Image Copy Detection and Evolution Visualisation Using Tree Graphs

Mohand Said Allili, Nathalie Casemajor, Aymen Talbi,

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 visualise 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 and search on the Web.

Index Terms

Image copy detection, image transformation, copy evolution graph, history visualisation.

I. Introduction

The growing popularity of Internet and social media has enabled ubiquitous and distributed sharing of digital photos among Internet users. This is accompanied by new possibilities to copy, alter and distribute digital content easily to a large number of recipients thanks to the availability and accessibility of image processing software [13]. 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. Tracking image copies can also be useful for analysing the dissemination and impact of photos in artistic/journalistic Web communities [2], [41]. Although several methods have been proposed in the past for image copy detection [2], [17], [13], not much research have been conducted for the purpose of tracing and/or visualising copy image evolution in the Web.

With the advent of content based image retrieval (CBIR) systems, the possibility of browsing and making instant searches in the Web, based on visual content, have opened new frontiers for image-based applications such as security, tourism, journalism, Web analytics [6]. Among important and common problems in these applications is the detection and management of image copies. In recent years, image copy detection has received the focus of several research works [13], [37], [41]. In opposite to general-purpose CBIR, where images exhibiting the same global visual properties as the query image are searched (e.g., color, texture, layout, etc.), copy detection aims at finding copies of an original image, which have undergone some kind of alteration [2]. An alteration can be of several types (e.g., photometric, geometric, editing, compression, etc.), which modifies locally or globally the visual content of the image.

Early approaches for copy detection are based on watermarking which consists of embedding signatures (or watermarks) for copyright protection [5]. A watermark is an invisible and embedded identification code carrying information about the copyright owner [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 [14], [20]. Recently, the concept of content-based copy detection (CBCD) has been proposed as an alternative to watermarking for detecting illegal image copies [8], [17]. The goal of a CBCD system 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 [27], [35]. Several methods have been proposed for CBCD in recent years [2], [17]. Worth mentioning are CBCD systems available online such as TinEye and Piximilar [27]. To detect possible copies, these systems are based on low-level feature extraction, followed by similarity measurement between images.

Copy detection systems are generally good at finding similarities between images and detecting identical content. However, they do not tell much about the type of transformation operated on an image to produce its copies. This is due to the difficulty of having a unique way of determining the variety of transformations that can be applied on an image. Indeed, there is a large number of image manipulation possibilities, ranging from simple geometric/photometric transforms or resizing to more complex transforms, such as image editing, cropping, stitching and compression [21]. 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 [25]. Therefore, having a unique algorithm to detect all types of transformation seems an illusive pursuit. Nonetheless, one can make the problem more tractable by making some assumptions about the generative process of copies from original images. Also, transformations can be categorized into several types (e.g., photometric, geometric) to facilitate investigation about the potential set of manipulations operated to produce each image copy.

In this paper, we propose a method for constructing an evolution graph for a set of image copies having the same origin. Given a root image or a set of reference images determined by experts, our method infers the most likely transformations used to produce the remaining image copies. For simplicity, we focus mainly on three types of image transformations for copy production: *geometric*, *photometric*, and *image editing*. *Geometric* transformations refer to changes due to affine transforms and image distortions [21]. *Photometric* transformations refer to image color enhancement, filtering and color-to-gray transform [10]. Finally, *editing* refers to basic operations such as image cropping, zooming, seam carving and text/object insertion [9]. To detect relations between images, our method is based on three main steps. The first step consists in detecting simple transformations such as image cropping, resizing, rotations and color-to-gray transforms. These transformations, although easy to detect, enable to establish strong ties between images. Therefore, they will be used to initialize the copy evolution graph. The second step consists of inferring the remaining relations between images by detecting the most likely predecessor for each image. Thus, each image will have its lineage assessed through the constructed graph. The final step consists of annotating the graph with the most likely transformations used to produce the copies. Several 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 II describes the proposed approach for identifying image transformations and evolution graph visualization of copies. Section III presents experimental results validating our approach. We end the paper with a conclusion and future work perspectives.

II. 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. The goal is to retrace the order of creation of the copies in a form of a tree graph depicting the history of copy creation from I_0 to its descendent(s). 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 the image copies and unravel the most likely transformations used through it.

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*. Note that since we can have several types of editions, some of which are carried out at a local level (e.g., text insertion/removal, object insertion, face blurring, etc.) and the others at a global level (e.g. image distortion, compression, etc.), we use a subscript to indicate the edition type: E_T: *text insection/removal*, E_B: *border insertion* and E_G: *global manipulation (e.g., noise, compression, etc.)*. Finally, 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 where 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 smallest angle, that is $I_A \to I_C$. The goal of our approach is to infer the most likely set of transformations used to produce $I_1, ..., I_n$, and build on an evolution graph depicting the lineage of each image copy. Our algorithm starts by forming relations with the strongest evidence among the images, and then completing the graph by inferring the rest of the relations using image processing techniques.

A. Estimating strong relations between images

One of the strongest relations that can be estimated with high confidence among images are: *cropping (C), color to gray (G), rotations (R) and scale changes (S)*. Indeed, these transformations do not produce major modifications in the image content which makes them relatively easy to be detected through correlation analysis. More formally, we measure normalized cross-correlation (NCC) 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. To simplify things, we

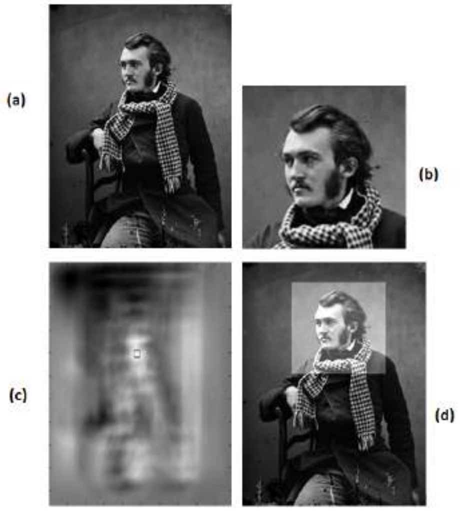


Fig. 1: Cropping detection using CNN. (a) represents the original image I_A, (b) represents the result after cropping image I_A. (c) represents the value of NCC at different locations in I_A and (d) represents I'_A a highlight of the location if I'_A.

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 $s \in \mathbb{N}^*$ 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.
- 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.

Note that to detect the above relations, our algorithm tests the three first transformations on each pair of images using different combinations of the transformation parameters. More specifically, we test the occurrence of cropping at each image location using 8 values of rotations and 5 scales, making up to 40 tests at each location. For instance, as shown in Fig. 1, a cropping of the image (a) followed by a scale change, have been used to produce the image (b). The images in (c) and (d) show, respectively, the NCC map and the cropping location corresponding to the highest correlation value.

B. 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 regard, two cases can arise. In the first one, no reference images are available and a fully unsupervised grouping is then performed. In the second one, some image references are available (that will constitute the roots of our graph), and we can perform a supervised grouping, thereof. The reference images can either be designated by an expert or found in official sites such as museums, archives and art galleries, for example. Each reference image is then used as centroid on which a cluster is built to constitute one *lineage*.

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 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 [7]. Finally, distance between two images is taken as follows:

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

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 previous section. The inclusion of the first condition 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. 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.

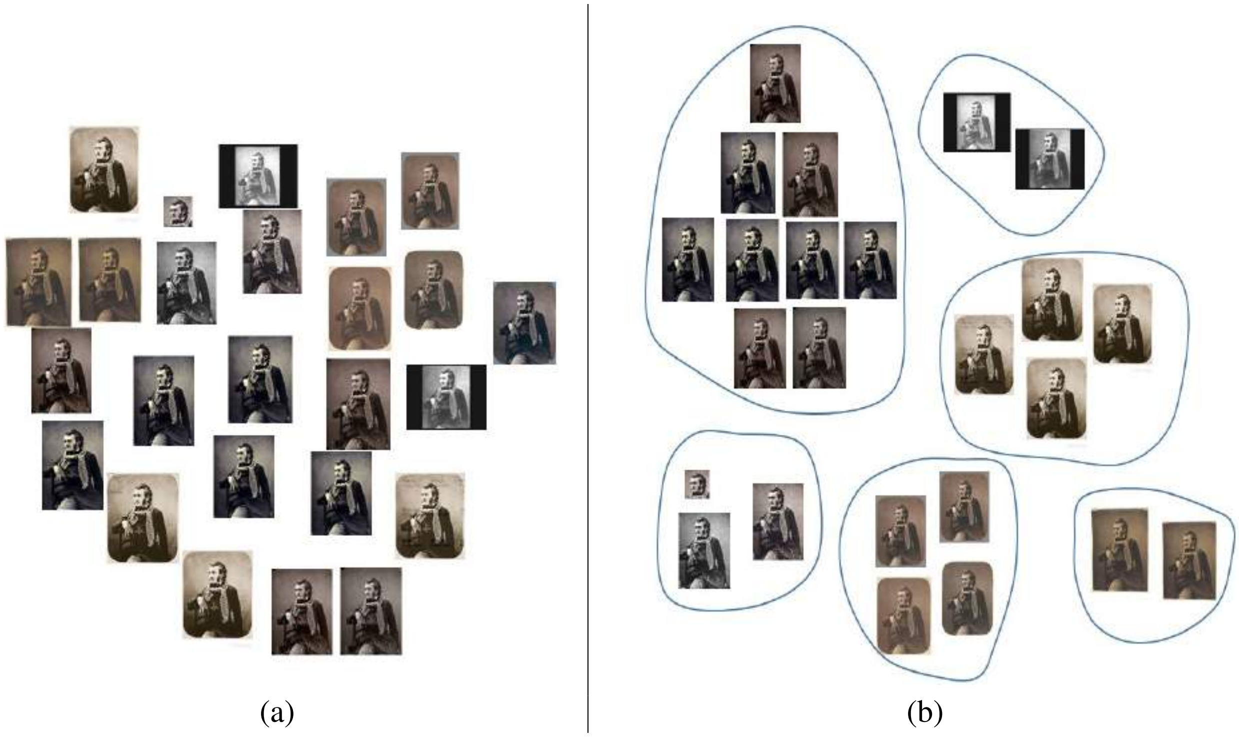


Fig. 2: Illustration of image grouping by histogram comparison: (a) copy image set, (b) obtained copy groups constituting *lineage* clusters.

C. Detecting image editions

Image editing refers to improving digital images by removing unwanted elements such as scratches, red eye effect, etc., or inserting elements such as objects, text or image sharpening. Image editing can be carried out at pixel or global (parametric) level. Pixel-level editing affects the image at the pixel level (e.g., text/object insertion/removal, scrach/red eye removal, etc.). Global-level editing refers to transformations affecting all the pixels of the image (e.g., image distortion, compression, illumination changes, etc.). In our work, we deal with these two types of editions by using appropriate features extracted from the image. To discriminate between the different types of editions, we use content-based image similarity measurements upon which a supervised classification is performed.

Let I_A and I_B be two images, where I_B is the result of editing I_A. To compare the two images at a local level, we first align them using [25]. 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 [30]:

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

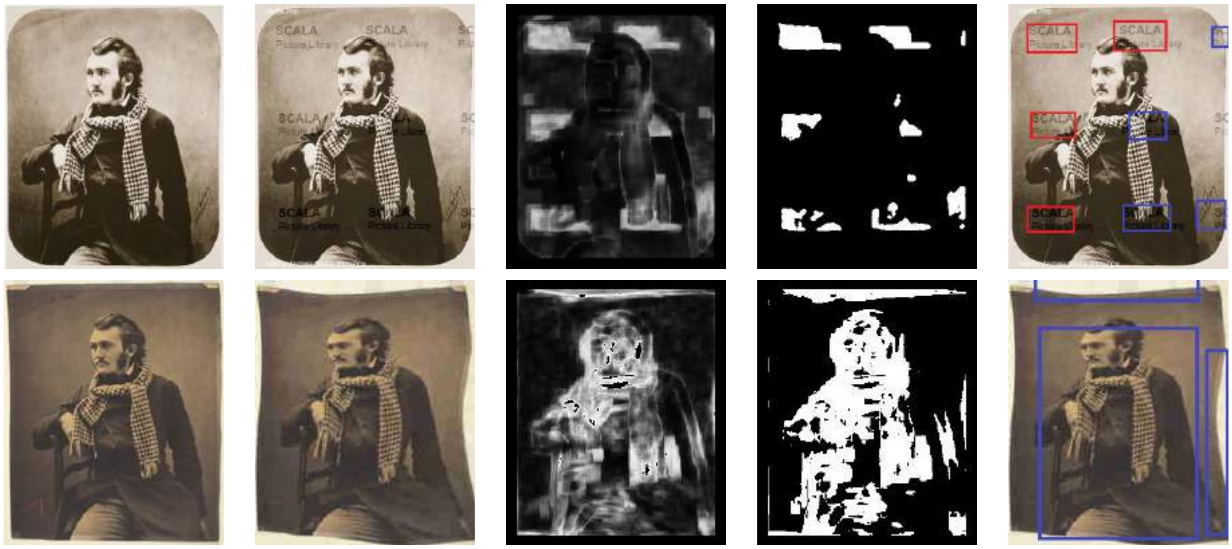
where $N(\mathbf{x})$ stands for the neighbourhood of location $\mathbf{x} = (x, y)$. The parameters $(\mu_A(\mathbf{x}), \sigma_A(\mathbf{x}))$ and $(\mu_B(\mathbf{x}'), \sigma_B(\mathbf{x}'))$ represent local mean and standard deviation in images I_A and I_B at locations $\mathbf{x}$ and $\mathbf{x}'$, 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.

Using Eq. (4), we generate a map as the one shown in the column (c) of Fig. 3, which is segmented to a binary map as shown in column (d). We then extract the most important blobs from the binary map, we analyse four of their geometric properties: *eccentricity*, *relative area*, *dispersion* and *precence/absence of text*. We train an SVM to classify wether a generated map stems from a local or global edition. Note that to detect a text insertion, we use the method [29] to analyse each generated blob. In the first row of Fig. 3, we show in column (e) some detections of regions considered as text insertions (red rectangles) and others as non-text insertions (blue rectangles). In the second row of the figure, we show an example of image distortion, where the edition is detected as a global one.

D. 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 operated on images to create copies (graph descendants). Since we can have several resulting groups in our clustering, a sub-graph is constructed within each group by incrementally ordering images in the group using 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.

The algorithm is composed of two main steps. In a 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



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

Fig. 3: Detection of image editions by comparing local image structure: (a) original image, (b) evicted image, (c) change map generated using Eq. (4), (d) binary segmentation of (d), (e) text detection using [29].

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 Eq. (2). Local information is measured by averaging the value of Eq. (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). We use the symbol G_c to designate images from group G_v added to the graph, whereas $\bar{G}_c$ designates the complementary of G_c to G_v. Note that since line 10 of the algorithm builds edges between images linked by strong relations, adding a parent image among these images to the set G_c will add all its descendants to G_c (see lines 22 to 25). Ultimately, all vertices in G_c will be 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 $\exists T \in \{R, G, C\}$ with parameter $\phi$ 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 Eq. (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 $I_{i_2}$;
            $G_c \leftarrow G_c \cup \{I_{i_2}\} \cup D$;
            $\bar{G_c} \leftarrow \bar{G_c} \setminus (\{I_{i_2}\} \cup D)$;
        end while
    end if
    $\mathcal{V} \leftarrow \mathcal{V} \cup G_c$;
end for
```

III. Experiments

To validate the proposed approach, experiments have been conducted on three datasets. In the first two datasets, 64 image copies are built manually by carrying out a series of transformations to the original *Mona Lisa* image. Figs. 4.a and 4.b show the two constructed datasets, each one containing a total of 32 images. The third dataset has been manually collected from the Web (see Fig. 7) and contains 53 copies of a photograph by famous French artist *Nadar*. The public archive that owns the negative has published several scans online. 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 two datasets are complete (comprising all roots and descendants in every lineage), the third dataset has gaps: some roots and descendants may be missing, which is likely to be the case when copies are largely collected from the Web.

To evaluate the accuracy of constructed evolution graphs, we compared them to ground truth 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 analysis of the context of publication of images on web pages (e.g., authority of the website, mention of source, presence of hyperlink). For the first two datasets, since we have generated all the copies, a reference graph is built based on the transformations used to create the copies. For the third dataset, a reference graph is manually generated. To assess the performance of our method, we measure the similarity of the obtained graphs with the ground truth. For this goal, such a similarity measurement must take into account the following specificities for our graphs. Since we are not interested in precise creation time of the image copies, horizontal order of vertices within the same level is not important. However, the vertical order of vertices in each vertical path from the root is important since it depicts the lineage of each copy that is generated from its ancestors. Therefore, we cannot use directly existing methods for comparing tree graphs since most of them are dedicated to binary ordered trees. [39], [38]. Instead, we designed another measure with details given as follows:

Let $I_i \neq I_0$ be a vertex in the constructed graph $\mathcal{G}$ and $Pa(I_i)$ its parent in the same graph. Let $Anc(I_i)$ be the set of ancestors of I_i different form $Pa(I_i)$ in the graph. Let I_j be the image from which I_i is originated. In a uni-level search, we consider that the parent of I_i is correctly identified if $I_j = Pa(I_i)$. We assign a score $s_i = 1$ if a parent is correctly identified, and $s_i = 0$, otherwise. In a multi-level search, a parent of I_i is correctly identified if $I_j = Pa(I_i)$ and partially identified if $I_j \in Anc(I_i)$. To differentiate between the two cases, we assign score $s_i = 1$ if a parent is correctly identified, $s_i = 1/2$ if a parent is partially identified and $s_i = 0$ otherwise. Finally, the total score s of a dataset $\mathcal{I}$ is given by the following formula: $s = \sum_{i=1}^{n} s_i/n$. A perfect match will obviously have a 100% score.

Table I shows values of our evaluation metric on each constructed graph from the three datasets. We can see that our algorithm has succeeded in identifying most of the transformations used to

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

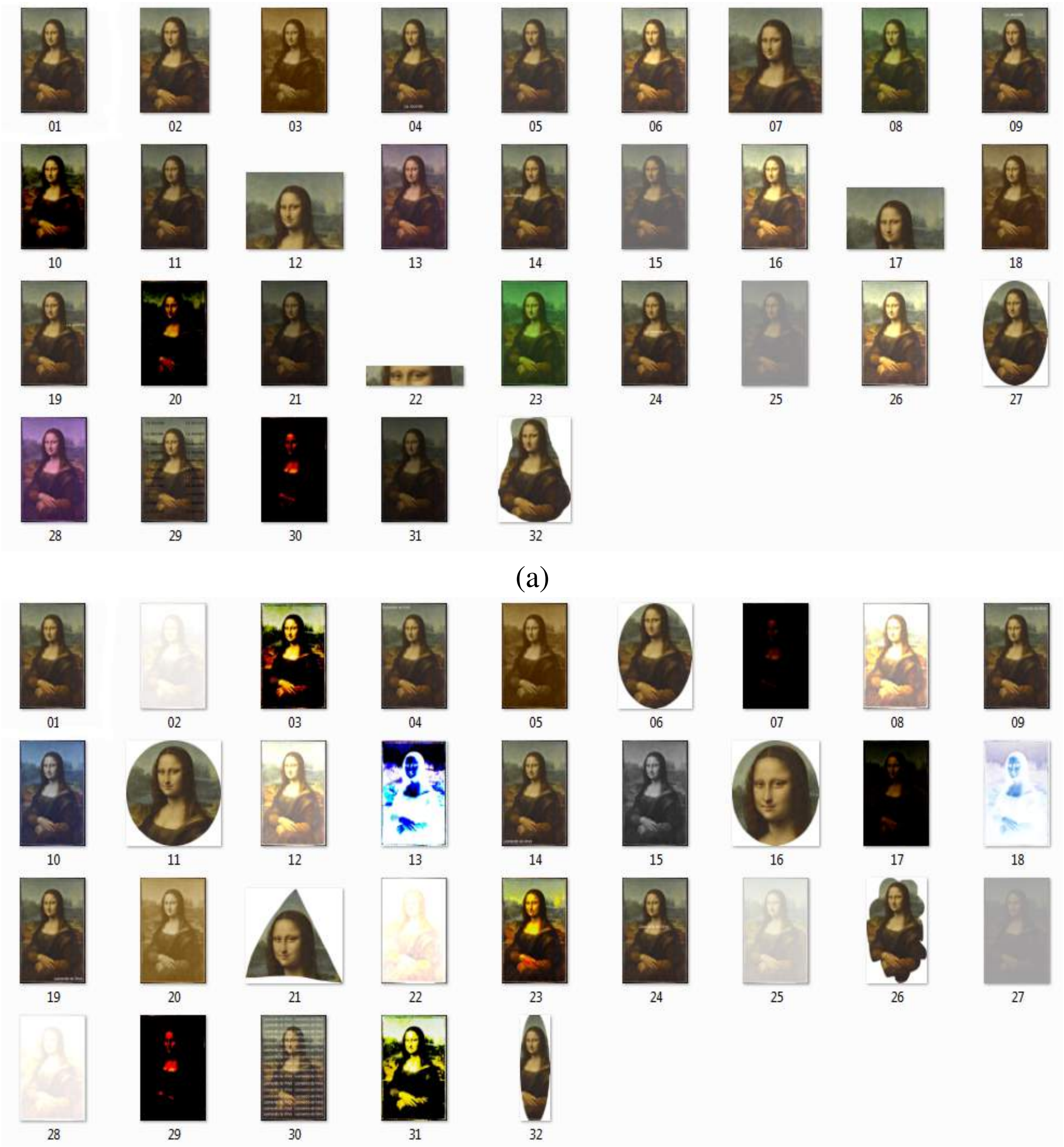
TABLE I: Values of our evaluation metric on each constructed graph from the three datasets. The first and second rows show values obtained using uni-level and multi-level parent search, respectively.

create the copies. Since the first two datasets have been generated manually, the algorithm has achieved its best performance. The two graphs obtained for these two data sets are shown in Figs. 5 and 6, respectively. As for the third dataset, we run our algorithm on two versions of the dataset, the first one does not contain any reference image (root), while the second contains reference images identified by an expert (see Fig. 9). 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%.

IV. 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.



(a)

(b)

Fig. 4: Copies of *Mona Lisa* image constituting our two first datasets (a) and (b).

V. Acknowledgements

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.

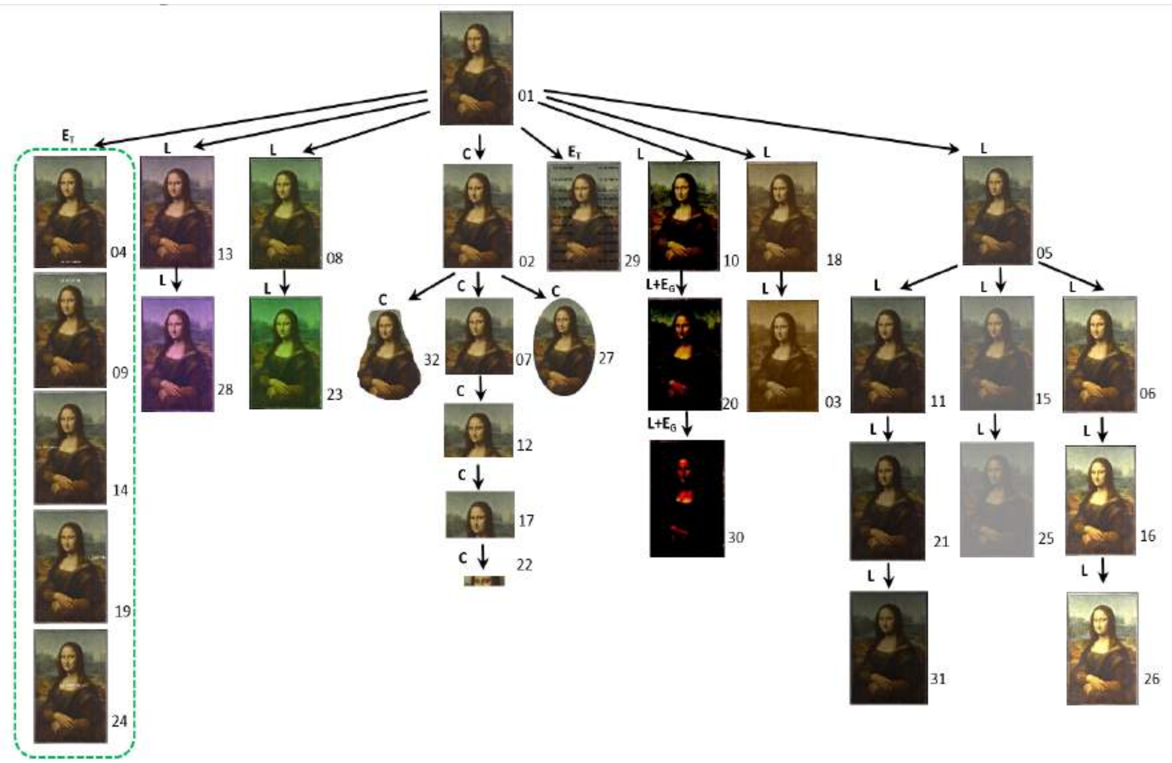


Fig. 5: Constituted copy graph of the second dataset in fig. 4.a

References

[1] A. Alzubi, A. Amira and N. Ramzan. Semantic Content-Based Image Retrieval: A Comprehensive Study. *J. of Visual Communication and Image Representation*, 32:20-54, 2015.

[2] G.K. Birajdar, V. H. Mankar. Digital Image Forgery Detection Using Passive Techniques: A Survey. *Digital Investigation*, 23(3):226-245, 2013.

[3] P. Brivio, M. Tarini and P. Cignoni. Browsing Large Image Datasets Through Voronoi Diagrams. *IEEE Trans. on Visualization and Computer Graphics*, 16(1):1261-1270, 2010.

[4] J. E. Camargo, J. C. Caicedo and F. A. Gonzalez. A Kernel-Based Framework for Image Collection Exploration. *Journal of Visual Languages and Computing*, 24(1):53-67, 2013.

[5] I. Cox, M. Miller, J. Bloom, J. Fridrich and T. Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann, 2nd Ed., 2007.

[6] R. Datta, D. Joshi, J. Li, and J.Z. Wang. Image Retrieval: Ideas, Influences, and Trends of the New Age. *ACM Computing Surveys*, 40(2), Article 5, 2008.

[7] R.O. Duda, P.E. Hart and D.G. Stork. *Pattern Classification*, 2nd Edition, Wiley, 2002.

[8] H. Farid. *Image Forgery Detection: A Survey*. *IEEE Signal Processing Magazine*, 26(2):16-25, 2009.

[9] A. Faulkner and C. Chavez. *Adobe-Photoshop CC Classroom in a Book*. Adobe Press, 2017.

[10] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Springer, 2007.

[11] Y. Gu, C. Wang, J. Ma, R.J. Nemiroff, D.L. Kao and D. Parra. Visualization and Recommandation of Large Image Collections Toward Effective Sensemaking. *Information Visualisation*, 16(1):2147, 2016.

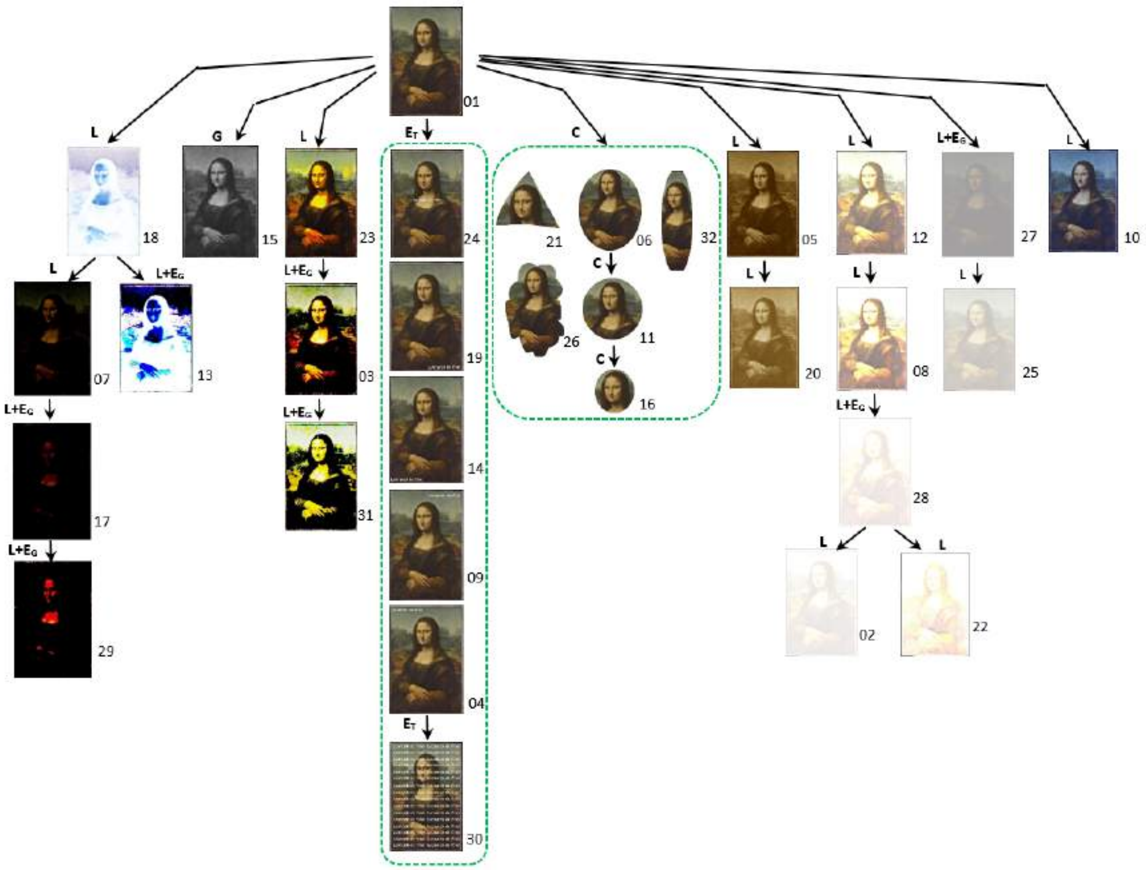


Fig. 6: Constituted copy graph of the second dataset in fig. 4.b

[12] I. Herman, G. Melançon and M. S. Marshall. Graph Visualization and Navigation in Information Visualization. *IEEE Trans. on Visualization and Computer Graphics*, 6(1): 24-43, 2000.

[13] J-H. Hsiao, C-S. Chen, L-F. Chien and M-S. Chen. A New Approach to Image Copy Detection Based on Extended Feature Sets. *IEEE Trans. on Image Processing*, 16(8):2069-2079, 2007.

[14] T.-C. Hsu, W.-S. Hsieh, J.Y. Chiang and T.-S. Su. New Watermark-Removal Method Based on Eigen-Image Energy. *IET Information Security*, 5(1):43-50, 2011.

[15] S-J. Hwang and K. Grauman. Learning the Relative Importance of Objects from Tagged Images for Retrieval and Cross-Modal Search. *Int'l Journal of Computer Vision*, 100(2):134-153, 2012.

[16] Y. Ke, R. Sukthankar and L. Huston. Efficient Near-Duplicate Detection and Sub-image Retrieval. *ACM Int. Conf. on Multimedia*, 869-876, 2004.

[17] C. Kim. Content-Based Image Copy Detection. *Signal Processing: Image Communication*, 18(3):169-184, 2003.

[18] L. Manovich. Media visualization: Visual Techniques for Exploring Large Media Collections. *Media Studies Futures*, 2012.

[19] L. Manovich. How to Compare one Million Images. *Understanding Digital Humanities*, 249-278, 2012.

[20] J. Park, Y-W. Tai and I.S. Kweon. Identigram/Watermark Removal Using Cross-Channel Correlation. *IEEE Conference on Computer Vision and Pattern Recognition*, 446-453, 2012.

[21] M. Petrou and C. Petrou. *Image Processing: The Fundamentals*. Wiley 2nd Ed., 2010.

[22] A. Rocha, W. Scheirer, T. Boult and Goldenstein. Vision of the Unseen: Current Trends and Challenges in Digital Image

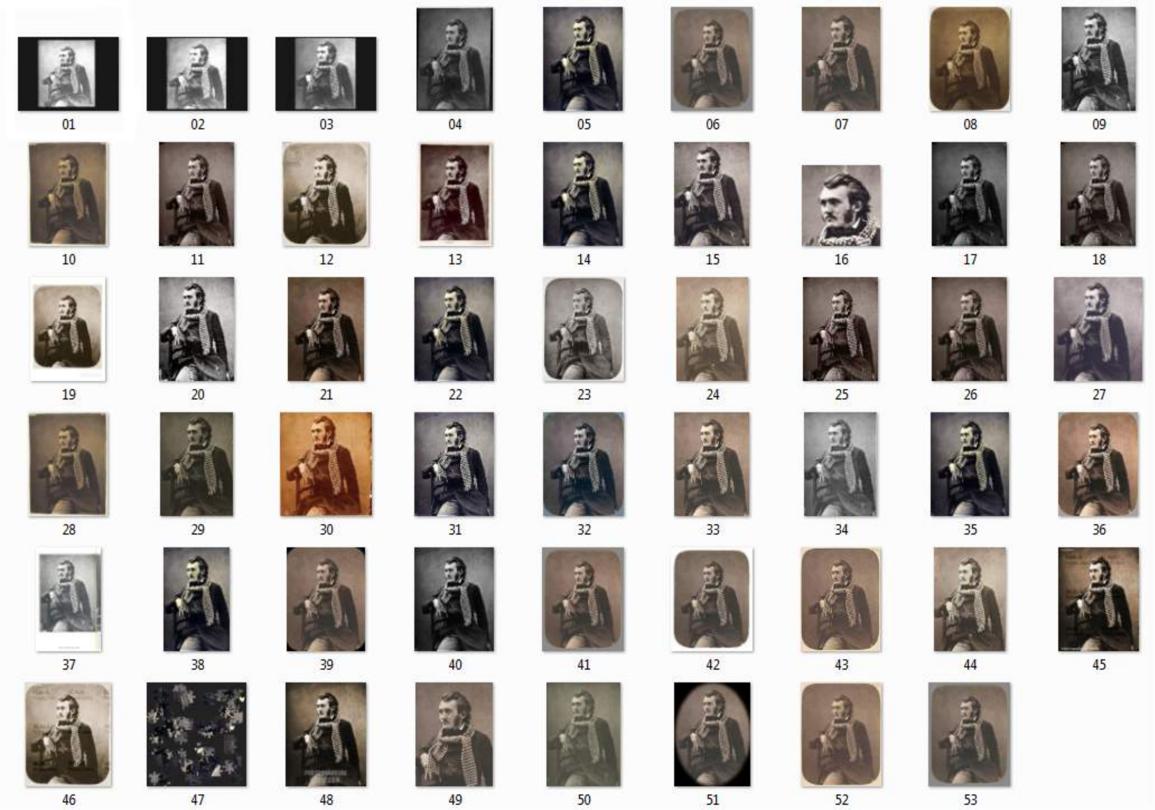


Fig. 7: Copies of *Nadar* photography constituting the third dataset.

and Video Forensics. *ACM Computing Surveys*, 43(4):1-26, 2011.

[23] K. Schoeffmann and D. Ahlström. Similarity-Based Visualization for Image Browsing Revisited. *IEEE Int'l Symposium on Multimedia*, 422-427, 2011.

[24] L. D. Stefano, S. Mattoccia and M. Mola. An Efficient Algorithm for Exhaustive Template Matching Based on Normalized Cross-Correlation. *Int'l Conf. on Image Analysis and Processing*, 322-327, 2003.

[25] R. Szeliski. Image Alignment and Stitching: A Tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2(1):1104, 2006.

[26] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2011.

[27] TinEye. *https://www.tineye.com/*.

[28] J. Z. Wang, J. Li and G. Wiederhold. SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(9):947-963, 2001.

[29] K. Wang, B. Babenko and S. Belongie. End-to-End Scene Text Recognition. *IEEE Int'l Conf. on Computer Vision*, 1457-1464, 2011.

[30] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. on Image Processing*, 13(4):600-612, 2004.

[31] C. Wang, J.P. Reese, H. Zhang, J. Tao and R. J. Nemiroff. IMap: A Stable Layout for Navigating Large Image Collections with Embedded Search. *Visualization and Data Analysis*, 1-14, 2013.

[32] C. Wang, J.P. Reese, H. Zhang, J. Tao, Y. Gu, J. Ma and R.J. Nemiroff. Similarity-Based Visualization of Large Image

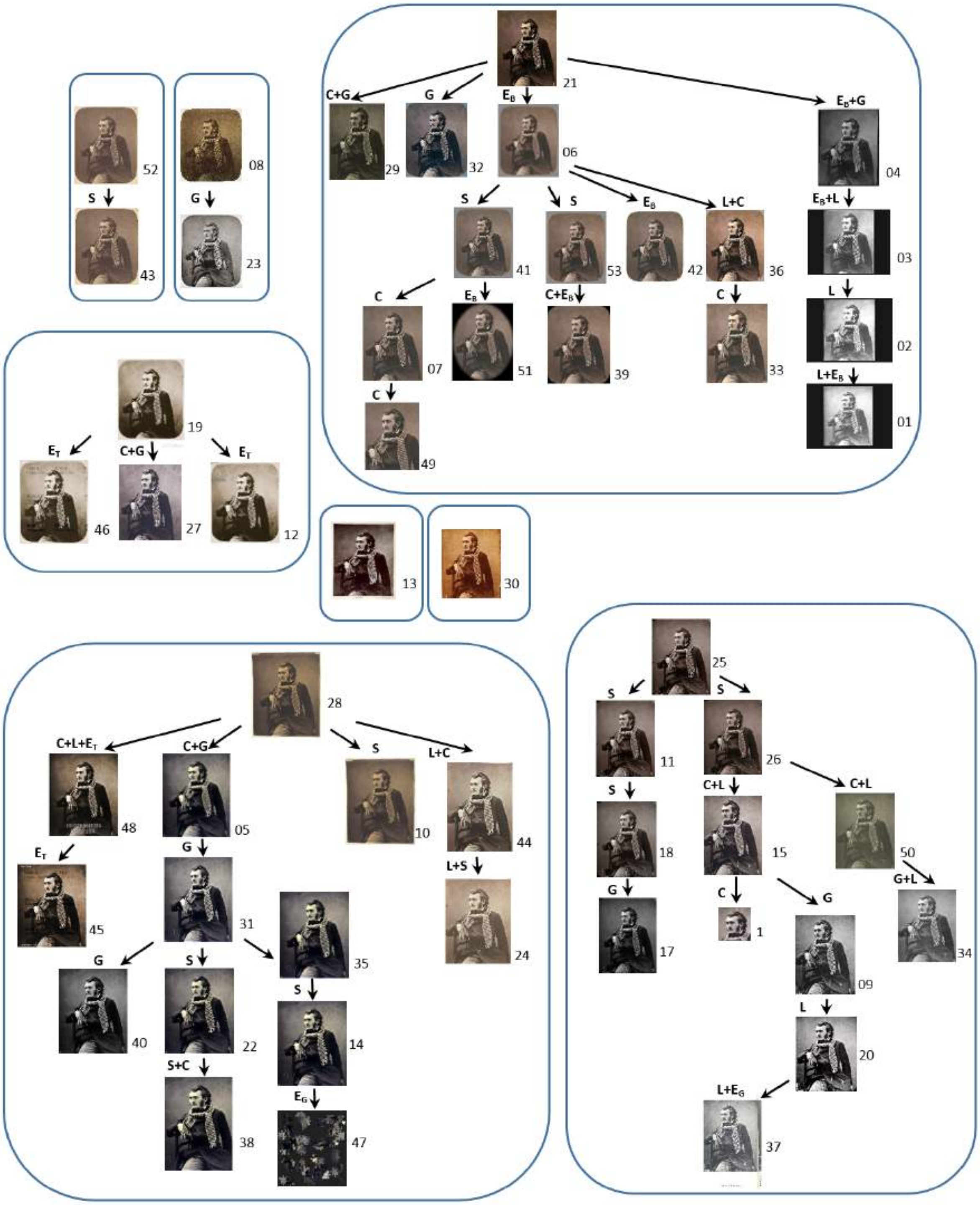


Fig. 8: Constituted copy graph of the third dataset of fig. 7 without using reference images.

Collections. *Information Visualization*, 14(3):183-203, 2015.

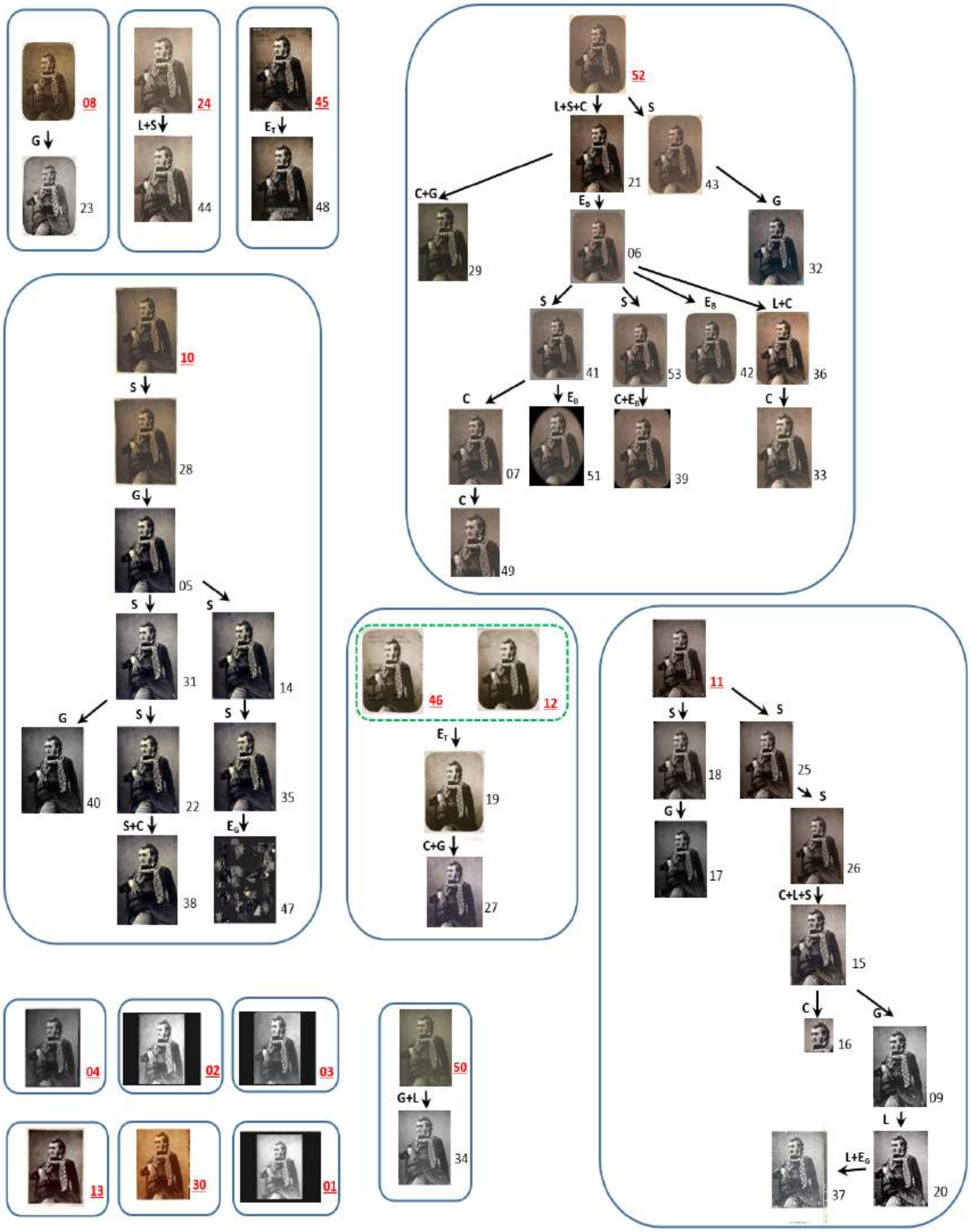


Fig. 9: Constituted copy graph of the third dataset of fig. 7 using reference images.

[33] G. Wolberg. Image Morphing: a Survey. *The visual computer*, 14(8):360-372, 1998.

[34] H. Xie, K. Gao, Y. Zhang, J. Li, Y. Liu and H. Ren. Effective and Efficient Image Copy Detection Based on GPU. *ECCV Workshops*, Part II, LNCS 6554, 338349, 2012.

[35] L. Yan, F. Zou, R. Guo, L. Gao, K. Zhou and C. Wang. Feature Aggregating Hashing for Image Copy Detection. *World Wide Web*, 19(2):217-229, 2016.

[36] J. Yang, J. Fan, D. Hubball, Y. Gao, H. Luo, W. Ribarsky and M. Ward. Semantic Image Browser: Bridging Information Visualization with Automated Intelligent Image Analysis. *IEEE Symposium on Visual Analytics Science And Technology*, 191-198, 2006.

[37] X. Yang, Q. Zhu and K-T. Cheng. Near-Duplicate Detection for Images and Videos. *ACM workshop on Large-scale Multimedia Retrieval and Mining*, 73-80, 2009.

[38] R. Yang, P. Kalnis and A.K.H. Tung. Similarity Evaluation on Tree-Structured Data. *ACM SIGMOD Int'l Conf. on Management of data*, 754-765, 2005.

[39] K. Zhang and D. Shasha. Simple Fast Algorithms for the Editing Distance Between Trees and Related Problems. *SIAM Journal on Computing*, 18:12451262, 1989.

[40] D-Q. Zhang and S-F. Chang. Detecting Image Near-Duplicate by Stochastic Attributed Relational Graph Matching with Learning. *ACM Int'l Conf. on Multimedia*, 877-884, 2004.

[41] D-Q. Zhang and S-F. Chang. Detecting Image Near-Duplicate by Stochastic Attributed Relational Graph Matching with Learning. *ACM Int'l Conf. on Multimedia*, 877-884, 2004.

Biography of 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.

Biography of Nathalie Casemajor:

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.

Biography of Aymen Talbi:

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.