creation of the copies). Using the estimated relations, a tree graph is then built to visualize the history of evolution of the original image into its copies. Our work is important for ensuring better interpretation of image copies after their detection. It also lays a new ground for enhancing image indexing, dissemination analysis and search on the Web.

Keywords Image copy detection · Image transformation · Copy evolution graph

✉ Mohand Said Allili
mohandsaid.allili@uqo.ca

Nathalie Casemajor
nathalie.casemajor@ucs.inrs.ca

Aymen Talbi
Tala07@uqo.ca

[1] Département d'informatique et d'ingénierie, Université du Québec en Outaouais, 101, St-Jean Bosco, Gatineau, QC, J8X 3X7, Canada

[2] Centre Urbanisation Culture Société, Institut national de la recherche scientifique, 385, rue Sherbrooke Est, Montreal, QC, H2X 1E3, Canada

1 Introduction

The growing popularity of Internet and social media has enabled ubiquitous and distributed sharing of digital photos among Internet users. This has been accompanied by new possibilities to easily copy, alter and distribute digital content to a large number of recipients thanks to the availability and accessibility of image processing software [24]. A huge challenge, therefore, arises for the ability of tracking and monitoring the evolution of original photos in the Web, in order to enforce intellectual property and copyright protection [5], for example. Tracking and visualizing copies can also be useful for analyzing and exploring the dissemination and use of photos in Web communities (e.g. arts, journalism) [9, 20, 22, 56]. Although several methods have been proposed in the past for image copy detection [5, 24, 29] and image collection visualization [35, 50, 51], not much research have been conducted for the purpose of tracing and/or visualizing the order of multiple image copies.

Early approaches for copy detection are based on watermarking which consists of embedding signatures (or watermarks) for copyright protection [13]. Thus, detecting copies amounts to identifying the watermark encoded in the image. However, these methods can be vulnerable since watermarks can be removed or altered via postprocessing techniques [25, 40]. Recently, content-based copy detection (CBCD) has been proposed as an alternative to watermarking for detecting (illegal) image copies [16, 29]. The goal of CBCD is to determine, using only the image content, whether near-replicas of a given image exist in the Web or through an unauthorized third party [5, 45, 52]. To detect possible copies, CBCD systems (e.g., TinEye and Piximilar [45]) extract image low-level features, followed by similarity measurement between images.

Copy detection systems are generally good at finding similarities between images and identifying identical content. However, they do not tell much about possible transformations operated on an image. Indeed, there is a large number of image manipulations, ranging from simple geometric/photometric transforms or resizing to more complex transforms, such as image editing, cropping, stitching and compression [41]. Moreover, even in the case of simple geometric transformations, for example, algorithms can fail to recover the exact transformation parameters when the image is significantly altered [44]. Recently, methods have been proposed for detecting and localizing specific image forgeries such as *image splicing* (IS) [10] and *copy-move* (CM) [12]. However, they are not able to recover the order of multiple image copies which may contain different types of modifications. Although having a unique algorithm to detect all types and order of transformations seems a very complex pursuit, one can make the problem more tractable by making assumptions about the generative process of copies. For example, grouping transformations into specific categories (e.g., photometric, geometric) [16] can facilitate investigation for searching potential manipulations operated on each copy.

In this paper, we propose a method for constructing an evolution graph for a set of image copies derived from the same original or a set of reference images determined a priori. Our method infers the most likely transformations used to produce the remaining image copies. For simplicity, we focus on mainly three types of image transformations in copy production: *geometric*, *photometric*, and *image editing*. *Geometric* transformations refer to changes due to affine transforms and image distortions [41]. *Photometric* transformations refer to image color enhancement, filtering and color-to-gray transforms [21]. Finally, *editing* refers to basic operations such as image cropping, copy-move, zooming, seam carving and text/object

insertion [17]. The identification of relations between images is based on the following steps:

- In the first step, we detect simple transformations such as image cropping, resizing, rotation, small illumination changes and color-to-gray. These transformations enable to build the first (strong) edges in our copy evolution graph.
- In the second step, we infer the remaining relations between images by detecting for each copy its most likely lineage in the graph. This step is performed by first building copy groups through agglomerative clustering and then using a combined local-global similarity measure to analyze image changes.
- Finally, all group subgraphs are linked to the main graph and each formed edge is annotated with its inferred transformation(s). Experiments conducted to validate to proposed approach have shown its performance to produce interpretable and meaningful graphs compared to manually constructed ones.

The remainder of this paper is organized as follows. Section 3 describes the proposed approach for identifying image transformations and evolution graph visualization of copies. Section 4 presents experimental results validating our approach. We end the paper with a conclusion and future work perspectives.

2 Related work

Since our work deals with detecting manipulations in image copies as well as copy ordering, it is important to give an overview of related work dealing with copy/forgery detection. Watermarking [31] and content-based copy detection (CBCD) [29] are the main methods used for searching image copies. For forgery detection, several methods have been proposed for specific image forgeries such as copy-move [12], image splicing/composites [5], image compression, and basic image processing operations such as illumination changes, rotation and cropping [16].

2.1 Image copy detection methods

Detecting image copies is important for several applications such as usage tracking and copyright protection. There are broadly two approaches for copy detection 1) watermarking and 2) content-based copy detection [28, 29]. Watermarking consists of embedding signatures (or watermarks) in images in the form of identification codes carrying information about a copyright owner. Watermarks can be inserted in the spatial, frequency and wavelet domains [13]. Therefore, detecting copies amounts to identifying the watermark encoded in images. The main limitation of these methods is their vulnerability to watermark removal or alteration using image processing techniques [25, 40]. Watermarking poses also an additional constraint since it must be inserted in the original image before its publication, sometimes at the time of its recording [16].
Content-based copy detection (CBCD) is a complementary approach to watermarking for detecting illegal image copies [16, 29]. It is based on the same principles as content-based image retrieval (CBIR) where systems extract signatures from an original image and each

image in a large database, which are compared to retrieve near-replicas [45, 52]. Image features capture visual properties of an image, either globally for the entire image (e.g., wavelets, discrete cosine transform (DCT), color, texture) or locally for a small group of pixels (e.g., shapes, contours, salient points). Global features give the image layout and the distribution of color/texture patterns. The overall image is thus represented by a vector of color/texture components such as wavelet distributions [1] and color histograms [2, 32]. Global features also generally fast similarity computation [1]. However, they may fail to identify important local visual characteristics, which can lead to a large number of false postives/negatives. Local features can be extracted at salient points [3, 58] or non-overlapping blocks and stacked into vectors [24, 29, 54]. Thus, depending on the scale of the key content or pattern, an appropriate representation should be chosen [14]. For example, [58] proposed a variant of the scale-invariant feature transform (SIFT) for fast copy/object detection in the presence of flipping transformations. DCT features have been used in [24, 29] to train classifiers for detecting copies produced by geometric manipulations and compression.

2.2 Image forgery detection methods

Image forgery detection aims at assessing the authenticity of a digital image by detecting and localizing potential manipulations. Following [16], forgery detection techniques can be broadly grouped into two main categories: 1) *statistical-based techniques*: study anomalies introduced at the pixel level via global transformations and lossy compression schemes, 2) *physically-based techniques*: detect anomalies by using models describing interaction between physical objects, light, and the camera. Models can also use measurements of objects in the world and their positions relative to the camera and account for artifacts introduced by the camera lens.

One of the main studied problems in the literature of image forgery detection is copy-move (CM) manipulation [59]. CM forgery consists of copying and pasting content within the same image, and potentially postprocessing it [4, 19]. It is usually done in order to hide certain details or to duplicate certain aspects of an image [5]. Proposed CM detection methods use either keypoint-based or or block-based matching techniques [12]. Keypoint-based methods extract image features at salient points such as SIFT [3, 39], which are then compared to the rest of the image to detect potential CM manipulations. Block-based methods subdivide an image into rectangular regions on which features such as color/intensity distribution [33], invariant moments [34], wavelet transform [36] or DCT coefficients [48, 54] are calculated. Most of these features require preprocessing steps (e.g., noise removal, converting color to intensity levels) to be used in the matching step.

Like CM forgery, image splicing (IS) is another simple and commonly used image tampering scheme in images [5, 16]. It consists of replacing image fragments using information from other images. Since CM and IS forgeries are similar, their detection can be addressed using the same tools. Authors in [37] developed an IS detection model based on the bicoherence magnitude and phase of the Fourier transform. In [10], camera response functions (CRF) are analyzed for efficient detection and localization of CM and IS manipulations. In [57], planar homography constraint is used to roughly localize tampered regions in images, followed by graph-cut segmentation for extracting fake objects. Worth mentioning are also methods proposed for automatic change detection between multi-temporal images of the same scene [27, 42]. The majority of these methods have been proposed for specific domains such as video surveillance [8, 38], remote sensing [23, 26] and medical diagnosis [6].

Most of the presented methods deal specifically with copy and/or forgery detection problems in single images. However, they can not be used for identifying relationships between multiple copies. Indeed, having an insight of the type and order of creation of copies can be useful for several high-level applications such as tracking copies for copyright protection [13], analyzing information dissemination and diffusion on the Web [20] and forecasting cascades of photo usage in Web communities [11, 30]. Our method addresses this problem by proposing an algorithm for estimating the lineage of multiple image copies and their order of creation. It recovers also the most likely transformations used for their generation. Finally, a visualization graph is constructed to give a visual insight about the evolution graph of the original image into its copies.

3 The proposed approach

Suppose that we have a set of n images $\mathcal{I} = \{I_1, ..., I_n\}$ representing different copies of an original image I_0. Our goal is to recover the order of creation of the copies in the form of a tree graph depicting the history of copy creation from I_0 to its descendent(s) and the lineage of each copy. Since an image can undergo arbitrary transformations, some of which are irreversible in their nature (e.g., image cropping, editing), it is hard to assess with certainty all types of transformations between two images in $\mathcal{I}$. However, an approximation of these transformations is still possible through a reverse-engineering-like approach using similarity measures between images. This will help understanding the generative process of related image copies and unravelling the most likely used transformations.

To facilitate problem formulation, and without loss of generality, we suppose that we have the following groups of transformations: *C: image cropping, E: edition, G: color to gray, L: illumination change, R: rotation, S: scale change*. To express that a copy I_j of an image I_i is generated using one of the transformations $T \in \{C, E, G, R, S, L\}$, we adopt the following notation: $I_j = T(I_i, \phi)$, where ϕ is the set of parameters used in transformation T. In case of an image rotation, for example, we have $\phi = \{\theta\}$ where θ is the rotation angle. Note that when there are multiple transformation candidates for explaining the creation of a particular copy, we use the *principle of least action* by selecting the simplest transformation among the candidates. For example, if we have two possible rotations to produce image I_C from images I_A and I_B, respectively, such that $I_C = R(I_A, \theta_1) = R(I_B, \theta_2)$ and $\theta_1 < \theta_2$, then we will suppose that I_C is produced by the rotation with the smallest angle, that is $I_A \rightarrow I_C$. Finally, since we can have several types of editions that can be carried out either at a local or global level, we use a subscript to indicate each type of edition that can be detected by our method:

- *Global editions:* refer to transformations affecting globally the visual aspect of the image, either by photometric (e.g., compression, noise, blurring, etc.) or geometric manipulations (e.g. image distortion, perspective transformation, etc.). Since it is difficult to discriminate between these types of editions, we denote them simply by the symbol E_G.
- *Local editions:* refer to transformations affecting locally the visual aspect of the image (e.g., text insertion/removal, object insertion/removal, border insertion, face blurring, etc). Our algorithm can detect three particular types of local editions, which are text, border and object insertion/removal, denoted by the symbols E_T, E_B and E_O, respectively.

3.1 Estimating strong relations between images

Our method starts by recovering relations with the strongest evidence among the images, and then completing the graph by inferring the rest of the relations. One of the strongest relations that can be estimated with high confidence among images are: *cropping (C), color to gray (G), rotations (R), illumination change (L) and scale changes (S)*. Indeed, these transformations do not produce major modifications in the image content which makes them relatively easy to detect through correlation analysis. More formally, we measure the normalized cross-correlation (NCC) [43] between all pairs of images. Given two gray-scale images of equal size, I_A and I_B, the NCC is given by the formula:

$$NCC(I_A, I_B) = \frac{\sum_{x,y} I_A(x, y) I_B(x, y)}{\sqrt{\sum_{x,y} I_A(x, y)^2 \sum_{x,y} I_B(x, y)^2}}, \tag{1}$$

where the summations are made over all the image coordinates. Note that $NCC(I_A, I_B) \in [0, 1]$, where $NCC(I_A, I_B) = 1$ if I_A and I_B are perfectly identical, and $NCC(I_A, I_B) \ll 1$, otherwise. Given two arbitrary images I_A and I_B in our set $\mathcal{I}$, with sizes let $H_A \times W_A$ and $H_B \times W_B$, respectively, the following relations can be asserted through NCC calculation:

- We consider I_B as a cropping of I_A, denoted by $I_B = C(I_A, \phi)$ with $\phi = \{x, y, H_B, W_B\}$, if there exist a sub-image I'_A of size $H_B \times W_B$ centered at location $(x, y) \in I_A$ where $NCC(I'_A, I_B) = 1$.
- We consider I_A as a rotation of I_B if there exist an angle θ where $NCC(R(I_A, \theta), I_B) = 1$, where $R(I_A, \theta)$ stands for a rotation of image I_A with an angle θ. For simplicity, we take the following angle values: $\theta = \frac{\delta\pi}{4}$ where $\delta \in \{1, ..., 8\}$.
- A scale change occurs between image I_A and I_B if there exists a factor $s \in \mathbb{R}^+$ with $NCC(S(I_A, s), I_B) \simeq 1$, where $S(I_A, s)$ is a scale change of image I_A with factor $s > 0$.
- We consider I_B as a grayscale version of I_A if $NCC(G(I_A), I_B) = 1$, where $G(I_A)$ is a grayscale transformation of image I_A obtained by averaging its RGB color channels.
- We consider I_B as a result of illumination change of of I_A, denoted by $I_B = L(I_A, \gamma)$ with γ is the parameter of the Gamma transform, if $NCC(I_A, I_B) = 1$ and there exist a parameter γ such that the histograms of $L(I_A, \gamma)$ and I_B are identical. Note that the negative of an image belongs also to this category of transformation.

To detect the above transformations, our algorithm tests each pair of images. We first test the occurrence of cropping at each image location using 8 values of rotation angles θ and 5 scales s, making up 40 tests at each location. To ensure fast calculation of the NCC, we use recursion as proposed in [46]. The NCC allows also to detect color-to-gray transform if one of the compared gray-scale images is originally in color. Finally, small illumination changes are detected by testing 5 values of γ. We then take one of the following decisions between each pair of images (I_A, I_B):

- A strong relation T is found if $NCC(T(I_A), I_B) \in [\delta_2, 1]$,
- A weak relation T is found if $NCC(T(I_A), I_B) \in [\delta_1, \delta_2]$,
- No relation is found, otherwise,

where $\delta_1, \delta_2 \in [0, 1]$ are two thresholds such that $\delta_1 < \delta_2$ (typically $\delta_2 = 0.99$ and $\delta_1 = 0.90$). Note that a strong relation implies automatically adding a strong edge to the evolution graph, but not a weak relation. Fig. 1 shows an example of strong relation detection, where image (b) results from a cropping of image (a) followed by a scale change. Images (c) and (d) show, respectively, the NCC map and the cropping location corresponding to the highest correlation value.

3.2 Image grouping for copy lineage detection

Once the strongest relations are detected in $\mathcal{I}$, we perform an agglomerative clustering on the images in $\mathcal{I}$. In this phase, two scenarios can be considered. In the first one, no reference images are available and a fully unsupervised grouping is then performed. In the second

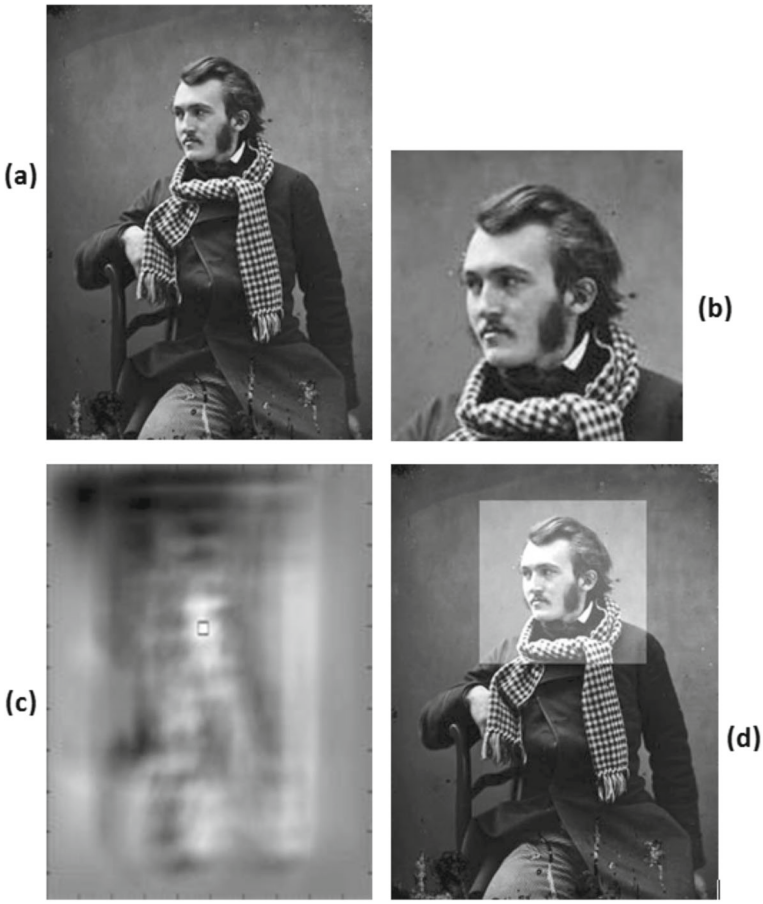


Fig. 1 Cropping detection using NCC. **a** represents the original image I_A, **b** represents a cropping of image I_A. **c** represents the value of NCC at different locations in I_A and **d** represents a highlight of the location of the cropping in image I_A

one, some reference images can be provided by an expert or found in official sites (e.g., museums, archives, art galleries). Each reference image is then used as centroid on which a cluster is built to constitute *lineages* in the final graph having reference image as a root.

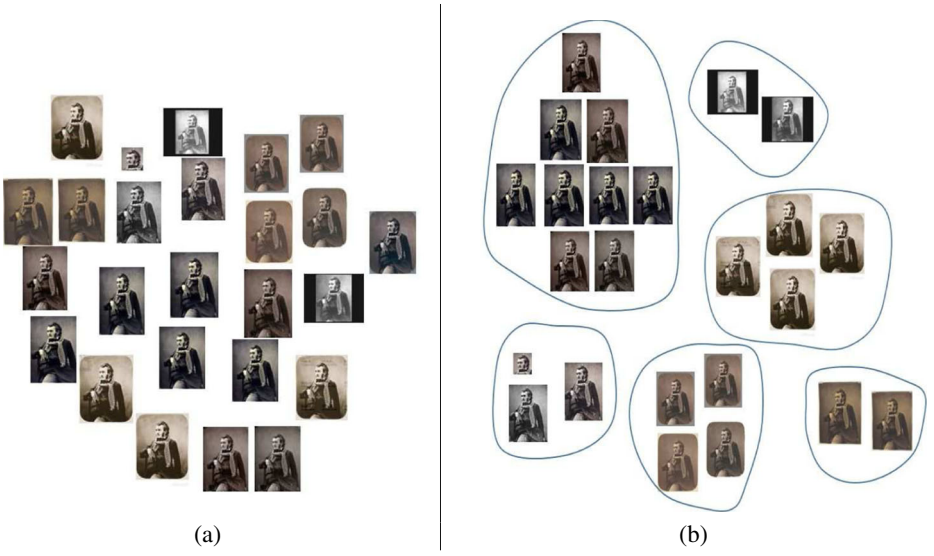
To group images into *lineage* clusters, we use image similarity based on a combination of strong relations detection and histogram comparison. Given two images I_A and I_B, we use the Bhattacharyya distance to measure the similarity between their histograms. Let H_A^j and H_B^j be the histograms of images I_A and I_B at color channel $j \in \{R, G, B\}$, and N_{bins} is the number of bins in the histograms. The histogram similarity between images I_A and I_B is given by:

$$S_H(I_A, I_B) = \sum_{j \in \{R,G,B\}} \frac{B(H_A^j, H_B^j)}{3} \tag{2}$$

where $B(H_A^j, H_B^j) = \sum_{i=1}^{N_{bins}} \sqrt{H_A^j(i) \cdot H_B^j(i)}$. The clustering is performed according to the following similarity measurement and the single link method to calculate distance between intermediary groups for cluster fusion [15]. Finally, distance between two images is taken as follows:

$$d(I_A, I_B) \begin{cases} 0 & \text{if } \exists \text{ strong relation} : I_A \rightarrow I_B \\ 1 - S_H(I_A, I_B) & \text{otherwise} \end{cases} \tag{3}$$

where $I_A \rightarrow I_B$ means that a strong precedence relation has been already established between I_A and I_B as described in the previous section. The first condition in (3) is intended to prevent clustering errors related to effects of transformations such image cropping, where the histogram of the cropped image can be significantly different from the original image.



(a) (b)

Fig. 2 Illustration of image grouping by histogram comparison: **a** copy image set, **b** obtained copy groups constituting *lineage* clusters

(a) (b) (c) (d) (e)

Fig. 3 Detection of image editions by comparing local image structure: **a** original image, **b** edited image, **c** change map generated using (4), **d** binary segmentation of **c** and **e** Red/blue rectangles indicate the presence/absence of text using the method proposed [49] on each detected blob

Fig. 2 shows an example of clustering for images in our third data set using the proposed method. Clearly, images showing similarities of their content have been grouped into homogenous clusters that will constitute different lineages in the final graph.

3.3 Detecting image editions

Image editing refers to modifying a digital image by removing unwanted elements (e.g., scratches, face blurring, etc.), or inserting/removing elements such as objects and text. Local-level editing affects the image locally (e.g., text/object insertion/removal, scrach/red eye removal, border insertion, etc.). Global-level editing refers to transformations affecting all the pixels of the image (e.g., image distortion, compression, etc.). To discriminate between the different types of editions, we use appropriate features extracted from the image, followed by supervised classification.

Let $I_A, I_B \in \mathcal{I}$, where I_B is the result of editing I_A. To compare the two images I_A and I_B at a local level, we first align them using [44]. Then, a change map B is calculated for I_A by comparing its local structure with I_B using a local image similarity metric inspired from [47]:

$$B(\mathbf{x}) = 1 - \max_{\mathbf{x}' \in N(\mathbf{x})} \left(\frac{[2\mu_A(\mathbf{x})\mu_B(\mathbf{x}') + c_1][2\sigma_{AB}(\mathbf{x}, \mathbf{x}') + c_2]}{[\mu_A^2(\mathbf{x})\mu_B^2(\mathbf{x}') + c_1][\sigma_A^2(\mathbf{x})\sigma_B^2(\mathbf{x}') + c_2]} \right) \tag{4}$$

where $N(\mathbf{x})$ stands for the neighbourhood of location $\mathbf{x} = (x, y)$. The parameters $(\mu_A(\mathbf{x}), \sigma_A(\mathbf{x}))$ and $(\mu_B(\mathbf{x}'), \sigma_B(\mathbf{x}'))$ represent local mean and standard deviation at locations $\mathbf{x}$ and $\mathbf{x}'$ in images I_A and I_B, respectively. The constants c_1 and c_2 are used to stabilize the division with a weak denominator. Finally, the maximum operator around a location neighborhood is introduced to make the measure more robust to misalignments and slight image shifts.

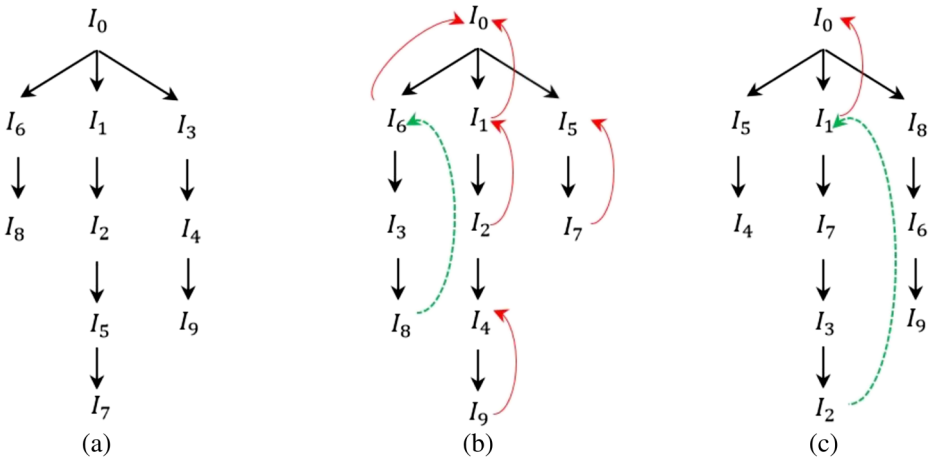


Fig. 4 Example of graph similarity measurement based on one-level and multi-level parent search: **a** represents the ground-truth, **b** and **c** represent two examples of constructed graphs. Red (continuous) and green (dashed) arrows are examples of one-level and multi-level search, respectively

We generate a map using (4) as illustrated in Fig. 3c, which is segmented to a binary image using [7] (see Fig. 3d). In the case that no blob is found, we consider that the image is edition-free. Otherwise, we extract the most important blobs from the binary map and analyze four of their geometric properties: *Compactness*, *relative size*, *dispersion* and *precence/absence of text*. *Compactness* measures the solidity of each detected blob. Local editions such text and object insertion/removal generally produce blobs with higher compactness than global manipulations such as distortions and compression. *Relative size* and *dispersion* indicate if detected blobs are local or scattered around the whole image. Based on the these features, we train an SVM classifier to detect wether a generated map stems from local or global edition. When local edition is detected, a second round of classification starts by analyzing first the occurrence of text in each blob using the method in [49]. If not text is detected, another SVM classifier is trained on the geometrical features of each blob to classify it as either an object E_O or border E_B insertion/removal. To train our classifiers, we use object examples from the Caltech dataset [18]. For examples of blobs occurring in global editions and border insertions, we collected images from the Web and added manually-generated ones[1].

Figure 3 shows two examples of image editions. The first row shows local text insertions detected on some blobs of the change map using [49]. We can see in column (e) the regions detected as text insertions (red rectangles) and others detected falsely as non-text editions (blue rectangles). In the second row, we show an example of image distortion that generated non-compact and scattered blobs which have been classified as a global edition by our method.

[1] http://w3.uqo.ca/allimo01/doc/editions.rar

3.4 Copy ordering and evolution graph construction

In this step, we build an evolution graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ for our image set $\mathcal{I}$ with vertices $\mathcal{V}$ representing images in $\mathcal{I}$ and oriented edges $\mathcal{E}$ representing transformations performed on images to create copies (graph descendants). Given the resulting clustering groups, a sub-graph is constructed within each group by incrementally ordering the group images based on their similarity. Algorithm 1 shows the script for building the sub-graph within each group of image copies. The algorithm is designed for the case of one available reference image I_0, but it can be run iteratively in case of multiple image references (see Fig. 4 for illustration).

The algorithm is composed of two main steps. In the first step (see lines 6 to 13), strong relations are detected in each group G_v, $v = 1, ..., K$, which are used to establish the first oriented edges in the graph. In the second step (see lines 14 to 28), starting from the group root r_v, vertices and edges are added incrementally to the graph by inferring the most likely transformations operated on the leaf images already added to the graph. To establish the next edge, a similarly measurement is used which combines local and global information of images. Global information consist in measuring distance between image histograms $S_H(I_A, I_B)$ as formulated using (2). Local information is measured by averaging the value of (4) in all image locations $S_L(I_A, I_B) = \frac{1}{|I_A|} \sum_{\mathbf{x} \in I_A} B(\mathbf{x})$, where $|I_A|$ represents the number of pixels in I_A. Finally, the combined image similarity measure is calculated using the following formula:

$$S(I_A, I_B) = \alpha S_H(I_A, I_B) + (1 - \alpha) S_L(I_A, I_B), \tag{5}$$

where the parameter α balances the contribution of local and global information ($\alpha = 0.5$ is used as a default value). It is clear that $S(I_A, I_B) \in [0, 1]$, where 1 designate a perfect match between the two images I_A and I_B. The process of graph completion constitutes the second part of the algorithm (see lines 14 to 28). Let G_c be the set of images in group G_v that are added to the graph, and its complement $\bar{G}_c$ defined by $\bar{G}_c = G_v \setminus G_c$. For each iteration, we use (5) to select the next link to add to the graph between images $I_{i_1} \in G_c$ and $I_{i_2} \in \bar{G}_c$ such that:

$$[I_{i_2}, I_{i_2}] = \arg \max_{I_j \in G_c, I_k \in \bar{G}_c} S(I_j, I_k). \tag{6}$$

Then, an edge $I_{i_1} \rightarrow I_{i_2}$ is added to the subgraph of G_c and we remove I_{i_2} and all its descendants from $\bar{G}_c$. The edge is annotated by first testing if an edition has occurred between I_{i_1} and I_{i_2} using (4), and use our trained SVM classifier to identify the type of edition. We also test the occurrence of weak relations $\{L, S, G, R, C\}$ when $NCC \in [\delta_1, \delta_2]$ as discussed in Section 3.1. Note that since line 10 of the algorithm builds strong edges between images, adding a parent image to the set G_c will incur adding all its descendants to G_c (see lines 22 to 25). This process ends for the group when $\bar{G}_c = \emptyset$ and G_c is completely ordered, and we carry out the same procedure for all formed groups. Finally, all resulted subgraphs are annotated and added to the main graph $\mathcal{V}$.

Algorithm 1 Algorithm for constructing copy evolution graph

Data: K copy clusters
Result: Copy evolution graph

$\mathcal{V} \leftarrow \{I_0\}$; $\mathcal{E} \leftarrow \emptyset$;
for $v = 1 \rightarrow K$ **do**
  Let r_v be the root of group G_v;
  $\mathcal{E} \leftarrow \mathcal{E} \cup \{I_0 \rightarrow r_v\}$;
  $G_c \leftarrow \{r_v\}$; $\bar{G}_c \leftarrow G_v \setminus G_c$;
  if $|G_v| > 1$ **then**
    for $i = 1 \rightarrow |G_v|$ **do**
      for $j = 1 \rightarrow |G_v|$; $j \neq i$ **do**
        if $\exists T \in \{R, S, G, C, L\}$ with parameter ϕ such that $I_j = T(I_i, \phi)$ **then**
          $\mathcal{E} \leftarrow \mathcal{E} \cup \{I_i \rightarrow I_j\}$; // Strong relation
        end if
      end for
    end for
    while $|G_c| < |G_v|$ **do**
      Set matrix M of size $|G_c| \times |\bar{G}_c|$;
      for $j = 1 \rightarrow |G_c|$ **do**
        for $k = 1 \rightarrow |\bar{G}_c|$ **do**
          $\mathbf{M}(j, k) \leftarrow S(I_j, I_k)$ using (5), where $I_j \in G_c$ and $I_k \in \bar{G}_c$;
        end for
      end for
      $[i_1 \ \ i_2] \leftarrow \arg\max_{(j,k;\, j \neq k)}(\mathbf{M})$; // Graph completion
      $\mathcal{E} \leftarrow \mathcal{E} \cup \{I_{i_1} \rightarrow I_{i_2}\}$;
      $D \leftarrow$ descendants of I_{i_2};
      Test for editions and weak relations between I_{i_1} and I_{i_2};
      $G_c \leftarrow G_c \cup \{I_{i_2}\} \cup D$;
      $\bar{G}_c \leftarrow \bar{G}_c \setminus (\{I_{i_2}\} \cup D)$;
    end while
  end if
  $\mathcal{V} \leftarrow \mathcal{V} \cup G_c$;
end for
Annotate all the edges of the graph;

4 Experiments

To validate the proposed approach, experiments have been conducted on four datasets. The first two datasets contain 32 image copies each, which are generated manually by carrying out a series of transformations to the original *Mona Lisa* painting image. Figure 5a and b show the images of the datasets. The third dataset contains 31 copies of a *van Gogh* painting depicting a landscape scene, which are generated in the same way as the first two datasets. The fourth dataset has been collected from the Web (see Fig. 10) and contains 53 copies of a photograph by famous French artist *Nadar*. The public archive that owns the negative

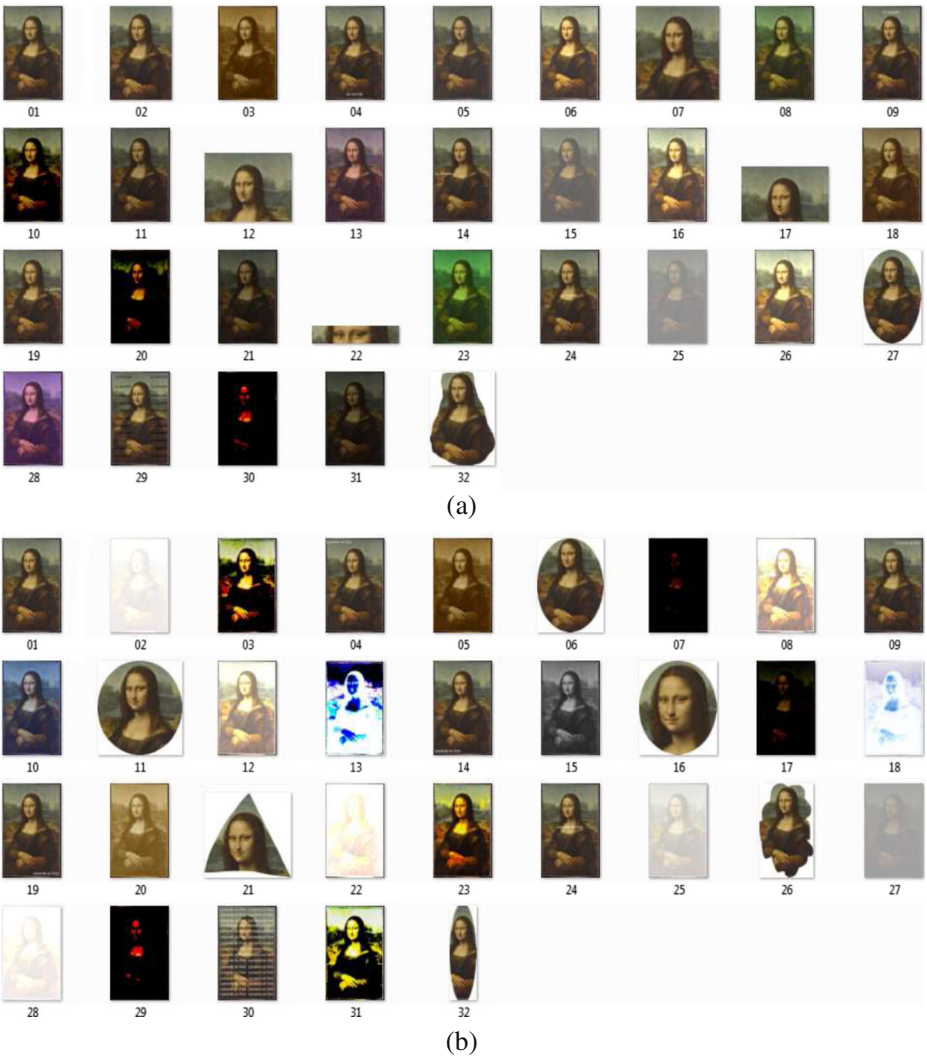


Fig. 5 Copies of *Mona Lisa* image constituting our two first datasets **a** and **b**

Table 1 Values of our evaluation metric on each reconstructed graph from the four datasets

Number of levels	Dataset I	Dataset II	Dataset III	Dataset IV without references	Dataset IV with references
One-level	88%	91%	93%	73%	82%
Multi-level	92%	95%	93%	79%	89%

The first and second rows show values obtained using one-level and multi-level parent search in the graph, respectively

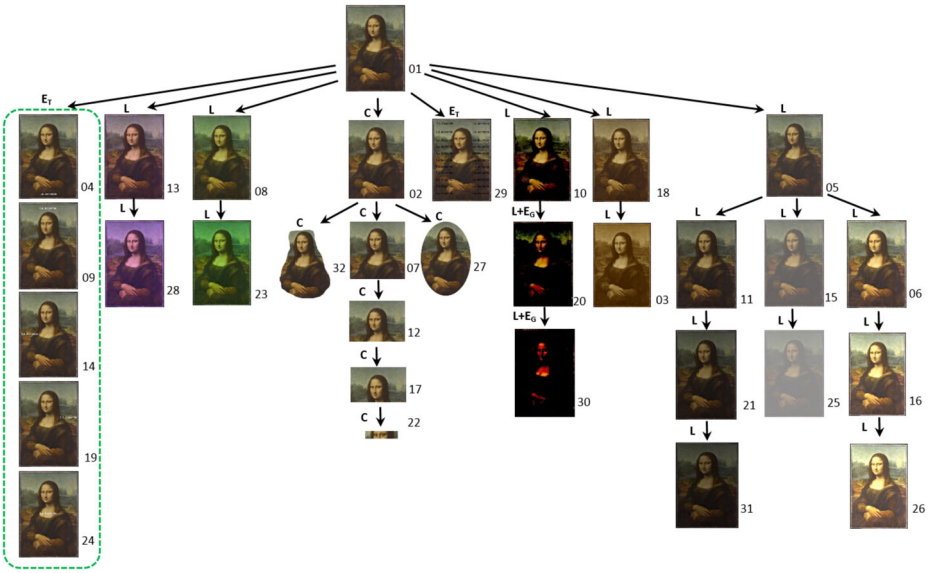


Fig. 6 Constructed copy graph of the second dataset in Fig. 5a

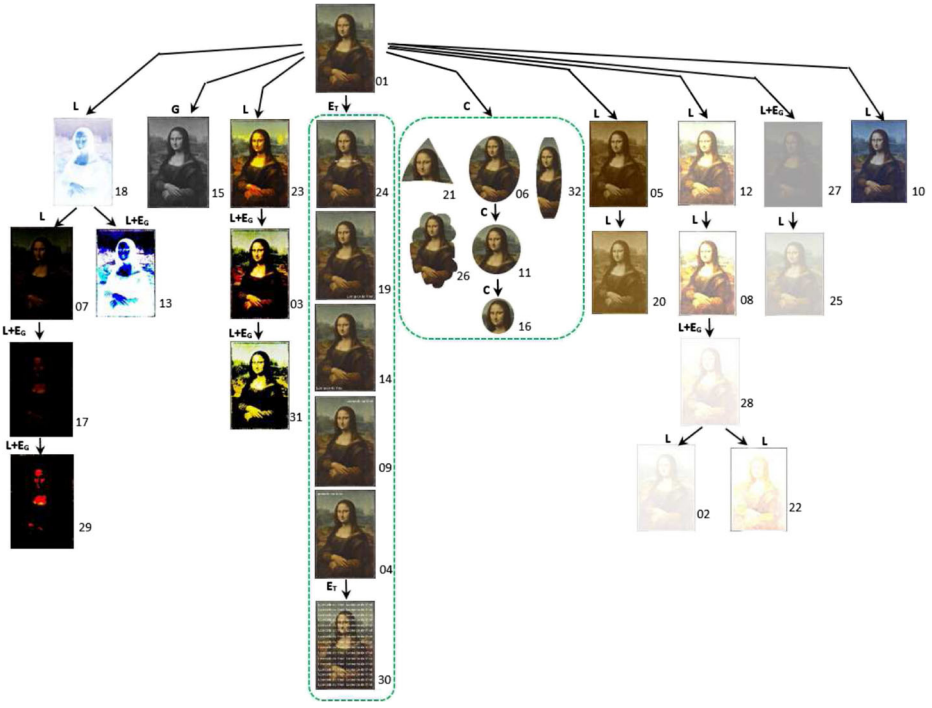


Fig. 7 Constructed copy graph of the second dataset in Fig. 5b

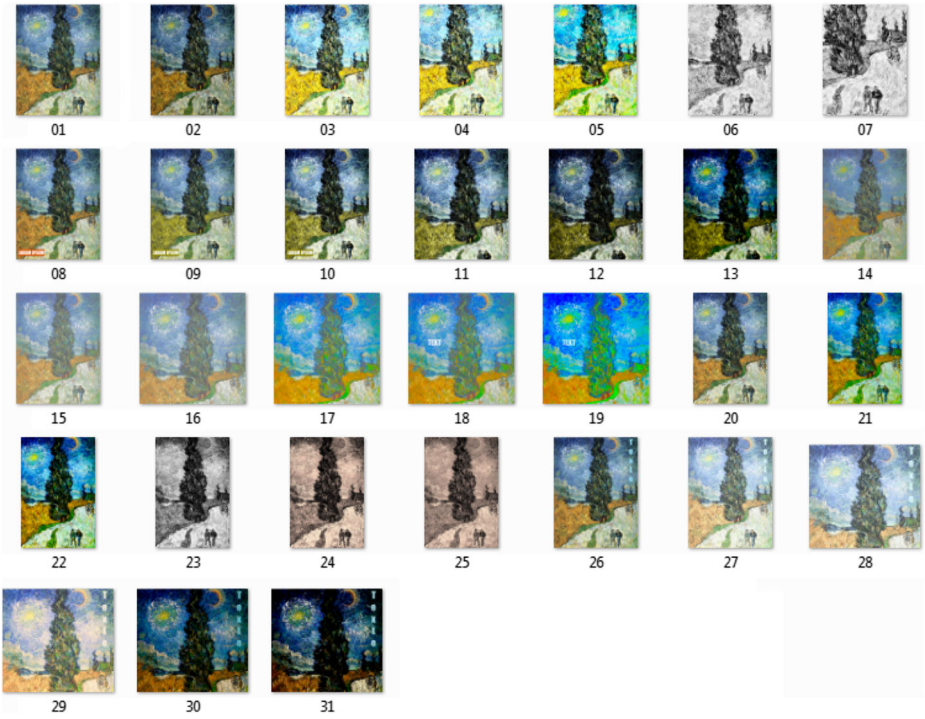


Fig. 8 Copies of a *Van Gogh* painting constituting the third dataset

has published online several scans of the image. Other institutions that own paper prints (museums, archives, libraries and auction houses) also published digital copies online, and other copies were found on blogs, media Web sites and on Wikipedia using Google search. While the first three datasets have all information about transformations used for copies generation, the fourth dataset has gaps: some roots and descendants may be missing, which is normal case when copies are largely collected from the Web.

To evaluate the accuracy of constructed graphs, we compared them to the ground truth (i.e., a reference graph) when available (datasets I and II) or to a graph created by an expert (dataset III) based on qualitative analysis of the visual content of the images and the context of their publication on the Web (e.g., authority of the Web sites, mention of source, presence of hyperlink, etc.) To measure similarity between graphs, we must take into account the following specificities. Since we are only interested in the temporal order of the copies, horizontal order of vertices within the same level is not important. However, the order of vertices in each vertical path from the root is important since it shows the lineage of each generated copy from its ancestors. Therefore, we cannot use directly existing methods for comparing tree graphs since most of them are dedicated to binary ordered trees [53, 55]. Instead, we propose a new measure as follows:

Given an image $I_i \neq I_0 \in \mathcal{I}$, let $P(I_i)$ and $\hat{P}(I_i)$ (resp. $A(I_i)$ and $\hat{A}(I_i)$) be its parent vertices (resp. set of ancestor vertices) in the reference and the reconstructed graphs, respectively. In a one-level search, we consider that the parent of I_i is correctly identified if $\hat{P}(I_i) = P(I_i)$ for which we assign a score $s_i = 1$, and $s_i = 0$ otherwise. In a multi-level search, a parent of I_i is correctly identified $\hat{P}(I_i) = P(I_i)$, for which we assign

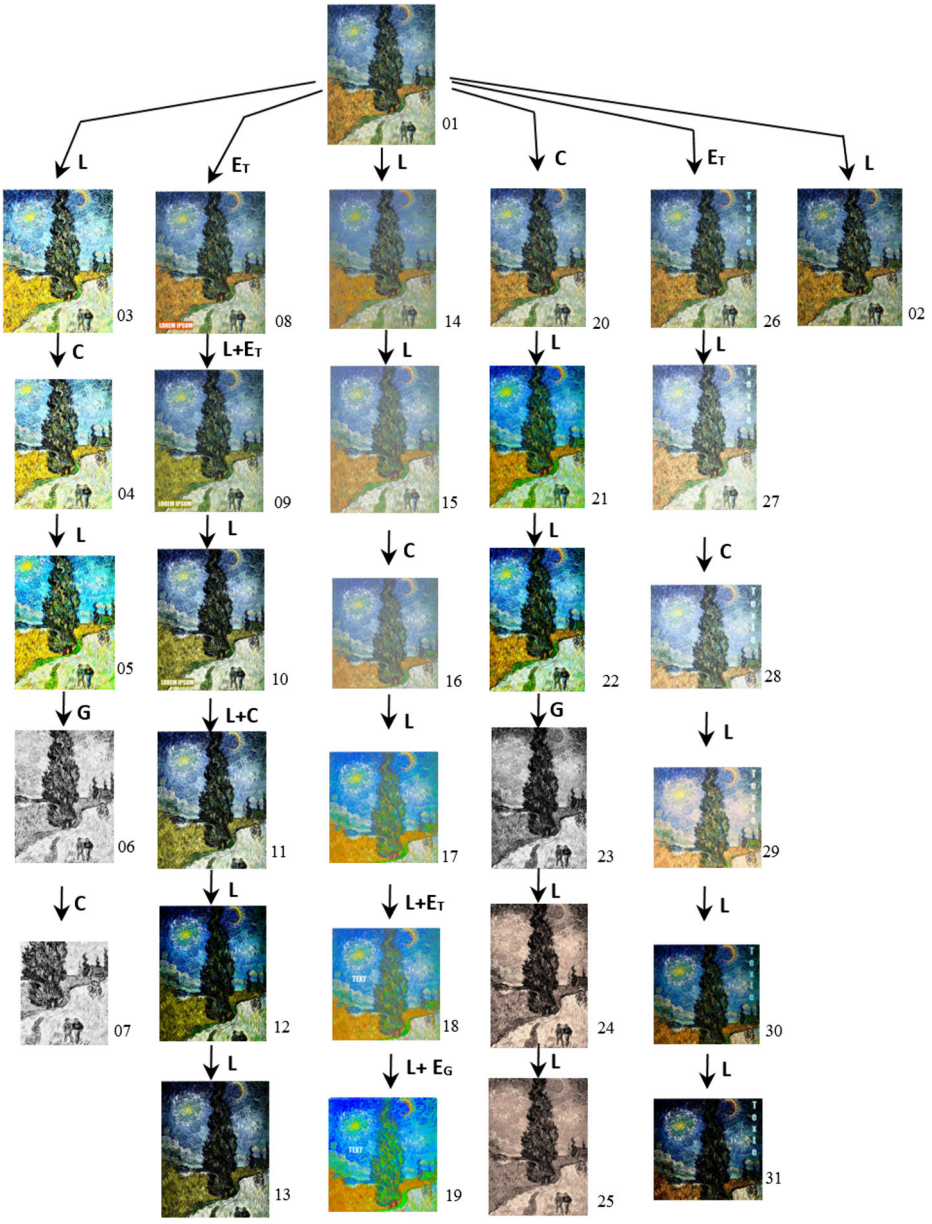


Fig. 9 Constructed copy graph of the fourth dataset of Fig. 8 without using reference images

a score $s_i = 1$, and partially identified if $P(I_i) \in \hat{A}(I_i) - \{I_0\}$, for which we assign a score $s_i = 1/2$. Finally, the total score s of $\mathcal{I}$ is given by the following formula:

$$s = \sum_{I_i \in \mathcal{I}}^{n} s_i / n \tag{7}$$

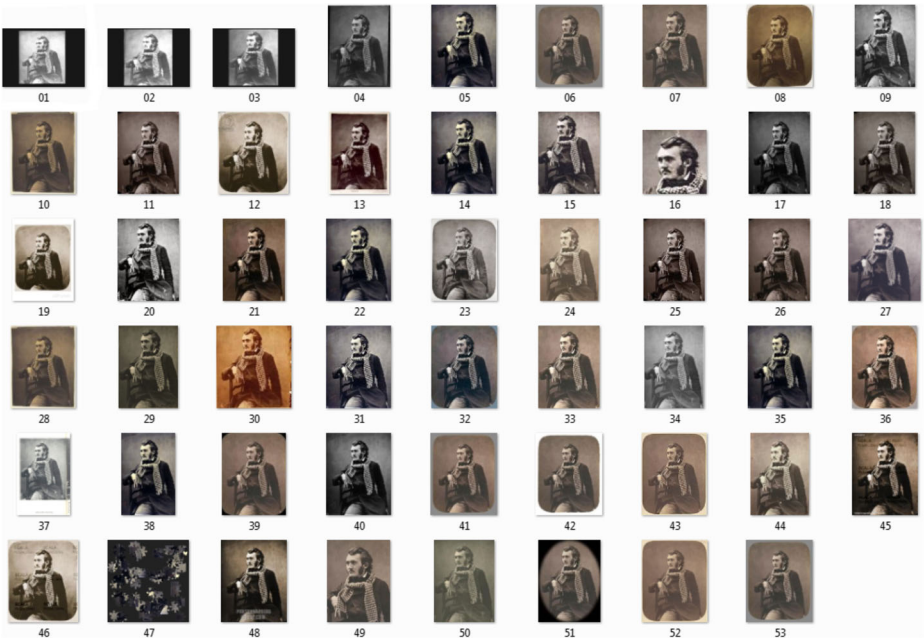


Fig. 10 Copies of *Nadar* photography constituting the third dataset

A perfect match will obviously have a 100% score. To better understand our approach, Fig. 4 shows an illustrative example for two candidate graph reconstructions, (b) and (c), to the reference graph (a) containing 9 edges. Clearly the structure of the reference graph (a) is more similar to (c) than (b). However, the number of identified parents in (c) is lesser than (b). Indeed, using a one-level search, the score of graph (b) is $5/9 \simeq 0.44$, as the number of identified parents is 5 ($I_0 \rightarrow I_1$, $I_0 \rightarrow I_6$, $I_1 \rightarrow I_2$, $I_4 \rightarrow I_9$, and $I_5 \rightarrow I_7$) and the score of graph (c) is $1/9 \simeq 0.11$, as the number of identified parents is 1 ($I_0 \rightarrow I_1$). Using a multi-level search, the number of partially identified parents is one for both graphs, making their total scores $5.5/9 \simeq 0.61$ and $1.5/9 \simeq 0.16$, respectively.

Table 1 shows values of our evaluation metric on each constructed graph from the four datasets. We can see that our algorithm has succeeded in identifying most of the transformations used to create the copies. The algorithm has achieved its best performance in the first three datasets which have been generated manually. The obtained graphs are shown in Figs. 6, 7, 8 and 9, respectively. For the fourth dataset, we run our algorithm on two versions of the dataset. In the first version, we do not provide any reference image (root) to the algorithm and graph reconstruction is preformed in a fully unsupervised fashion. In the second version, reference images identified by an expert were provided to the algorithm (see Fig. 12). For the first version, we can see that even when references were not provided, the algorithm has succeeded to identify the majority of image transformations. By including the reference images in a second version, the accuracy of the algorithm has been increased by almost 10% (see Fig. 10).

The above examples demonstrate the capability of our method for recovering the order of image copies and estimate the most likely transformations used for their generation. However, limitations exist for the algorithm for identifying image modifications where severe

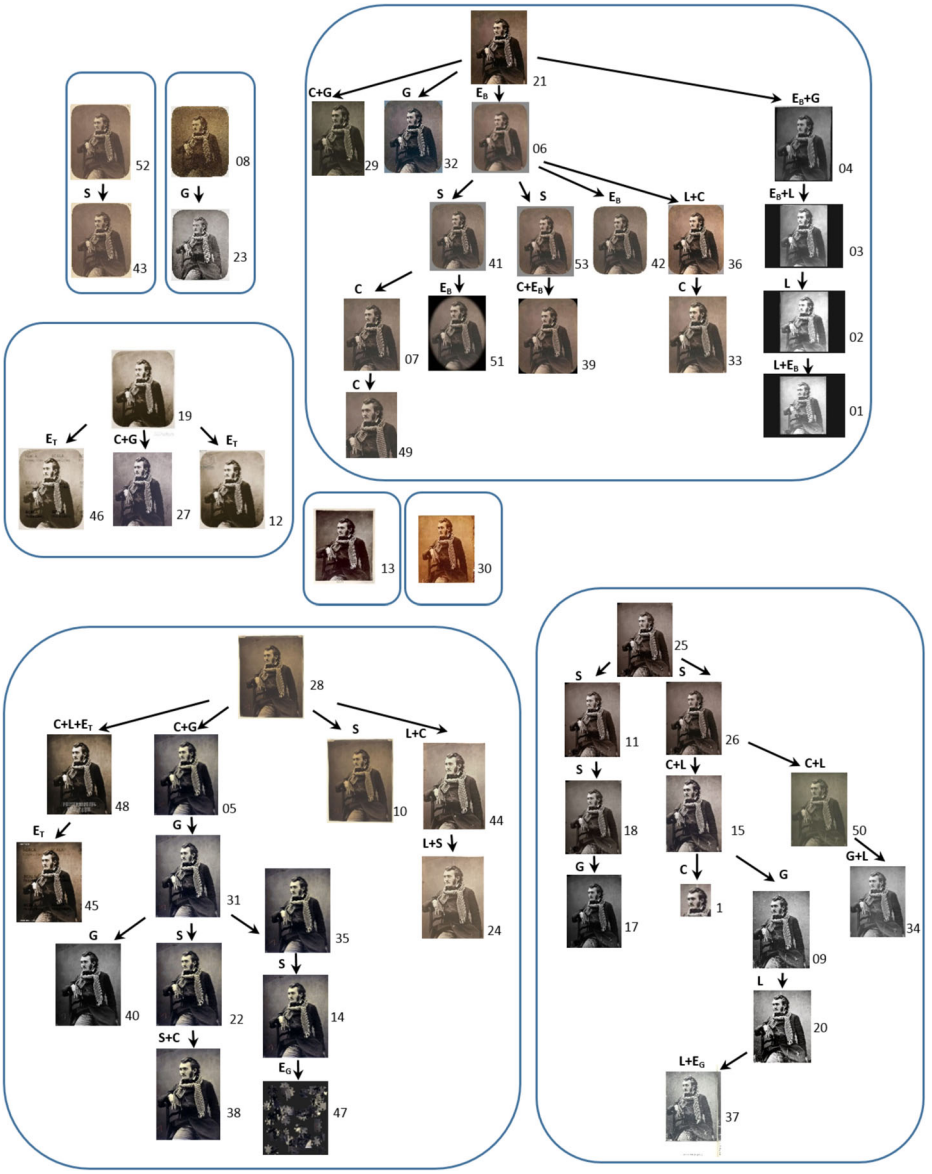


Fig. 11 Constructed copy graph of the third dataset of Fig. 10 without using reference images

global/local changes are operated on the image. This can be seen in some images of our datasets where transformations such as editions and illumination changes have drastically changed the image content. For instance, in images 28 and 31 of Fig. 7, the illumination have been so altered that an edition has been detected by our algorithm. In the same vein, image 47 of Fig. 11 has been so altered that an edition has been detected, but the reconstructed lineages are not accurate (see Fig. 12).

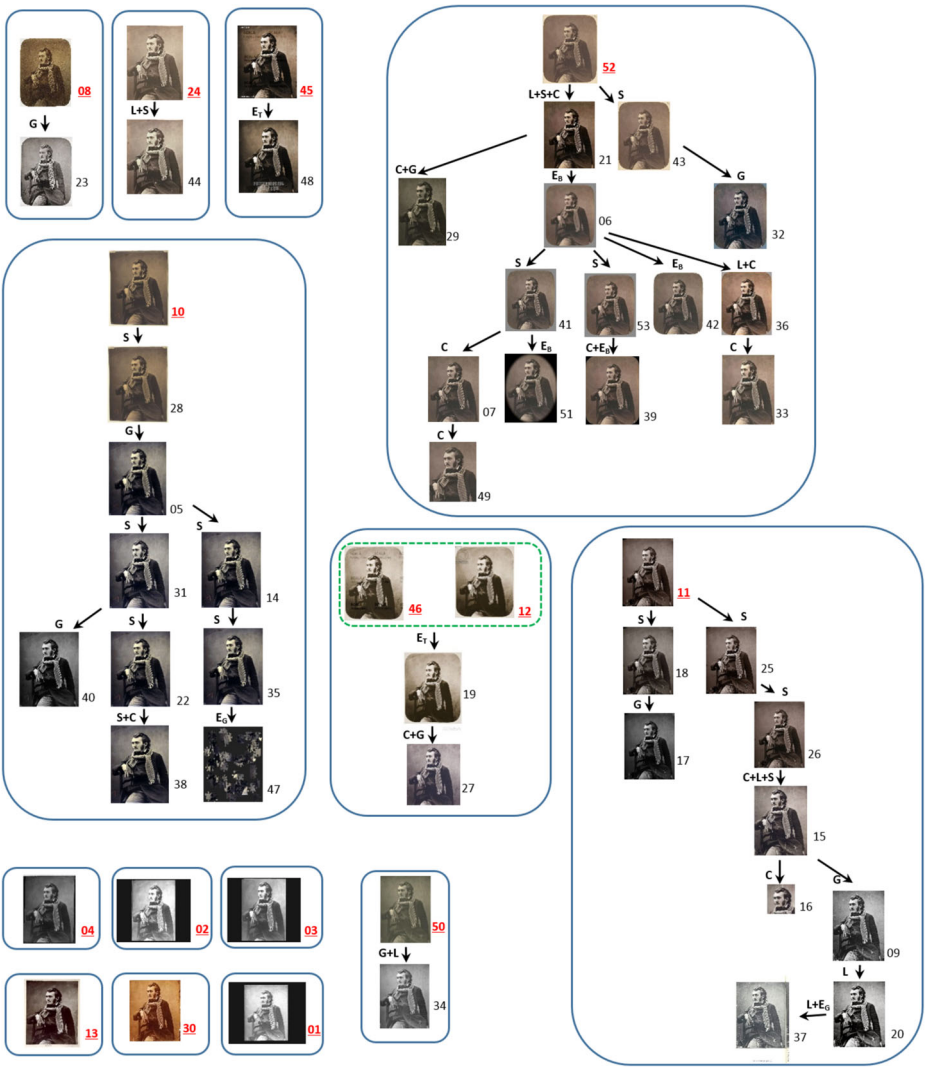


Fig. 12 Constructed copy graph of the third dataset of Fig. 10 using reference images

5 Conclusions

We have proposed a method for building evolution graphs for image copies. The method relies only on visual content of images where several features are used to infer types of transformations operated on images to produce the copies. Experiments results have shown that graph constructed by our algorithm are very close to manually-obtained ones. Future work will focus on enhancing the algorithm by enriching the set of possible transformations and exploring the use of metadata (e.g., URL location, file header, etc.) to improve the precision of the obtained graphs.

Future perspectives for extending this research include setting up a protocol to identify potential roots in a non supervised context. This protocol would take into account the following parameters: analysis of web pages URL and matching with categories of authority (museum, archive, library, auction house, government database), text analysis on the web page that could show citation of a source website (or hyperlink), presence of a watermark that signals a reference to a source or copyright owner. The development of new copyright protection and copy tracking services relying on the blockchain technologies will also provide new ways for identifying original images and roots through their inscription in a distributed directory.

Acknowledgments This work has been achieved thanks to the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) and the University of Quebec en Outaouais. The authors would like to thank Rosa Iris Rodriguez Rovira and Karine Michaud Tessier for their collaboration in dataset collection and processing.

References

1. Allili M (2012) Wavelet modeling using finite mixtures of generalized Gaussian distributions: Application to texture discrimination and retrieval. IEEE Trans Image Process 21(4):1452–1464
2. Allili M, Ziou D (2015) Likelihood-based feature relevance for figure-ground segmentation in images and videos. Neurocomputing 167:658–670
3. Amerini I, Ballan L, Caldelli R, Bimbo AD, Serra G (2011) A SIFT-based forensic method for copy-move attack detection and transformation recovery. IEEE Trans Inform Forensic Secur 6(3):1099–1110
4. Bayram S, Sencar HT, Memon N (2009) An efficient and robust method for detecting copy-move forgery. In: IEEE international conference on acoustics, speech and signal processing, pp 1053–1056
5. Birajdar GK, Mankar VH (2013) Digital image forgery detection using passive techniques: a survey. Dig Investig 23(3):226–245
6. Bosc M, Heitz F, Armspach J.-P., Namer I, Gounot D, Rumbach L (2003) Automatic change detection in multimodal serial MRI application to multiple sclerosis lesion evolution. NeuroImage 20(2):643–656
7. Boulmerka A, Allili M, Ait-Aoudia S (2014) A generalized multiclass histogram thresholding approach based on mixture modelling. Pattern Recogn 47(3):1330–1348
8. Boulmerka A, Allili M (2015) Background modeling in videos revisited using finite mixtures of generalized Gaussians and spatial information. In: IEEE international conference on image processing, pp 3660–3664
9. Camargo JE, Caicedo JC, Gonzalez FA (2013) A Kernel-based framework for image collection exploration. J Vis Lang Comput 24(1):53–67
10. Chen C, McCloskey S, Yu J (2017) Image splicing detection via camera response function analysis. In: IEEE conference on computer vision and pattern recognition, pp 1876–1885
11. Cheng J, Adamic L, Dow PA, Park M, Kleinberg JM, Leskovec J (2014) Can cascades be predicted? In: International conference on world wide web, pp 925–936
12. Christlein V, Riess C, Riess C, Angelopoulou E (2012) An evaluation of popular copy-move forgery detection approaches. IEEE Trans Inform Forensic Secur 7(6):1841–1854
13. Cox I, Miller M, Bloom J, Fridrich J, Kalker T (2007) Digital watermarking and steganography, 2nd edn. Morgan Kaufmann, Burlington
14. Datta R, Joshi D, Li J, Wang JZ (2008) Image retrieval: ideas, influences, and trends of the new age. ACM Comput Surveys 40(2):1–60
15. Duda RO, Hart PE, Stork DG (2002) Pattern classification, 2nd edn. Wiley, Hoboken
16. Farid H (2009) Image forgery detection: a survey. IEEE Signal Proc Mag 26(2):16–25
17. Faulkner A, Chavez C (2017) Adobe-photoshop CC classroom in a book. Adobe Press

18. Fei-Fei L, Fergus R, Perona P (2004) Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: IEEE computer vision and pattern recongition workshop on generative-model based vision, pp 1–9
19. Glumova NI, Kuznetsov AV (2011) Detection of local artificial changes in images optoelectronics. Instrum Data Process 47(3):207–214
20. Goel S, Anderson A, Hofman J, Watts DJ (2016) The structural virality of online diffusion. Manag Sci 62(1):180–196
21. Gonzalez RC, Woods RE (2007) Digital image processing. Springer, Berlin
22. Gu Y, Wang C, Ma J, Nemiroff RJ, Kao DL, Parra D (2016) Visualization and recommandation of large image collections toward effective sensemaking. Inf Vis 16(1):21–47
23. Gu W, Lv Z, Hao M (2017) Change detection method for remote sensing images based on an improved Markov random field. Multimed Tools Appl 76(17):17719–17734
24. Hsiao J-H, Chen C-S, Chien L-F, Chen M-S (2007) A new approach to image copy detection based on extended feature sets. IEEE Trans Image Process 16(8):2069–2079
25. Hsu T-C, Hsieh W-S, Chiang JY, Su T-S (2011) New watermark-removal method based on Eigen-Image energy. IET Inf Secur 5(1):43–50
26. Hussain M, Chen D, Cheng A, Wei H, Stanley D (2013) Change detection from remotely sensed images: from pixel-based to object-based approaches. ISPRS J Photogramm Remote Sens 80:91–106
27. Ilsever M, Unsalan C (2012) Pixel-based change detection methods. Two-dimensional change detection methods: remote sensing applications. Springer, Berlin
28. Ke Y, Sukthankar R, Huston L (2004) Efficient near-duplicate detection and sub-image retrieval. In: ACM international conference on multimedia, pp 869–876
29. Kim C (2003) Content-based image copy detection. Signal Process Image Commun 18(3):169–184
30. Krishnan S, Butler P, Tandon R, Leskovec J, Ramakrishnan N (2016) Seeing the forest for the trees new approaches to forecasting cascades. In: ACM conference on web science, pp 249–258
31. Lin CY, Wu M, Bloom JA, Cox IJ, Miller ML, Lui YM (2001) Rotation, scale, and translation resilient watermarking for images. IEEE Trans Image Process 10(5):767782
32. Liu G-H, Yang J-Y (2013) Content-based image retrieval using color difference histogram. Pattern Recogn 46(1):188–198
33. Luo W, Huang J, Qiu G (2006) Robust detection of region-duplication forgery in digital images. In: IEEE international conference on pattern recognition, pp 746–749
34. Mahdian B, Saic S (2007) Detection of Copymove forgery using a method based on blur moment invariants. Forensic Sci Int'l 171(23):180–189
35. Manovich L (2012) Media visualization: visual techniques for exploring large media collections. Media Studies Futures
36. Muhammad G, Hussaina M, Bebis G (2012) Passive copy move image forgery detection using undecimated dyadic wavelet transform. Digit Investig 9(1):49–57
37. Ng T, Chang S (2004) A model for image splicing. In: IEEE international conference on image processing, pp 1169–1172
38. Ouyed O, Allili M (2018) Feature weighting for multinomial kernel logistic regression and application to action recognition. Neurocomputing 275:1752–1768
39. Pan X, Lyu S (2010) Region duplication detection using image feature matching. IEEE Trans Inf Forensic Secur 5(4):857–867
40. Park J, Tai Y-W, Kweon IS (2012) Identigram/watermark removal using cross-channel correlation. In: IEEE conference on computer vision and pattern recognition, pp 446–453
41. Petrou M, Petrou C (2010) Image processing: the fundamentals, 2nd edn. Wiley, Hoboken
42. Radke RJ, Andra S, Al-Kofahi O, Roysam B (2005) Image change detection algorithms: a systematic survey. IEEE Trans Image Process 14(3):294–307
43. Stefano LD, Mattoccia S, Mola M (2003) An efficient algorithm for exhaustive template matching based on normalized cross-correlation. In: International conference on image analysis and processing, pp 322–327
44. Szeliski R (2006) Image alignment and stitching: a tutorial. Found Trends Comput Graph Vis 2(1):1–104
45. TinEye. https://www.tineye.com/
46. Tsai D-M, Lin C-T (2003) Fast normalized cross correlation for defect detection. Pattern Recogn Lett 24:2625–2631
47. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612
48. Wang W, Dong J, Tan T (2009) Effective image splicing detection based on image chroma. In: IEEE international conference on image processing, pp 1257–1260

49. Wang K, Babenko B, Belongie S (2011) End-to-end scene text recognition. In: IEEE international conference on computer vision, pp 1457–1464
50. Wang C, Reese JP, Zhang H, Tao J, Nemiroff RJ (2013) IMap: a stable layout for navigating large image collections with embedded search. In: Visualization and data analysis, pp 1–14
51. Wang C, Reese JP, Zhang H, Tao J, Gu Y, Ma J, Nemiroff RJ (2015) Similarity-based visualization of large image collections. Inf Vis 14(3):183–203
52. Yan L, Zou F, Guo R, Gao L, Zhou K, Wang C (2016) Feature aggregating hashing for image copy detection. World Wide Web 19(2):217–229
53. Yang R, Kalnis P, Tung AKH (2005) Similarity evaluation on tree-structured data. In: ACM SIGMOD international conference on management of data, pp 754–765
54. Ye S, Sun Q, Chang E-C (2007) Detecting digital image forgeries by measuring inconsistencies of blocking artifact. In: IEEE international conference on multimedia and expo, pp 12–15
55. Zhang K, Shasha D (1989) Simple fast algorithms for the editing distance between trees and related problems. SIAM J Comput 18:1245–1262
56. Zhang D-Q, Chang S-F (2004) Detecting image near-duplicate by stochastic attributed relational graph matching with learning. In: ACM international conference on multimedia, pp 877–884
57. Zhang W, Cao X, Qu Y, Hou Y, Zhao H, Zhang C (2010) Detecting and extracting the photo composites using planar homography and graph cut. IEEE Trans Inf Forensics Security 5(3):544–555
58. Zhao W-L, Ngo C-W (2013) Flip-invariant SIFT for copy and object detection. IEEE Trans Image Process 22(3):980–991
59. Zhong J, Gan Y, Young J, Huang L, Lin P (2017) A new block-based method for copy move forgery detection under image geometric transforms. Multimed Tools Appl 76(13):14887–14903

Mohand Said Allili Received the M.Sc. and Ph.D. degrees in computer science from the University of Sherbrooke, Sherbrooke, QC, Canada, in 2004 and 2008, respectively. Since June 2008, he has been an Assistant Professor of computer science with the Department of Computer Science and Engineering, Université du Québec en Outaouais, Canada. His main research interests include computer vision and graphics, image processing, pattern recognition, and machine learning. Dr. Allili was a recipient of the Best Ph.D. Thesis Award in engineering and natural sciences from the University of Sherbrooke for 2008 and the Best Student Paper and Best Vision Paper awards for two of his papers at the Canadian Conference on Computer and Robot Vision 2007 and 2010, respectively.

Nathalie Casemajor is an Assistant Professor in the Urbanisation Culture Société Research Centre at INRS (Institut national de la recherche scientifique, Montreal). Her work focuses on cultural development and digital culture. She is the coordinator of the Groupe de recherche sur la mediation culturelle and coedited the book Expériences critiques de la médiation culturelle (PUL, 2017). She also conducted research projects on cultural institutions and Wikipedia, arts and public space and the circulation of news and artworks on the Web.

Aymen Talbi obtained is license degree in computer science in Tunisia in 2012. He then earned his Master Degree in computer science at the University du Quebec in Outaouais in 2016. He has worked under the supervision of professors Mohand Said Allili and Nathalie Casemajor.