

Exploring Image Dependencies: a New Challenge in Image Forensics

A. De Rosa^a, F. Uccheddu^a, A. Piva^a and M. Barni^b, A. Costanzo^b

^aUniversity of Florence, Dept. of Electronics and Telecommunications, Firenze, ITALY
{alessia.derosa, francesca.uccheddu, alessandro.piva}@unifi.it

^bUniversity of Siena, Dept. of Information Engineering, Siena, ITALY
barni@dii.unisi.it andreacos82@gmail.com

ABSTRACT

Though the current state of the art of image forensics permits to acquire very interesting information about image history, all the instruments developed so far focus on the analysis of single images. It is the aim of this paper to propose a new approach that moves the forensics analysis further, by considering groups of images instead of single images. The idea is to discover dependencies among a group of images representing similar or equal contents in order to construct a graph describing image relationships. Given the pronounced effect that images posted on the Web have on opinions and bias in the networked age we live in, such an analysis could be extremely useful for understanding the role of pictures in the opinion forming process. We propose a theoretical framework for the analysis of image dependencies and describe a simple system putting the theoretical principles in practice. The performance of the proposed system are evaluated on a few practical examples involving both images created and processed in a controlled way, and images downloaded from the web.

Keywords: image dependencies, image forensics, image registration, dependency test, dependency graph

1. INTRODUCTION

The advent of technologies easing the manipulation of digital images has inspired the parallel advent of tools for the analysis of image originality and content integrity verification. Such techniques, broadly termed as image forensic techniques, can help to establish the source of an image, to identify tampered images and distinguish between original and manipulated images.

Several approaches have been proposed in the last few years for studying digital images and capturing information about their history: starting from active approaches based on watermarking and digital signatures,¹ through passive approaches typical of image forensics. Specifically, a bunch of image forensics techniques have been developed that permit to extract information about the origin of the content,² or to detect the application of a wide variety of manipulations, including image resampling, single and double JPEG compression, cut & paste and slicing operations.^{3,4}

Though the current state of the art of image forensics permits to gather very interesting information about the *history* of an image, all the instruments developed so far focus on the analysis of single images. In several applications, though, the investigation of image dependencies, i.e. the relationships between a group of images, may be of similar, or even greater, importance. For instance, knowing how a set of images are related one to each other could allow the clustering of images originating from the same root image; in this way, it could be possible to discover that several images regarding a particular event have been actually produced from a limited set of source images. Such an information could then be used to understand the role of different web sites (and the groups behind them) in the formation of opinions on the web, permitting to identify opinion leaders, common feelings about specific events, and the preferred sources of information in a given temporal or geographical context. In other situations, knowing how a few source images have evolved into a large set of derived pictures, could allow to reconstruct how the usage of the information contained in the original images has evolved in time and space, thus permitting to identify, for instance, how these images have been used by groups of people with different opinions and cultures.

In this paper we propose a formalization of the above ideas from an image forensics perspective, and present a simple system for the detection of the dependencies between a set of images sharing similar or identical contents.

Specifically, in section 2 we elaborate on the concept of *image dependency*, by paying attention to distinguish the dependency between image contents from the dependency between the digital images representing the content. In section 3, we give a mathematical formalization of the concepts expressed in section 2. In section 4, we present a simple system to detect image dependencies. Some experimental results proving the validity of the system approach are discussed in section 5. The experiments consider both images generated in a controlled way and images downloaded from the web whose exact relationships are not known a-priori. The paper ends with a concluding section, where we outline some directions for future research.

2. FINDING DEPENDENCIES

Let us consider an event occurring in the real world. Such an event could be temporally and spatially extended however here we focus on an event occurring at a fixed time and seen from a particular viewpoint. Let us call such an event the *real scene*. Let us suppose that a set of images representing the same *real scene* is available. We are interested in finding the dependencies among such images in order to construct a sort of graph which could help to know how these images have been generated and how the information about the *real scene* contained in such images has evolved in time and space.

The first question we need to answer is: what does finding the dependencies between images mean exactly? Among all the possible meanings we can give to the term *image dependency*, we are interested to understand if a digital image has been produced by starting from another image representing the same *real scene*. Note that, since by definition the images correspond to the same content, the relationship we are looking for should not be related to the content itself; otherwise, all the images representing the same scene would be judged as dependent. In order to better clarify this concept, let us consider the example of two artists working on two different paintings; if they are free to paint any possible subject, then the possibility that the two painters draw the same topic is extremely low. In this case, a similar content could be taken as an evidence that some form of communication (or some dependency) between the painters occurred. On the contrary, if the subject of the paintings was imposed to the artists beforehand, then the fact that their paintings represent the same scene could not be taken as a demonstration that the artists communicated between them or that one of them copied the work of the other. But if they painted the same content by using exactly the same colors and the same pictorial metaphors, then we could conclude that the painters had some kind of contact or that one artist copied the other.

In this paper we are interested exactly in this situation, that is, we look for a form of dependency that does not rely on the semantic content of the images. More precisely, we will suppose that any image can be described as the composition of two parts: a part conveying the semantic information related to the *real scene* and a content-independent part representing the peculiarities of the process that produced the images. We will consider two images as dependent if some form of similarity exists between their content-independent components. In the next section we give a rigorous formalization of the above concept.

3. PROBLEM FORMALIZATION

We consider a set of color images \mathcal{I} , where an image $I \in \mathcal{I}$ is a $N \times M \times 3$ matrix, whose entries are integer values $\in [0, 255]$. We consider a set of fundamental image processing functions (f-IPFs) Φ_f , consisting of a number of functions ϕ_f , described as: $\phi_f(\cdot) : \mathcal{I} \times \wp_{\phi_f} \rightarrow \mathcal{I}$, where \wp_{ϕ_f} is the set of parameters characterizing the f-IPF. The domain \mathcal{D}_{ϕ_f} of ϕ_f is the set of input images on which the f-IPF can work, the codomain \mathcal{C}_{ϕ_f} is the set of output images defined as: $\mathcal{C}_{\phi_f} : \{I \in \mathcal{I} : \exists I^* \in \mathcal{I}, \exists p^* \in \wp_{\phi_f} : I = \phi_f(I^*, p^*)\}$.

Given two f-IPFs $\phi_1 : X \rightarrow Y$ and $\phi_2 : V \rightarrow Z$, they can be composed by firstly applying ϕ_1 to an argument x and then applying ϕ_2 to the result: $\phi_2(\phi_1(x))$, yielding a composite image processing function (c-IPF) $\phi_3 = \phi_2 \circ \phi_1 : X \rightarrow Z$. Note that the codomain of ϕ_1 must be included in the domain of ϕ_2 : $Y \subseteq V$. We define the set of composite image processing functions (c-IPFs) Φ_c as a set of all the possible compositions of a number r of f-IPFs $\in \Phi_f$, where r is the maximum order of the composition. In general, according to the application scenario, we will limit the analysis to a, possibly small, subset of image processing functions, namely $\Phi \subset \Phi_f \cup \Phi_c$. Finally, given an image $I_k \in \mathcal{I}$ and an image processing function $\phi \in \Phi$, we say that I_k is compatible with ϕ if $I_k \in \mathcal{C}_\phi$.

3.1 Dependency test

In order to formalize the dependency concept outlined in the previous section, we consider a set of images \mathcal{I} representing the same *real scene*. Our interest is to explore the pairwise dependency between such images. As we anticipated, we make the hypothesis that any image I belonging to \mathcal{I} can be univocally described as the composition of two separable and independent parts: $[I]_{\mathcal{C}}$ describing the content of the *real scene* and $[I]_{\mathcal{R}}$ representing the content-independent characteristics of the image, a sort of random part of the image:

$$I \leftrightarrow [[I]_{\mathcal{C}}, [I]_{\mathcal{R}}] \quad \forall I \in \mathcal{I}. \quad (1)$$

To verify the dependency between two images I_A and I_B , hereafter considered as two random information sources, we consider their mutual information:

$$\mathbf{I}(I_A; I_B) = H(I_A) - H(I_A|I_B), \quad (2)$$

where $H(I_A)$ is the entropy of the source I_A and $H(I_A|I_B)$ is the conditional entropy of the source I_A conditioned to I_B . By representing the images through the independent components introduced before, equation (2) can be rewritten as:

$$\mathbf{I}([I_A]_{\mathcal{C}}, [I_A]_{\mathcal{R}}; [I_B]_{\mathcal{C}}, [I_B]_{\mathcal{R}}) = H([I_A]_{\mathcal{C}}, [I_A]_{\mathcal{R}}) - H([I_A]_{\mathcal{C}}, [I_A]_{\mathcal{R}} | [[I_B]_{\mathcal{C}}, [I_B]_{\mathcal{R}}]) \quad (3)$$

$$= H([I_A]_{\mathcal{C}}) + H([I_A]_{\mathcal{R}}) - H([I_A]_{\mathcal{C}} | [I_B]_{\mathcal{C}}, [I_B]_{\mathcal{R}}) \quad (4)$$

$$- H([I_A]_{\mathcal{R}} | [I_B]_{\mathcal{C}}, [I_B]_{\mathcal{R}}, [I_A]_{\mathcal{C}}) \quad (5)$$

$$= H([I_A]_{\mathcal{C}}) + H([I_A]_{\mathcal{R}}) - H([I_A]_{\mathcal{C}} | [I_B]_{\mathcal{C}}) - H([I_A]_{\mathcal{R}} | [I_B]_{\mathcal{R}}) \quad (6)$$

$$= \mathbf{I}([I_A]_{\mathcal{C}}; [I_B]_{\mathcal{C}}) + \mathbf{I}([I_A]_{\mathcal{R}}; [I_B]_{\mathcal{R}}), \quad (7)$$

where (5) has been obtained by exploiting the independence between $[I_A]_{\mathcal{C}}$ and $[I_A]_{\mathcal{R}}$ and the chain rule⁵ and (6) has been obtained by exploiting the independence between $[I_A]_{\mathcal{C}}$ and $[I_B]_{\mathcal{R}}$ and the independence between $[I_A]_{\mathcal{R}}$ and $[I_A]_{\mathcal{C}}$ and $[I_B]_{\mathcal{C}}$. The mutual information between the images can thus be expressed as the sum of the mutual information between the \mathcal{C} components and the \mathcal{R} components. For our analysis, we only consider the second term (the content-independent one), since the first term will never be null, due to the intrinsic dependency between the \mathcal{C} components (since they refer to the same real scene).

We can now cast the problem of determining the dependency between I_A and I_B as a hypothesis testing problem, in which we want to test the hypothesis that $[I_A]_{\mathcal{R}}$ and $[I_B]_{\mathcal{R}}$ are independent, that is we want to test whether $\mathbf{I}([I_A]_{\mathcal{R}}; [I_B]_{\mathcal{R}}) = 0$. The design of an optimal criterion for such a test would require the availability of a good statistical model to describe $[I_A]_{\mathcal{R}}$ and $[I_B]_{\mathcal{R}}$ and their possible relationship through the functions contained in Φ . Modeling such a relationship is very complicated, hence we will use a simplified strategy.

More precisely, by considering a set of image processing functions Φ , we make the assumption that if there is some relationship between two images $I_A, I_B \in \mathcal{I}$, then one of the two images can be obtained, at least approximately, by applying a $\phi_j \in \Phi$ to the other. A possibility, then, would be to compute the correlation coefficient ρ_j between $[I_B]_{\mathcal{R}}$ and $[\phi_j(I_A)]_{\mathcal{R}}$, for each $\phi_j \in \Phi$, and use as decision statistic the maximum value ρ_{max} in this set of correlations:

$$\rho_{max} = \max_{\phi_j \in \Phi} \rho([I_B]_{\mathcal{R}}, [\phi_j(I_A)]_{\mathcal{R}}) \quad (8)$$

where the maximization over the parameters in \wp_{ϕ_j} has been omitted for simplicity. Let us note that the previous statistic is voluntarily asymmetric, i.e. it tests the dependency of I_B on I_A and not the other way round. Looking for all possible $\phi \in \Phi$ requires a huge computational effort, all the more that for each function all the parameters in \wp_{ϕ} should be considered. In addition, the probability of detecting a false dependency would increase with the number of functions and with the size of the parameter space. In order to overcome the above problems, we devised a different strategy: we try to *guess* the function $\phi^* \in \Phi$ that has been used to pass from I_A to I_B by relying on the content part of the images. Suppose, for instance, that the set Φ contains only the *rotation* f-FIP, i.e. a function that rotates the input image by a certain angle. Instead of computing the correlation coefficient between the random part of I_B and the random part of all the rotated versions of I_A , the most likely

rotation angle is estimated by relying on $[I_A]_{\mathcal{C}}$ and $[I_B]_{\mathcal{C}}$. Then the correlation coefficient between the random part of I_B and the random part of I_A rotated by the estimated angle is computed and used as decision statistic. Assuming that an efficient way exists to estimate the rotation angle, this approach is much faster than searching exhaustively for ρ_{max} as in equation (8). To finally accept/reject the hypothesis of independence we compare the correlation coefficient ρ^* between $[I_B]_{\mathcal{R}}$ and $[\phi^*(I_A)]_{\mathcal{R}}$ with a suitable threshold T : if ρ^* is lower (res. higher) than the threshold then the images are judged as independent (dependent). The threshold T should be set rigorously by studying the statistical characteristics of ρ^* and by fixing a value for the false positive probability. Alternatively an empirical analysis may be carried out and T determined experimentally.

3.2 Dependency graph

The final aim of our analysis is the construction of the so-called dependency graph, i.e. a graph representing the relationships between the images in \mathcal{I} . In particular, each node of the graph will represent an image of \mathcal{I} , and the presence of an oriented edge linking two nodes (e.g. an edge going from I_A to I_B) will indicate that I_B has been produced by starting from I_A . Edges could be labeled with the image processing function that was used to pass from I_A to I_B and with the corresponding value of the correlation coefficient. The graph seems a natural choice to synthetically represent image dependencies: by using a graph we inherit all the existing algorithms for querying the history, in terms of relations, of a picture in a specific context. By giving the graph a semantic nature/setting we could also build a set of rules (an *ontology*) to infer other relations between images. For instance, if we design the graph according to the so called semantic web principles, its links and ontology could be shared to other web applications.

To actually construct the graph, the dependency test values for each pair of images in \mathcal{I} are collected (either by computing ρ_{max} or ρ^*), and a first version of the dependency graph is built by taking only those oriented links for which the correlation is above the threshold T . By critically observing the whole dependency graph, it is possible to modify it to resolve ambiguous situations that could not be disambiguated by a pairwise analysis. For instance, when two nodes are connected by two edges (oriented in opposite directions), the weakest link could be removed. A similar strategy could be applied to avoid the presence of loops in the graph. More generally, an *ontology* could allow the system to automatically infer higher order relations between images and to solve ambiguous situations.

4. SYSTEM ARCHITECTURE

In this section we describe a practical system implementing the ideas expressed so far in a realistic, yet rather simple, scenario. We first introduce the general structure and then discuss the various parts of the system.

4.1 Basic assumptions

As a first assumption we suppose to work only with JPEG images, that is after any processing step the manipulated image is JPEG compressed with an arbitrary quality factor.

By keeping the same notation we used in the previous sections, our instantiation of the set Φ includes the concatenation of only 3 elementary functions: geometric transformation, including scaling, rotation and cropping (let us indicate this function by ϕ_g), color manipulation, e.g. color transfer or histogram equalization (ϕ_c) and JPEG compression (ϕ_J). We believe that the combination of these simple manipulations can cover most of the activities of a “typical image user”. It goes without saying that our system could be extended to incorporate a larger set of modifications. For example, given a photomontage we could try to detect the presence of tampering and then feed original and tampered regions to the system separately as different images.

As a further working assumption we suppose that all the images have been processed by the consecutive application of ϕ_c , ϕ_g and ϕ_J . The case in which one or more of these functions have not been applied is handled by adjusting the parameters of the missing function(s), so that its (their) effect is null. It is up to the system to estimate the parameters of ϕ_c , ϕ_g and ϕ_J by relying on the \mathcal{C} components of the images under inspection. Once the parameters of these functions have been estimated, the functions are applied to the source image and the decision statistic ρ^* calculated and compared against the decision threshold.

As required by the formalization given in section 3, the images are seen as the composition of two contributes, the content component $[I]_C$ and the randomness $[I]_{\mathcal{R}}$. In our implementation, such contributions are computed by implementing the wavelet domain denoising algorithm proposed by Mihkaç et al,⁶ with the denoised image corresponding to $[I]_C$ and the noise to $[I]_{\mathcal{R}}$.

4.2 System overview

Figure 1 provides an overview of the general architecture of the system we used to understand whether I_B could have been generated from I_A or not. To do so we choose the parameters of the ϕ_c , ϕ_J and ϕ_g functions to be applied to I_A in such a way that after their application I_A is as similar as possible to I_B .

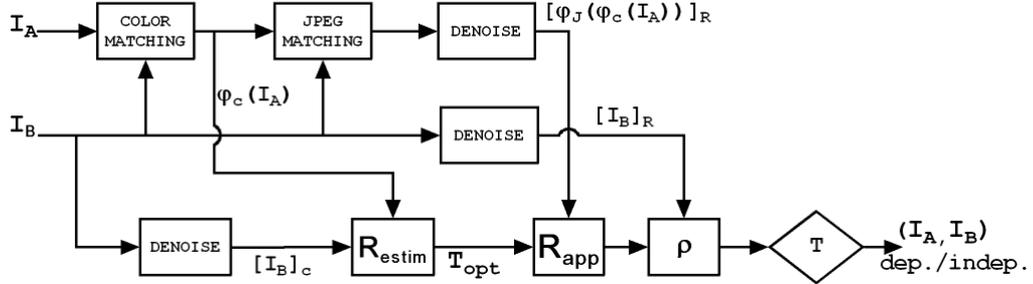


Figure 1. Scheme of the proposed system. R_{estim} indicates the registration process that we perform on image contents to estimate the best parameters for the function ϕ_g ; R_{app} applies such a transformation to $[\phi_J(\phi_c(I_A))]_{\mathcal{R}}$; the block indicated by ρ computes the correlation coefficient between the registered randomness and $[I_B]_{\mathcal{R}}$; T indicates the comparison of ρ against the decision threshold.

In the first block (color matching) the image I_A is modified so that its colors correspond to the colors of I_B . According to our notation, the output of the color matching block is indicated as $\phi_c(I_A)$. Then the color-matched version of I_A is JPEG-compressed by using the same quality factor of the image I_B . The output of the JPEG-matching block, namely $\phi_J(\phi_c(I_A))$ is split into its content and random parts. At this point a geometric transformation is applied to $[\phi_J(\phi_c(I_A))]_{\mathcal{R}}$ in order to align it with $[I_B]_{\mathcal{R}}$. The parameters of the geometric transformation are estimated by relying on the content parts of I_B and $\phi_c(I_A)$ in the R_{estim} block. The estimation steps works exclusively on the content to avoid that the correlation between the random parts is artificially increased as a consequence of the registration process. Finally, the system calculates the correlation coefficient ρ^* between the registration output and $[I_B]_{\mathcal{R}}$. To accept or reject the hypothesis of dependence ρ^* is compared with a threshold T .

One may wonder why the geometric registration process is applied as a last step after JPEG compression and denoising. With regard to denoising we may argue that the order of geometric registration and denoising is not particularly important given that the two steps are approximately commutative. However, this is not the case with the JPEG compression. Indeed we found experimentally that applying the registration step before the JPEG matching block results in a rather high false detection rate, i.e. several false dependencies are detected. While the exact reason for this phenomenon is not clear, a possible explanation is that the JPEG compression introduces a signature that will be mostly contained in the random component and will match with the corresponding signature present in $[I_B]_{\mathcal{R}}$, thus artificially increasing the value of ρ^* .

In the following we give a more detailed description of each of the above blocks.

4.3 Color matching

The goal of the color matching block is to transfer the color characteristics of I_B (hereafter referred to as target image) to I_A (source image). Both source and target images are three channels RGB images. Our method for the color matching relies on a slightly modified version of the algorithm proposed by Reinhard et al.⁷ In a nutshell, Reinhard et al's algorithm for the color transfer works as follows. In the RGB color space the image channels present a very strong correlation and this makes color operations difficult to perform. In order to overcome this

problem it is necessary to migrate to a color space where axes have very low or no correlation at all. While there are several of these spaces, the paper proposes to use the decorrelated color space $l\alpha\beta$. The conversion from RGB to $l\alpha\beta$ is obtained with an intermediate step (the *LMS* color space). Each conversion is performed by multiplying every pixel of the image by a 3×3 matrix whose coefficients are given in the paper.⁷

In the $l\alpha\beta$ space some characteristics of the target's color distribution are passed to the source image. Such characteristics include mean and standard deviation of each of the three axes, computed separately for both images. The first step consists in subtracting the mean from each channel of the source image:

$$\begin{aligned} l^* &= l_s - \bar{l}_s \\ \alpha^* &= \alpha_s - \bar{\alpha}_s \\ \beta^* &= \beta_s - \bar{\beta}_s. \end{aligned} \quad (9)$$

Then source channels are scaled by factors corresponding to the ratio of target and source standard deviations:

$$\begin{aligned} \hat{l} &= (\sigma_t^l / \sigma_s^l) l^* \\ \hat{\alpha} &= (\sigma_t^\alpha / \sigma_s^\alpha) \alpha^* \\ \hat{\beta} &= (\sigma_t^\beta / \sigma_s^\beta) \beta^*. \end{aligned} \quad (10)$$

Finally, mean values of the target image channels are added to the scaled source and the resulting color matched image $\phi_c(I_A)$ is transferred back to the RGB space.

4.4 Compression matching

As we already said, our system works only with images stored in JPEG format. In this way the JPEG matching block may read all the information regarding the way the images have been compressed directly from the JPEG files. Specifically, the system can access the DCT coefficients, the quantization tables, the Huffman coding tables, color space informations and comment markers. This allows to estimate the compression quality factors of the images I_A and I_B , respectively qf_A and qf_B . The compression matching block, then, simply compresses $\phi_c(I_A)$ with qf_B obtaining $\phi_j(\phi_c(I_A))$.

4.5 Image registration

Image registration is commonly used in a vast variety of applications like object recognition, motion tracking, segmentation, biomedical imaging, remote sensing. In our system we used the registration method developed by Sorzano et al.⁸ and further refined by Arganda-Carreras⁹ to which we refer for a more detailed overview of the theory behind the algorithm. In a nutshell, this algorithm for elastic registration tries to find a deformation field that transforms the coordinates of the source image into the coordinates of target image. The deformation field is calculated by means of B-splines, that are rather inexpensive from a computational point of view and return good results. Such algorithm works on intensity values of image pixels and uses a similarity metric defined by the energy functional E shown in equation (11):

$$E = w_i \cdot E_{img} + w_\mu \cdot E_\mu + (w_d \cdot E_{div} + w_r \cdot E_{rot}) + w_c \cdot E_{cons}. \quad (11)$$

This energy functional consists of 5 contributes: the energy of the error (dissimilarity) E_{img} between target and warped source; the error in mapping the user defined landmarks E_μ ; the regularization terms E_{div} and E_{rot} that control the smoothness of the deformation; a consistency term E_{cons} that evaluates the geometric consistency between direct deformation (source to target) and inverse deformation (target to source). The effect of each energy component on the final result is controlled by its respective weight. We will describe in the next section the values we assigned to such weights. Finally, the adopted optimization method is an enhanced version of the classic Marquardt-Levenberg minimization.

The performance of the registration step are improved by using SIFT features¹⁰ as landmarks. The idea is to use SIFT features as anchors for the registration algorithm to increase its accuracy. We are aware that a landmark-based registration can be sensitive to noise or to inaccurate landmarks, however in our case this problem is mitigated by the fact that we work with images with very similar contents. Scale Invariant Feature

Transform (SIFT) is an algorithm for the extraction of invariant features from an image. It has been proposed by Lowe¹⁰ and further refined by Alhwarin et al.¹¹ Such algorithm has been used with success in many computer vision applications because of its invariance to image scaling, rotation, translation and partially to change of illumination and perspective. In particular, we are interested on one of these applications, that is finding a set of corresponding points of interest in two images with similar visual content. In brief, the SIFT algorithm consists of 3 steps: detection of good candidate features (*keypoints*), suppression of weak and unstable candidates and matching. Candidate features are determined as local maxima and minima of Difference of Gaussian pyramids. Such features are usually too many and some of them are unstable and sensitive to noise. In particular, the SIFT algorithm discards features with low contrast and features along the image edges. In this step robustness to rotation is achieved by computing one or more orientations for keypoints based on the directions of local image gradient.

5. EXPERIMENTAL RESULTS

To evaluate the effectiveness of our system, we have built 3 different case studies with a limited set of images \mathcal{I} representing the same *real scene*. In the next subsections we first describe the construction of the case studies, then we discuss the settings we used for each of the blocks the system consists of, and finally we present and comment the experimental results.

Our system has been mainly implemented in a Matlab® environment. However, we delegated some of the most time and memory consuming tasks to the Java-based image processing software *ImageJ*. This software is designed with an open architecture that is extensible via custom plugins. We used two of these plugins respectively for SIFT features matching and for image registration. Communication between the two environments has been made possible by the Matlab-ImageJ (MIJ) interface that exploits Matlab’s JVM support. Table 1 shows where such tools can be downloaded.

Table 1. URLs at which used software is freely available for download.

ImageJ	http://rsbweb.nih.gov/ij
bUnwarpJ	http://biocomp.cnb.uam.es/~iarganda/bUnwarpJ
SIFT	http://pacific.mpi-cbg.de/wiki/index.php/Feature_Extraction
MIJ	http://bigwww.epfl.ch/sage/soft/mij
JPEG Toolbox	http://www.philsaltee.com/jpegtbx/index.html

5.1 Construction of case studies

In our first two case studies the set \mathcal{I} consists of 10 RGB images. I_1 and I_2 are two independent natural images taken by a digital camera with its native JPEG compression not considered as a processing. These images are rather large (2048×1536 pixels), so in order to keep the computational burden under control we subsampled them, for a final size of 410×308 . Given N images, in fact, we need to apply our scheme N^2 times, with an average processing time of 4 minutes for full size images and 18 seconds for the resized images. This choice does not represent a limitation mainly for two reasons: first, because one of our goals is to test the system on a real Internet case, where most of the times images are small in size. Second, some exploratory results obtained on the full resolution images provided very similar results to those obtained on the subsampled images. The remaining 8 images (I_3 to I_{10}) have been obtained by post processing I_1 and I_2 through the Matlab® functions reported in table 2. As we previously stated, in both cases we are taking into account only a subset of possible image processing functions including color transformations, scaling, rotation and JPEG compression. The original dependency graph (the same for both both cases) is shown in figure 3.

In our first case study there is a slight change of perspective of the camera that took the two independent pictures. This leads to the creation of an example that we consider an *easy* case study: in fact the images generated by starting from I_1 and I_2 inherit the difference of content between I_1 and I_2 thus making the work

of the dependency detector easier. The second case study can be seen as the worst possible scenario: I_1 and I_2 have been taken by the same camera with no change of point of view and show exactly the same visual content. Figure 2 shows I_1 and I_2 for the two case studies.



Figure 2. Independent images. From left to right: I_1 for first case study; I_2 for first case study; I_1 for second case study; I_2 for second case study.

The third case study represents an attempt to test our system on a set of images that have been downloaded from various locations on the Web, for which we obviously do not have a ground truth. We collected 20 images with different sizes, ranging from 268×400 to 800×951 pixels, and different JPEG compression quality factors but with the same visual content: the famous painting *The girl with a pearl earring* of Johannes Vermeer (1665). The 20 images used for the third case study are shown in figure 4. Note that the set includes also some outliers, namely an amateur copy of the painting, and 3 photos of Scarlet Johansson, the actress who played the role of the Girl with a pearl earring in the homonym movie of 2003. The image data sets corresponding to the three case studies are freely available for download at the address <http://clem.dii.unisi.it/~vipp/files/SPIE2010>.

Table 2. Type of processing for each of the dependent images of the set \mathcal{I} . Each processing is a combination of 3 functions: color operations ϕ_c , geometric operations ϕ_g and JPEG compression ϕ_j . Note that JPEG compression always closes the processing, while color and geometric operations have been used in any order. For each processing ϕ_i ($i=1,2,3$) we also report the parameters p_{ϕ_i} we used (if any). For the compression we report the quality factor used by the JPEG coder.

I	ϕ_1	p_{ϕ_1}	ϕ_2	p_{ϕ_2}	ϕ_3	p_{ϕ_3}
I_3	Compression of I_1	50%	-	-	-	-
I_4	Histogram stretch of I_2	-	Compression	70%	-	-
I_5	Rotation of I_3	3°	Compression	100%	-	-
I_6	Scaling of I_3	1.1	Compression	100%	-	-
I_7	Histogram stretch of I_2	-	Scaling	1.2	Compression	90%
I_8	Compression of I_7	60%	-	-	-	-
I_9	Rotation of I_3	-2°	Compression	100%	-	-
I_{10}	Scaling of I_4	0.9	Compression	100%	-	-

5.2 System settings

Our system features a large number of configuration parameters. However we did not observe a strong dependence upon most of these parameters, with the exception of a small subset that drastically modifies the behavior of the system. We now briefly explain the meaning of such parameters and the values we used. For a more in-depth explanation of the various parameters we refer to their respective papers.^{7,10,12}

We performed the color matching in the $l\alpha\beta$ color space but unlike suggested by the authors⁷ we worked on linear rather than logarithmic values. Moreover, we kept the luminance channel l out of the processing.

For the SIFT algorithm we used the values suggested by Lowe¹⁰ with the exception of two of them: *steps per scale octave* and *minimum image size*. Keypoint candidates are extracted at all scales between maximum

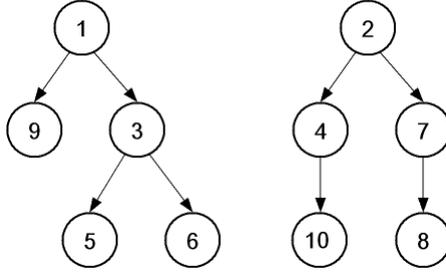


Figure 3. The original graph representing dependencies among the ten images of the first two case studies. Each node of the graph corresponds to an image, each oriented edge to the relationship between two images.

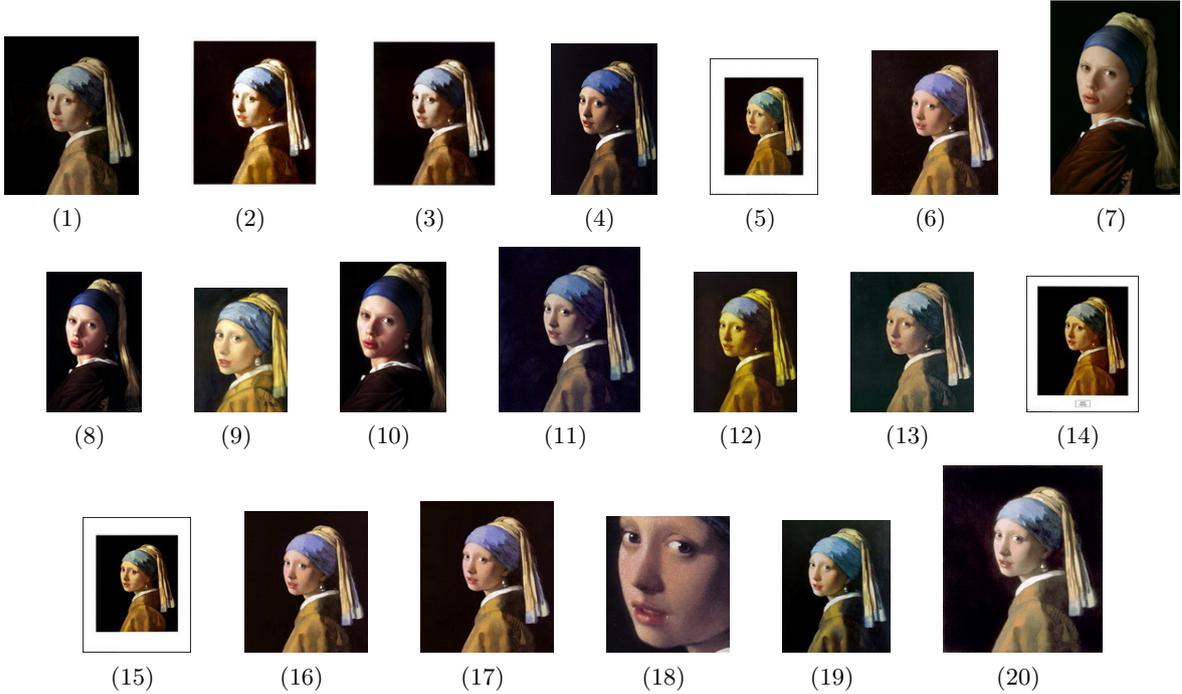


Figure 4. Dataset of 20 images for the Internet case study. For sake of clarity we have downsampled those images that are too large to fit the page. These images are: I_6 (714×822); I_7 (644×980); I_{17} (667×768); I_{19} (541×672); and I_{20} (800×951).

image size and minimum image size. This scale-space is represented in octaves each covering a fixed number of discrete scale steps. In order to improve performances with images that present very different zooming factors we increased *steps per scale octave* to 4 and lowered *minimum image size* to 16.

The registration step is the one that required the most testing efforts. The registration algorithm we used is a very powerful tool that takes into account not only rigid body operations but also affine transformations and local warping. Given that we are mainly interested in rigid deformations, we set the parameters of the algorithm so to prefer rigid transforms. The system allows also to perform a bi-directional registration to evaluate how consistent the direct and inverse transformations are. However, we did not use such feature, since we focused only on one registration at a time. As we explained in section 4.5, the behavior of the registration algorithm depends on a set of weights \mathbf{w} for which we have chosen the following values: $w_{div} = 30$ and $w_{curl} = 30$ (the maximum value for a stable behavior according to the original paper¹²) to penalize the smoothing of the estimated deformation; $w_{landmark} = 1.0$ and $w_{img} = 0.4$; finally, we can assign to w_{cons} any value because it does not have any impact on system's behavior when performing a mono-directional registration. In our implementation we have chosen 0. The parameters *initial deformation* and *final deformation* determine the level of detail of deformations and

refer to the type of grid used by the algorithm. The choice of their values depends on how misaligned are the images we are registering. We opted for the medium misalignment configuration (*Very coarse* and *Fine*).

5.3 Results and discussion

For each case study we calculated the correlation coefficient ρ^* between each couple of images of the set \mathcal{I} and we built the correlation matrix C . Such matrix is a square matrix $n \times n$ where n is the number of images. Each element $C(i, j)$ corresponds to the correlation $\rho_{i,j}^*$ between images i and j . Starting from C we computed the dependency graph. To build the graph we compared all the elements of C with a threshold T and discarded the links with a correlation lower than the threshold. At this point the newly constructed graph is not yet completed and needs further processing. Specifically, we applied some simple ontology rules (such rules can be easily extended to deal with more complex situations). We first removed direct loops: it is not possible, in fact, that image i generated image j and viceversa at the same time. In this case we kept only the link with the highest correlation coefficient. It is also not possible that an image has been generated by two different images. This might be true in presence of a photomontage, but we did not include such possibility in our scenario. Therefore, if we noticed an image with more than one parent, we simply kept the parent with the highest correlation coefficient. The threshold value was derived empirically by visual inspection of C . In our experiments we used $T = 0.5$ for the first two case studies and $T = 0.25$ for the third experiment. In the following we present the results we obtained in the 3 cases. For each case we report and comment the output dependency graph.

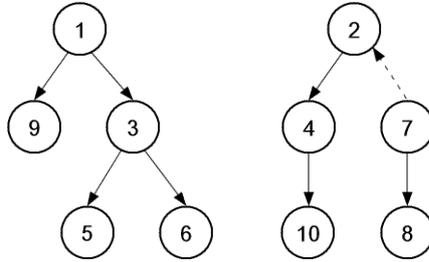


Figure 5. Result for the case studies 1 and 2. All relationships are correctly retrieved with the exception of the link between I_2 and I_7 presents a wrong orientation.

In the first and second case study our system found all the original relationships between the sets of images. However, one of them is not correct (see figure 5): while the system claims I_7 to be parent of I_2 , it is actually the opposite that is true. Nevertheless, this result is not totally wrong because the system was still able to find that the two images are directly linked to each other. Indeed after application of the threshold, both the links from I_2 to I_7 and that from I_7 to I_2 survive, however the latter (wrong) link is stronger than the true one, thus resulting in the reported error. To give an idea of the correlations we found among images the correlation matrix C of the second case study is shown in the following. Similar results are obtained with the first case study.

$$C = \begin{pmatrix} 0.9709 & 0.4391 & 0.9648 & 0.3286 & 0.2351 & 0.2305 & 0.3536 & 0.1774 & 0.6312 & 0.3832 \\ 0.4427 & 0.9788 & 0.2487 & 0.9627 & 0.2200 & 0.2188 & 0.4314 & 0.2216 & 0.4583 & 0.4088 \\ 0.3239 & 0.2720 & 0.9865 & 0.2440 & 0.5520 & 0.5719 & 0.1842 & 0.1599 & 0.2393 & 0.2688 \\ 0.3798 & 0.4718 & 0.2197 & 0.9881 & 0.2005 & 0.2000 & 0.3006 & 0.2226 & 0.2834 & 0.6255 \\ 0.2244 & 0.1920 & 0.2227 & 0.1623 & 0.9846 & 0.4104 & 0.1341 & 0.1057 & 0.1945 & 0.1996 \\ 0.2256 & 0.2167 & 0.2496 & 0.1834 & 0.4425 & 0.9758 & 0.1805 & 0.1408 & 0.2048 & 0.2336 \\ 0.4339 & 0.5232 & 0.1499 & 0.2893 & 0.1693 & 0.1966 & 0.9767 & 0.9582 & 0.3294 & 0.3407 \\ 0.2788 & 0.3178 & 0.1504 & 0.2681 & 0.1449 & 0.1659 & 0.4374 & 0.9929 & 0.2218 & 0.2937 \\ 0.6047 & 0.3785 & 0.1589 & 0.2126 & 0.1959 & 0.1932 & 0.2308 & 0.1397 & 0.9806 & 0.2828 \\ 0.3620 & 0.3725 & 0.1675 & 0.3074 & 0.2233 & 0.2286 & 0.2596 & 0.1552 & 0.3027 & 0.9770 \end{pmatrix}$$

Finally, we have tested the system on the Internet case study for which we do not know the real dependency graph. In this case we can only evaluate if the results are plausible by visual inspection. The 20 images cluster into different groups corresponding to disjoint subgraphs. In figure 6 the clusters formed by more than one

images are shown (we have labeled each resulting group with letters from (a) to (d)). Cluster (d) consists only of one image, however we have shown it in the figure since it corresponds to an outlier that clearly does not have any relationships with the other images. Let us consider the most complex cluster, i.e. cluster (a). From a visual inspection we can not say whether I_{12} could have generated I_1 and I_3 . However, we can point out the fact that I_{12} 's colors are rather different than those of the original painting, hence manipulations to correct them are highly probable, thus making the link found by the system plausible. Moreover, I_2 appears to be the result of a brightness correction of I_3 . Finally images I_5 , I_{14} and I_{15} indeed form a cluster (though the link with I_1 is a bit doubtful). Again we can tell that upon visual inspection: these three images are the only that show a white frame around the painting and a label under it. As for cluster (b) it is plausible that I_{17} generated I_{16} : they show exactly the same content and colors but I_{17} has a higher resolution. In cluster (c) we have 3 images, namely I_7 , I_8 and I_{10} , that are not pictures of the original painting but photos of the movie actress. In this case the system clustered correctly these images and found out that I_7 is the "original". Moreover, I_7 generated I_8 and I_8 generated I_{10} . Finally, as we can clearly see from figure 4, image I_9 is an amateur imitation of the real painting, therefore no image of the set should be related to it. All the other images have been considered as stand-alone images with no dependency relationships with the other images in the set.

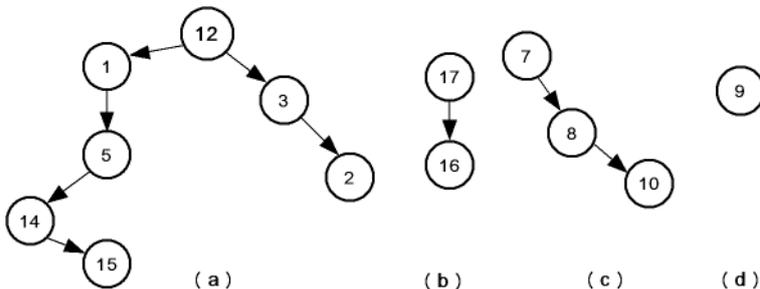


Figure 6. Dependency graph for the Internet case study. For sake of clarity we omit those images that do not share any kind of relationships, with the exception of I_9 .

6. CONCLUSIONS

The use of image forensics tools to discover the relationships between groups of images with the same or similar content may find interesting applications in diverse fields, including tracing the illegal distribution of copyrighted images on the internet and understanding how images contribute to the formation and evolution of opinions over the internet. In particular, the latter application opens a new frontier in image forensics research for the difficult challenges it poses both from an image processing and a cognitive point of view. In this paper we focused on the image processing aspects of the problem, with two main contributions: i) we proposed a rigorous formalization of the problem, and ii) we presented a simple system putting some of our ideas into practice. We carried out a few experiments on synthetic and real-world data, obtaining promising results, that encourage further research in this area.

There is no need to say that many difficult challenges are still ahead of us, including: the development of a theoretically sound formulation of the hypothesis testing problem lying at the heart of our system, going beyond the heuristic approach we used in our implementation; the development of a practical system that includes a wider variety of manipulations with particular emphasis on photomontages; the construction of an ontology capable of inferring higher order relationships starting from a pairwise analysis; and, finally, the use of the output of the forensic analysis at a higher level to infer information about the mechanisms whereby images contribute to form opinions and the way images may be used and are actually used to bias opinions in networked communities.

ACKNOWLEDGMENTS

This work has been partially supported by the project LivingKnowledge - Facts, Opinions and Bias in Time funded by the European Commission under contract no. 231126.

REFERENCES

- [1] Kundur, D. and Hatzinakos, D., “Digital watermarking for telltale tamper proofing and authentication,” *Proceedings of the IEEE* **87**, 1167–1180 (July 1999).
- [2] Lukas, J., Fridrich, J., and Goljan, M., “Digital camera identification from sensor pattern noise,” *IEEE Trans. on Information Forensics and Security* **1**, 205–214 (February 2006).
- [3] Popescu, A. C. and Farid, H., “Statistical tools for digital forensic,” in [*Proc. 6th Int. Work. on Information Hiding, IH’04*], **3200**, 128–147, Springer-Verlag, Toronto, Canada (May 2004).
- [4] Swaminathan, A., Wu, M., and Liu, K., “Digital image forensics via intrinsic fingerprints,” *IEEE Trans. on Information Forensics and Security* **3**, 101–117 (January 2008).
- [5] Cover, T. M. and Thomas, J. A., [*Elements of Information Theory*], John Wiley & Sons, New Jersey (Second Ed., 2006).
- [6] Mihçak, M. K., Konitsev, I., and Ramchandran, K., “Low-complexity image denoising based on statistical modeling of wavelet coefficients,” *IEEE Signal Processing Letters* **6**, 300–303 (December 1999).
- [7] Reinhard, E., Ashikhmin, M., Gooch, B., and Shirley, P., “Color transfer between images,” *IEEE Computer Graphics and Applications* **3**, 34–41 (September 2001).
- [8] Sorzano, C. O. S., Thevenaz, P., and Unser, M., “Elastic registration of biological images using vector-spline regularization,” *IEEE Transactions on Biomedical Engineering* **52**, 652–663 (April 2005).
- [9] Arganda-Carreras, I., Sorzano, C. O. S., Marabini, R., Carazo, J. M., de Solorzano, C. O., and Kybic, J., “Consistent and elastic registration of histological sections using vector-spline regularization,” in [*Computer Vision Approaches to Medical Image Analysis*], *Lecture Notes in Computer Science* **4241**, 85–95, Springer Berlin / Heidelberg (May 2006).
- [10] Lowe, D. G., “Distinctive image features from scale-invariant keypoints,” *IEEE Computer Graphics and Applications* **60**, 91–110 (January 2004).
- [11] Alhwarin, F., Wang, C., Ristić-Durrant, D., and Gräser, A., “Improved sift-features matching for object recognition,” in [*BSC International Academic Conference*], 179–190 (2008).
- [12] Arganda-Carreras, I., “bunwarpj: Consistent and elastic registration in imagej, methods and applications,” *Second ImageJ User & Developer Conference* (2008).