A perceptual image hashing algorithm for hybrid document security

Sébastien Eskenazi, Boris Bodin, Petra Gomez-Krämer, Jean-Marc Ogier

A perceptual image hashing algorithm for hybrid document security

Sébastien Eskenazi, Boris Bodin, Petra Gomez-Krämer, and Jean-Marc Ogier
L3i, University of La Rochelle, Avenue Michel Crépeau, 17042, La Rochelle, France Avenue Michel Crépeau - 17000 La Rochelle

Email: sebastien.eske@hotmail.com, {petra.gomez, jean-marc.ogier}@univ-lr.fr

***Abstract*—In order to create an automatic document security system one needs to secure the textual content but also the graphical content of the document. This paper proposes a hashing algorithm capable of securing the graphical parts of paper and digital documents with unprecedented performance and a very small digest. The main challenge for such an algorithm is that of stability, in particular with respect to print and scan noise. We define the generic notion of stability and how to evaluate it. To achieve such performance we use both dense local information and global descriptors. We have tested our method on two datasets totaling nearly 45000 images.**

With the ever increasing digitization of our world, document fraud is becoming a significant issue. For instance, an easy way to get a fake identity card is not to forge one, but to obtain a real one with a fake electricity bill and a fake birth certificate [1]. These documents frequently change between a paper and a digital format. Ensuring the security of these documents even if they change format is called hybrid security.

In the context of the content-based signature for hybrid security of [2] and to authenticate an image or the graphical parts of a document, there is a need for an adequate perceptual image hashing algorithm that is both stable for print and scan noise and precise enough to detect image modifications. Both stability and precision are critical to ensure a proper security.

Perceptual based image hashing was introduced by Schneider and Chang [3]. Figure 1 shows a generic perceptual image hashing algorithm. It extracts the robust features from the image to generate a compact representation, the digest/hash. This digest can then be encrypted to make a signature. One can compute a similarity measure between two image digests or signatures to verify if the images are similar or not. We propose here a new perceptual image hashing algorithm and a new benchmark for them in the context of document security.

In this paper, we start by formalizing the definition of a stable algorithm in Section I and the metrics to evaluate it. Then we present the challenges related to image authentication and perceptual image hashing and the state of the art in Section II. We continue with the presentation of our proposed algorithm and its evaluation and comparison with two state of the art algorithms on two extensive datasets in Sections III and IV before concluding this paper.

I. Formal definition of the notion of stability

One can consider that a stable algorithm is an algorithm capable of producing similar (respectively dissimilar) outputs given similar (respectively dissimilar) inputs. Notice the absence of any ground truth in this definition. We consider that an algorithm is a specific kind of function.

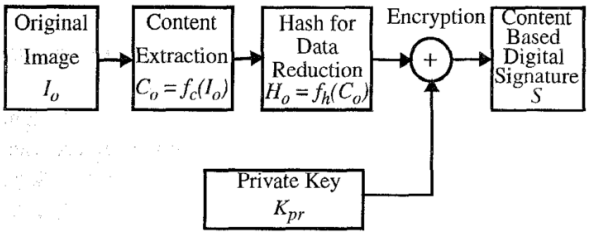

Fig. 1: The process to compute a content based hash. Image reproduced from [3].

Definition I.1 (Stable function). *Let us have*

- *A function f (the algorithm): $f : I \longmapsto O$.*
- *A binary similarity function s_1 for its input space I and a binary similarity function s_2 for its output space O.*

f is stable with respect to s_1 and s_2 if and only if

$$\forall \{a, b\} \in I^2, s_2(f(a), f(b)) = s_1(a, b) \tag{1}$$

In our case, s_1 tests if two documents are similar modulo print and scan noise and s_2 tests if two digests are a match.

Before defining evaluation metrics, let us summarize what we need to verify the definition of a stable function f:

- A similarity function for the input space: s_1
- A similarity function for the output space: s_2
- A set of similar and different inputs

Obviously $s1$ and $s2$ depend on the space in which they are defined but they should be made as independent of f as possible in order to keep a generic definition of stability.

To measure how much stable is a function, we need to measure how much Definition I.1 holds. Since this definition can only take a Boolean value, we will instead measure how frequently it is true. More precisely, given two inputs a and b what is the probability that $s2(f(a), f(b)) = s1(a, b)$?

For this we consider that a positive condition occurs when the inputs are similar and a negative condition occurs when they are not. This should not be confused with many medical or security related conventions where a test is said to be positive when the outcome is not equal/not normal.

The similarity of the algorithm's output can be considered as a prediction. It is a true prediction if the output similarity/positiveness is the same as that of the inputs e.g. if the two sides of Equation (1) are equal, and false otherwise. The

question becomes: what is the probability that the prediction matches the condition e.g. that it is true? Several classical metrics have already been defined to estimate this probability on a given dataset. Among them we choose a set of four metrics that is independent of the dataset bias towards positive or negative conditions:

$$\text{False negative rate (FNR)} = \frac{\sum \text{False negative}}{\sum \text{Condition positive}} \quad (2)$$

$$\text{False positive rate (FPR)} = \frac{\sum \text{False positive}}{\sum \text{Condition negative}} \quad (3)$$

$$\text{False omission rate (FOR)} = \frac{\sum \text{False negative}}{\sum \text{Prediction negative}} \quad (4)$$

$$\text{False discovery rate (FDR)} = \frac{\sum \text{False positive}}{\sum \text{Prediction positive}} \quad (5)$$

They all require the ground-truth to be computed during testing. However, once they are computed for a given algorithm, they can be used in the following cases. FOR and FDR provide information about the veracity of the prediction for a given prediction result (without knowing the ground truth) and thus are widely used in commercial applications. FNR and FPR estimate the veracity of the prediction for a given condition (with knowledge of the ground truth) and are thus used to evaluate an algorithm on a given dataset.

- The false negative rate (FNR) is the probability that an authentic document is wrongly detected as modified.
- The false positive rate (FPR) is the probability that a modified document is wrongly detected as authentic.
- The false omission rate (FOR) is the probability that a document detected as modified is actually authentic.
- The false discovery rate (FDR) is the probability that a document detected as authentic is actually modified.

We will now properly define the problem at hand.

II. Problem statement and state of the art

Some important requirements are expected from a perceptual image hashing function [4], [5], [6], [7]: robustness to accidental changes such as print and scan noise, fragility to modifications, security from attackers, confidentiality of the hashed content and compactness of the digest.

In our case, the goal is to properly identify the legit copies of the same images and the modified copies of these images. A copy is considered modified if it differs significantly from the original image.

It is difficult to pinpoint what is really meaningful in an image and what is not. What level of color modification, of intensity modification is significant? What is the minimal size of a significant modification? The SIGNED project [8] which included some industrial partners suggested that a modification should be detected if it had a size of at least 42 by 42 pixels at a resolution of 600 dpi. For the rest we will resort to making sure that the algorithms perform well on a very challenging dataset. This dataset will be described in Section IV-A. Its main characteristic will be to contain print and scan noise and several versions of the same images.

Performance-wise, FNR, FPR, FOR and FDR should be below 5% for the system to be commercially viable. The digest size should be below or around 500 bytes. This comes from the payload capacity of existing hybrid security embedding technologies with the assumption that the document contains a logo and a handwritten signature.

Finally our problem should not be confused with near duplicate image detection or other perceptual hashing schemes designed to retrieve similar looking images even though they may not be identical. Here we only want to find the exact same images modulo the print and scan noise. This is a difference of similarity function for the input space, s_1.

The different perceptual image hashing techniques can be roughly classified into the following categories based on their approach [4], [7]:

- Coarse representation-based approaches
- Statistical approaches
- Relationship-based approaches
- Sparse feature-based approaches
- Matrix factorization-based approaches

In the state of the art, only Yu [9], Wu [10] and Smoaca [7] seriously tested their algorithms with print and scan noise. Monga et al. [11] also tested it on one image. The most thorough study is that of Smoaca who used 30 images and copied them 31 times making 930 copies. Unfortunately, his algorithm is not versatile enough for our scenario. Yu's method is likely to be too sparse to identify image modifications and thus only Wu's method is really applicable to our scenario. For comparison purposes we will also use Venkatesan's method [12] whose statistical nature and popular tiling approach should yield interesting results. The algorithm of Wu is a relationship-based approach and the one of Venkatesan is a statistical one. The algorithm that we propose uses a coarse representation.

Notably, the previous papers on this topic neglect to study the FDR and FPR. As we will see in Section IV, they are key criteria.

III. A Simple Yet Complex Hashing Algorithm (ASYCHA)

Similarly to most perceptual hashing approaches we plan to compute a compact description of the image and then we use an adequate matching algorithm.

The general idea is that the digest should be an extremely lossy and high ratio compression of the image. Then we use image registration techniques to compare two digests. Similarly to the state of the art, the challenge of our work lies not so much in the techniques used but rather in their careful choice and combination to achieve the required level of performance, hence the name of the algorithm. We differ from classical techniques by working in the spatial domain. This reduces the noise related to the discrete nature of the image space. This noise would be more significant with frequency based approaches especially if the floating point frequency values have to be quantized to fit in one byte like an image pixel. Our registration technique is also more advanced than

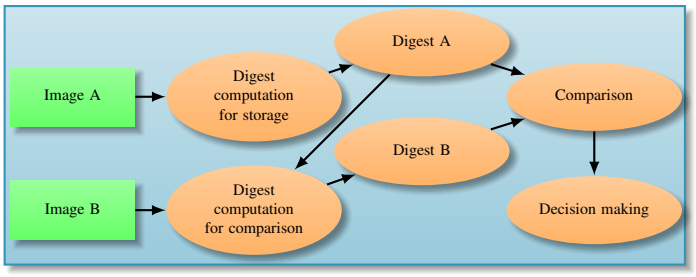

Fig. 2: Overview of the image hashing and authentication process.

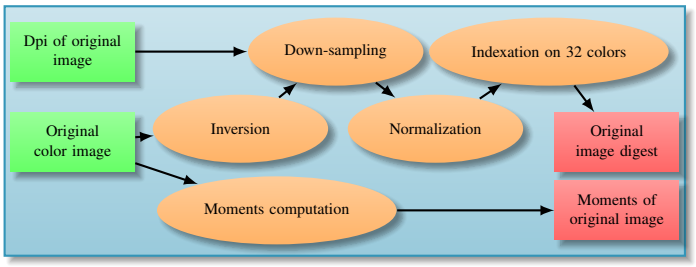

Fig. 3: Algorithm for digest generation.

the one of [13] and allows a better handling of geometric distortions. Finally we devise a color quantization scheme that is deterministic and retains the flexibility of the k-means clustering without adding other parameters than k.

The hashing and matching algorithms are made to work on both logos and handwritten signatures. They are actually content agnostic apart from the fact that we suppose that the background is white. This is not a limitation since replacing the background color by white is fairly easy to do for many types of images (when there is a background). Many background or foreground detection algorithms are available for this.The process of image hashing and authentication is displayed on Figure 2

A. Hashing algorithm

This subsection follows the generation of the digest that is illustrated in Figure 3. The input is the original color (RGB) image, I, accompanied by its resolution information ρ (dpi). The dpi is often stored in the image meta-data produced by the scanner. Its output is a specific indexed image and the second order moments of the input image.

Moments computation: The central second order moments are computed as:

$$\begin{aligned} \forall\{p,q\} \in [\![0;2]\!],\ \ p+q=2, \\ \mu_{pq} = \textstyle\sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q I(x,y) \end{aligned} \tag{6}$$

They are stored with the digest of the image and used in the matching for image registration. Since they are only used to compute the direction of the inertia axes of the image, we do not need the moments of higher order.

Inversion: This step relies on our assumption that the background is white. It is accomplished because we want the correlation of the matching algorithm to operate on the foreground image pixels. To perform this inversion, the RGB image is converted into the YCbCr color space, which is already used for JPEG compression. Thus it reduces conversion colorimetric noise. It separates the intensity information in the Y channel from the color information in the Cb and Cr channels. The inversion function B is defined as: $I_B = 255 - I_Y(x,y)$ where I_Y is the Y channel of image I, and I_B is the inverted image. Then the image is converted back to the RGB color space.

Down-sampling: The down-sampling compresses the image and reduces salt and pepper noise as well as small color variations introduced by the print and scan process. All images are resized to a resolution of ≈ 14.3 dpi. This scale is chosen as it means that we consider that modifications with a size smaller than a square of 1.4 mm are insignificant. This is similar to the objectives of the project SIGNED [8]. The down-sampling uses a bilinear interpolation with a reduction factor of $\frac{600}{42\times\rho}$. Our experiments show that this achieves the best trade-off between a precise image and little artifacts.

Normalization: The normalization compensates for brightness variations introduced by the print and scan process. The normalization stretches the histogram of the Y channel to use all the range of values. This increases the space between the colors and allows a better selection of them in the indexing step. This is not a histogram equalization which would smooth the histogram.

Indexing: The indexing (or quantization) intends to remove the colorimetric noise introduced by the print and scan while retaining the significant color information. This step also helps reducing drastically the size of the digest. The image is indexed on a maximum of 32 colors i.e. on 5 bits. Logos usually contain far less colors. This produces an index matrix H and its color mapping table T.

The indexing is done by a k-means clustering of the image on the RGB cube with the Euclidean distance. For a good initialization of the centroids of the k-means, we use a variance minimization quantization [14] which makes the indexing deterministic. Having a deterministic color quantization algorithm is paramount for the stability. The variance minimization quantization is deterministic but not satisfactory as it is a hierarchical partitioning algorithm which may miss some color clusters. This is why the variance minimization quantization is followed by a k-means clustering which solves this issue.

Finally, the digest of the image contains the moments $\{\mu_{11}, \mu_{02}, \mu_{20}\}$, the resolution of the image ρ, the index matrix H and its color mapping table T.

B. Matching algorithm

The matching algorithm's input is the digest of the original image and the color test image J and I' (test values are denoted with a quote). The matching process has two steps: the generation of the digest of the test image and its comparison with the digest of the original image.

1) Generation of the test image digest: The generation of the test image digest depicted in Figure 4 is similar to the process of generating the digest of the original image (Section III-A) except for the linear image registration. We add an image registration step in order to handle the rotation and

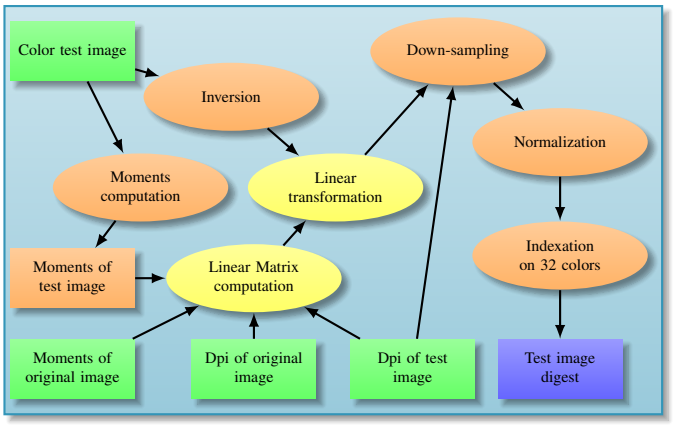

Fig. 4: Generation of the test image digest

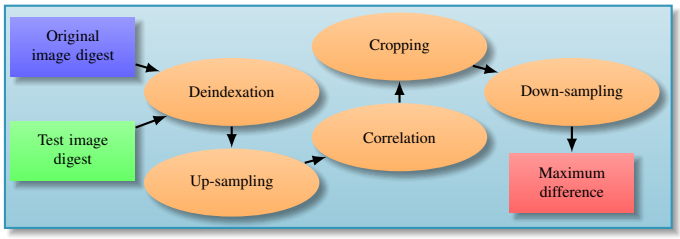

Fig. 5: Algorithm for digest comparison

scale variations which might be introduced by the print and scan process.

The base idea of the linear image registration is to compute the equivalent ellipse orientations and sizes based on the second order moments of the images to be compared. Then, it rotates and resizes the test image so that its corresponding ellipse is the same as that of the original image. The linear transformation matrix T is computed with the second order moments and the resolution information of both images. It is based on a scale factor and a rotation angle. The rotation angle called θ is determined by Equation (7), and the scale factor called Δ by Equation (8). The matrix T is computed as shown in Equation (9).

$$\theta = \arctan(\sqrt{\frac{\mu'_{20}}{\mu'_{02}}}) - \arctan(\sqrt{\frac{\mu_{20}}{\mu_{02}}}) \tag{7}$$

$$\delta' = \sqrt{\frac{(\mu'_{20}+\mu'_{02})+\sqrt{(4*(\mu'_{11})^2)+(\mu'_{20}-\mu'_{02})^2}}{2}} \qquad \Delta = \frac{\rho'}{\rho} \cdot \sqrt{\frac{\delta}{\delta'}} \tag{8}$$
$$\delta = \sqrt{\frac{(\mu_{20}+\mu_{02})+\sqrt{(4*(\mu_{11})^2)+(\mu_{20}-\mu_{02})^2}}{2}}$$

$$T = \begin{bmatrix} \Delta \cdot \cos(\theta) & \Delta \cdot \sin(\theta) \\ -\Delta \cdot \sin(\theta) & \Delta \cdot \cos(\theta) \end{bmatrix} \tag{9}$$

When applying the rotation with the linear transformation G, the unknown pixels are filled with black (background) color. The background is black because the image has been inversed.

The rotation and scaling may introduce some significant artifacts, in particular for rotations of small angles. To deal with this issue is we simply test the image with and without the image registration.

2) Digest comparison: This section follows the comparison process depicted in Figure 5.

Deindexation: The deindexation will convert back the digests J and J' to RGB images for further processing.

Up-sampling: Because of the very small size of the digests, it can happen that the optimal correlation position occurs between two pixels. Thus, each digest is resized by a factor 12. 12 can be divided by 2, 3, 4, 6 and 12 thus allowing a precise image positioning on these fractions of a pixel without dealing with too large images. We use a bicubic interpolation. This may create artifacts but because we use the same factor for both images, we expect that if the images are similar, so will be the artifacts and they should not increase significantly the difference between the images.

Correlation: The correlation of the digests, allows us to get the best overlay between the two digests and to compensate for any translation. We perform one normalized correlation per color channel. Then, the resulting matrices R_i are summed to obtain the coordinates of their maximum, (x_m, y_m):

$$(x_m, y_m) = \underset{x,y}{\operatorname{argmax}}(\sum_i R_i(x,y)) \tag{10}$$

where i indicates the color channel. This maximum defines the translation which gives the optimal overlay of both digests. To perform the correlation for most image sizes, the test digest is padded on each side by half the size of the original digest.

Cropping: The cropping keeps only the overlapping area of the digests after the translation by the correlation. At the end of this step, the digests have the same size.

Down-sampling: The image down-sampling resizes the digests back to their original scale with a factor $1/12$ and a bilinear interpolation.

Maximum difference: The maximum difference υ is the Hausdorff distance between the two digests, e.g. the maximum absolute value of the digest differences along each color channel.

$$\upsilon = \max_i(\text{abs}(K'_i - K_i)). \tag{11}$$

where i is the color channel and K and K' are the original and test outputs of the previous step. In [13] they mention that the Hausdorff distance is sensitive to outliers, but this is not the case here because outliers have been removed during the down-sampling steps.

The above steps form the matching process which provides the distance between both images: υ. It is between 0 and 255 for 8-bit integer images.

C. Decision making

The decision making is done through a decision tree integrated with the comparison process. The goal of this decision tree is not to make an elaborated decision scheme but rather to discard obvious cases as soon as possible in order to save computation power. The critical decision is really only made at the last stage.

If the rotation angle θ (7) is too large or if the scale factor Δ (8) is too far from the identity value 1, then the images are

considered as different. Similarly, if their size does not allow to make the correlation, they are considered as different. From the cropping step we extract another feature, the coverage β, defined by:

$$\beta = \frac{\text{area}(K')}{\max(\text{area}(J'), \text{area}(J))} \quad (12)$$

If this coverage β falls below a threshold, the images are again considered as different. Finally, if the distance between the images υ is too large, the images are considered as different. This is this last decision criteria that significantly impacts the performance of the algorithm.

When the rotation is very small, the linear transformation adds more noise than it removes. This increases the distance between the images to be compared. Hence we compute two matches with and without the image registration and the images are matched if any of them is positive.

This successive case pruning allows the algorithm to eliminate obvious different images without having to perform too much computation. Furthermore, as we evaluate weak constraints first, we safely reduce the potential noise for the last constraint (the distance) which becomes a better classifier.

IV. Evaluation of ASYCHA

In order to evaluate ASYCHA we created a dataset of photocopies of logos and handwritten signatures.

Following our evaluation framework defined in Section I we define the similarity function of the input space as the indicator of whether the two images being compared are photocopies of the same image. The similarity function of the output space is given by the matching algorithm and the decision process. We first present the testing dataset and then the evaluation results.

A. Testing datasets: SignCopies and LogoCopies

We have compiled two challenging datasets to evaluate our algorithm. SignCopies requires that the algorithm should be capable of detecting the differences between two signatures made by the same person while being able to identify photocopies of the same signature as identical.

LogoCopies has similar requirements, but this time the differences to detect can involve color as well as localized modifications in contrast to signatures where slight differences occur on the whole signature.

Both datasets include JPEG compression with quality factors of 75, 82 and 94 as produced by the scanners. The printers have also produced different levels and kinds of noise.

a) Signature dataset (SignCopies): SignCopies is a dataset of photocopied handwritten signatures. The original images are from the training dataset of SigComp2009 [15] containing 1898 images of handwritten signatures. They were printed by three printers: Sharp MX 904, Lexmark x543 PS and Konica Minolta Bizhub 223. Then, they were scanned by four scanners at two different resolutions making $3 \times 6 = 18$ copies of each image (see Table I). This makes a total of $18 \times 1898 = 34164$ signatures.

This dataset contains several signatures made by the same person as well as forged signatures. In our case we will consider that only the photocopies of the same signature are identical. Thus several signatures from the same author are considered different as well as their forged versions. Our algorithm is not made for handwritten signature authentication.

		SignCopies		LogoCopies	
Resolution (dpi)		300	600	300	600
Fujitsu fi 6800		1	0	0	0
Konica Minolta Bizhub	223	1	2	1	2
	C364e	0	1	1	1
Lexmark x543 PS		1	0	1	0

TABLE I: Number of copies for each scanner and each resolution.

Fig. 6: An example of images of different logos of the LogoCopies dataset.

b) Logo dataset (LogoCopies): LogoCopies is a dataset of photocopied logos. The original images taken from the site of the logo library [1] are composed of 200 logos of beer brands. They were scaled at three different sizes: 20 mm, 25 mm, 30 mm and they were printed by three printers: Sharp MX 36N, Lexmark x544 and Ricoh pro c7100x. Then they were scanned by three scanners at two different resolutions as shown in Table I.

Some logos are from the same brand but at different times and thus have only small differences. These logos should be considered different unless the difference is smaller than the spatial resolution of the digest which is a square of 42 by 42 pixels. Figure 6 shows some of the logos.

This makes a total of $18 \times 3 \times 200 = 10800$ logos and a total dataset size (SignCopies and LogoCopies) of 44964 images. They are available at http://shades.univ-lr.fr/datasets/.

B. Results

We compare our results with the method of Venkatesan et al. [12] and that of Wu et al. [10]. We used 250 blocks instead of the original 150 blocks for the method of Venkatesan in order to have roughly the same spatial resolution than that of our method. Keeping a constant number of blocks independently from the image size (in cm) would result in a significant loss of performance for large images. This conclusion was verified by our experiments and we only present the best results obtained with 250 blocks. Wu's method works best as described in the original paper so no adaptation was made.

The thresholds for Venkatesan's and Wu's methods are chosen to optimize both robustness and fragility on the whole

[1] www.lalogotheque.com

Metric	Venkatesan	Wu	ASYCHA
FNR (%)	**0.3**	5.2	8.2
FPR (%)	8.9	39.3	$\mathbf{3.2 \times 10^{-3}}$
FOR (%)	2.7×10^{-2}	3.4×10^{-3}	$\mathbf{3.3 \times 10^{-3}}$
FDR (%)	49.9	99.9	**8.0**
Digest size	500 Bytes	**50 bits**	median 427 Bytes

TABLE II: Best results for the different methods. All the values should be as small as possible.

dataset. They are respectively 0.009 and 0.12.

The optimal decision parameters of ASYCHA are a maximum angle of rotation of $t_\theta = 2°$, a maximum scale difference of $t_\delta = 8\%$ and a minimum coverage of $t_\beta = 85\%$ and a maximum distance of $\upsilon = 83$. They produce the best performance trade-off on the whole dataset.

We can see from Table II that Venkatesan's and Wu's methods are very robust (low FNR and FOR) but not very fragile (or precise) in our context (high FPR and FDR). Our method is far more fragile than the state of art (FPR, FDR) and thus more able to distinguish different and potentially fraudulent images. It also maintains a similar robustness (better FOR than Venkatesan's method, but not on FNR). Globally our method achieves a better trade-off than the state of the art. The stability performance is really given by the maximum error rate for each algorithm. From this point of view our algorithm brings a very significant improvement to the state of the art (8% compared to 50% and 100%).

The digest size for Venkatesan is 500 bytes and that of Wu is 50 bits. ASYCHA does not produce a fixed size digest. This is done in order to maintain a fixed spatial precision for the image. The digest size s for an image of size (m, n) pixels is given by Equation (13) in bits for a color map of 32 colors. The first $2 \times 10 + 10 + 3 \times 16 = 78$ bits are used to store the image size, resolution and moments. Then $3 \times 8 \times 32 = 768$ bits are used for the color map and the rest is the color index (5 bits) for each pixel of the digest.

$$\begin{aligned} s &= 78 + 768 + 5 \times (m \times n) \times \frac{600 \times 600}{(42 \times 42 \times \rho^2)} \\ &= 846 + (m \times n) \times \frac{50\ 000}{(49 \times \rho^2)} \end{aligned} \tag{13}$$

On the test dataset the minimal, median and maximal digest sizes are respectively 186, 427 and 1174 Bytes. The majority of the digest fits within the allotted space and only the digests of handwritten signatures are too large. This is in relation with the sizes (in cm) of the handwritten signatures that can also be quite big. Thus the combination of the digests of a logo and of a handwritten signature should have and adequate mean size.

V. Conclusion

In this paper we have proposed an algorithm, ASYCHA, which has been tested on an extensive dataset and it outperforms the state of the art in terms of stability in particular for the detection of different or fraudulent images. It achieves a reasonable digest size and computation times.

The decision process to compare two images takes four thresholds that have the advantage of relating to physical properties of the image which makes them easier to tune. Thanks to keeping the color map it is also very easy to compare a gray level or binary photocopy of an image with its color digest. If this is the case, the image can still be authenticated with a warning that it has lost its color information.

We acknowledge that the basic blocks involved in this algorithm are simple. However, it is the careful choice of these algorithms and their precise combination that allows us to deal with all the sources of noise, artifacts and instability and to achieve our level of performance. Considering the performance improvement that is almost as desired, this work is a significant step in the right direction which could be improved further.

Acknowledgment

This work is financed by the ANR (French national research agency) project SHADES referenced under ANR-14-CE28-0022.

References

[1] A. Smith, "Identity fraud: a study," Economic and Domestic Secretariat Cabinet Office, Tech. Rep. July, 2002.

[2] S. Eskenazi, P. Gomez-Krämer, and J.-M. Ogier, "When document security brings new challenges to document analysis," in *International Workshop on Computational Forensics (IWCF)*. SPIE, 2015, pp. 104–116.

[3] M. Schneider and S.-F. Chang, "A robust content based digital signature for image authentication," *International Conference on Image Processing (ICIP)*, vol. 3, pp. 227–230, 1996.

[4] Y. Lei, Y. Wang, and J. Huang, "Robust image hash in Radon transform domain for authentication," *Signal Processing: Image Communication*, vol. 26, no. 6, pp. 280–288, 2011.

[5] F. Ahmed, M. Y. Siyal, and V. Uddin Abbas, "A secure and robust hash-based scheme for image authentication," *Signal Processing*, vol. 90, no. 5, pp. 1456–1470, 2010.

[6] G. Zhu, J. Huang, S. Kwong *et al.*, "Fragility analysis of adaptive quantization based image hashing," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 133–147, 2010.

[7] A. Smoaca, "ID photograph hashing: a global approach," Ph.D. dissertation, Université Jean Monnet, 2012.

[8] A. Malvido Garcià, "Secure Imprint Generated for Paper Documents (SIGNED)," Bit Oceans, Tech. Rep. December 2010, 2013.

[9] L. Yu and S. Sun, "Image authentication in print-and-scan scenario," in *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE, 2007, pp. 295–298.

[10] D. Wu, X. Zhou, and X. Niu, "A novel image hash algorithm resistant to print-scan," *Signal Processing*, vol. 89, no. 12, pp. 2415–2424, 2009.

[11] V. Monga and M. K. Mihçak, "Robust and secure image hashing via non-negative matrix factorizations," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 376–390, 2007.

[12] R. Venkatesan, S.-M. Koon, M. Jakubowski *et al.*, "Robust image hashing," in *International Conference on Image Processing (ICIP)*. IEEE, 2000, pp. 664–666.

[13] V. Monga, D. Vats, and B. L. EvanS, "Image authentication under geometric attacks via structure matching," in *International Conference on Multimedia and Expo*. IEEE, 2005, pp. 229–232.

[14] X. Wu, "Efficient statistical computation for optimal color quantization," in *Graphics Gems II*, A. S. Glassner and J. Arvo, Eds. Academic Press Inc., 1991, pp. 126–134.

[15] V. L. Blankers, C. E. Van Den Heuvel, K. Y. Franke *et al.*, "The ICDAR 2009 signature verification competition," in *International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2009, pp. 1403–1407.