# A System Based On Intrinsic Features for Fraudulent Document Detection

Romain Bertrand*†, Petra Gomez-Krämer*, Oriol Ramos Terrades†, Patrick Franco* and Jean-Marc Ogier*
*Laboratory L3i, University of La Rochelle, Avenue Michel Crépeau, 17042 La Rochelle, France
{romain.bertrand, petra.gomez, patrick.franco, jean-marc.ogier}@univ-lr.fr
†Computer Vision Center, Universitat Autònoma de Barcelona, 08193 Bellaterra (Barcelona), Spain
oriolrt@cvc.uab.es

***Abstract*—Paper documents still represent a large amount of information supports used nowadays and may contain critical data. Even though official documents are secured with techniques such as printed patterns or artwork, paper documents suffer from a lack of security. However, the high availability of cheap scanning and printing hardware allows non-experts to easily create fake documents. As the use of a watermarking system added during the document production step is hardly possible, solutions have to be proposed to distinguish a genuine document from a forged one. In this paper, we present an automatic forgery detection method based on document's intrinsic features at character level. This method is based on the one hand on outlier character detection in a discriminant feature space and on the other hand on the detection of strictly similar characters. Therefore, a feature set is computed for all characters. Then, based on a distance between characters of the same class, the character is classified as a genuine one or fake one.**

***Keywords*—paper document, document analysis, fraudulent document, forgery, fake.**

## I. Introduction

Nowadays, paper documents are still largely used as information support, often due to legal reasons. With the improvement of numerical hardware and software resources, we observe a real enthusiasm for digitizing paper documents and to automatic extraction, in order to improve the way humans will process their content. As a backlash, the generalization of this kind of resources makes document forgery techniques commonly available. Thus, automatic fake document detection is needed with respect to the large amount of paper documents still being produced.

Many ways can be used to produce a fake document: a piece of paper can be glue on the genuine document in order to mask or to add information before a copy; reproducing the genuine document in a word processing software while keeping the same shape but with the addition of information. But one of the most common techniques is the use of image processing software such as Gimp or Photoshop. In this paper, we focus on the Scan-Edit and Print (SEP) technique where a genuine paper document is digitized in order to be edited using image processing software. The resulting fake image is then directly sent by email or printed as an original copy of the document. The SEP technique consists either of a copy and paste forgery where a set of characters is copied and pasted in a different area of the document, or of an imitation forgery where the fraudster adds or modifies information of the document by imitating the document's font properties. In both cases, we assume that the fraud is not perfect and marks such as misalignment or skew can be found in the document.

There are two main challenges of forgery detection. First, the digitization of paper documents involves different solutions according to the document type, the number of documents to digitize and the size of the information system that will manage the digital copies. It makes sense to digitize an identity card in color to keep the identity picture as close as possible to the original one because of its significance. In case of an incoming bill or pay slip, a black and white digitized copy is often sufficient to preserve the main information. Hence, in order to reduce costs due to the time consuming color scan process and the large storage capacity required for digitized color documents, it is common to digitize such documents as black and white low resolution images.

Second, we have to deal with different document types (bills, pay slips, proof documents...) from wide sources. Thus, due to the large dissimilarity of those documents, the use of specific document models to detect forgery is pointless. Another way to approach this problem is to use intrinsic document features such as its font properties, character shapes, and character/word alignments for forgery detection.

To this end, we focus in this paper on the detection of binarized low resolution documents grossly frauded (i.e. the fakes are eye-catching). The approach relies on the use of a set of intrinsic features computed from the document at character level. As mentioned above, one of the main idea is to identify characters that are very similar which is the case of a copy and paste forgery, or in the opposite very distant in case of an imitation forgery as shown in Figures **??**a and **??**b respectively. In a first step, a set of intrinsic features is extracted for each character in the document. Those feature sets will analyzed in two ways: The first one is character shape comparison in order to detect similar characters and the second is a distance measure of characters features to detect imperfections generated by the forgery step. Then, a score is associated to each character used in a global scoring method to classify if the character has been forged or not.

This paper is organized as follows. Section **??** presents the state of the art of document forgery detection. Section **??** describes the proposed forgery detection method and Section **??** defines the experimentation used to evaluate our system and discusses our results. Finally, Section **??** gives a conclusion and outlines perspectives and future work.

## II. Related work

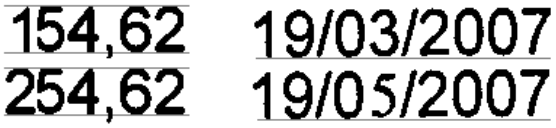
Many works have been presented for the detection of copy and paste forgeries in natural color images. A good overview of these methods can be found in [**?**]. Anyway, such methods can not be used in our case due to the different nature of black and white document images.
A common way to detect fraudulent documents is the use of extrinsic document features in watermarking techniques such as printed patterns or background security patterns [**?**], [**?**]. Nevertheless, such security features are not present in most of the paper documents that we use everyday (i.e. bill or voucher). Thus, the forgery detection is not feasible using watermarking approaches as the document production process can not be controlled.
Due to a low efficiency of watermarking techniques and the impossibility of controlling the production process of everyday life's paper documents, some methods using intrinsic features documents have been presented recently. Printer classification and recognition techniques [**?**], [**?**], [**?**] allow to determine the kind of hardware used in the paper document production step. Another approach is presented in [**?**], where intrinsic features at a connected components level are used to generate a document signature in order to compare the document with a reference model. The assumption is that different parts in a certain document type are fixed and a variation of their position can be due to the fraudster's imperfections caused by adding or deleting information. The authors of [**?**], [**?**] propose to detect document forgery at the document's line level. In [**?**], a statistical model is used to detect lines with a deviant orientation with respect to the other lines. Such distortion occurs when a line (or paragraph) is glued on the genuine document and copied afterwards. In [**?**], line positions are compared relatively to the document alignment lines (left, center and end points of the text-lines) in order to detect misaligned lines or paragraphs. The method [**?**] is based on a similar approach than [**?**], but the use of a specific document model is avoided by matching all the documents from the same source in a cluster in order to obtain a matching quality score.

## III. Proposed method

Here, we present a forgery detection method that uses intrinsic document features. In contrast to [**?**], [**?**], forgery is detected at character level and not at line level. Furthermore on the contrary to [**?**], our aim is to propose a generic method which avoids the use of a dedicated document model or document sources clustering as we have to deal with different unknown document types.

When producing a fake document using a SEP forgery, as far as we know, the fraudster may prefer two kinds of techniques to create a fraudulent document:

- Copy and paste forgery: a set of characters is copied from the document and pasted on another area to replace or add information as shown in Figure **??**a. Thus, the final digitized fake document will contain characters with exactly the same shape. When the characters are copied from another document with another font properties, the fraud belongs to the forgery process below.
- Imitation forgery: the fraudster adds or replaces information by trying to find the same document font properties or the closest ones as show in Figure **??**b. Hence, the final digitized fake document contains words with a different font type, character size, misalignment or skew.




a) Copy and paste technique b) Imitation technique

Fig. 1. Examples of a SEP forgery. In a), The top number is the genuine one. The bottom number is a fake: the first *2* if a copy of the last one past over the *1*. In b), The top date is the genuine one. The bottom data is a fake: the month number is replace by a *5*. Whereas the main font is a sans serif one, the fraudster use a serif *5* number.

Based on these two forgery techniques, different indicators are used for the detection of a fraudulent character. Indeed, on the one hand the detection of too similar characters is a good clue to detect a copy and paste forgery, while on the other hand the detection of outlandish characters (in terms of shape, alignment or skew) is a good clue for imitation forgery. Those indicators are related to the way a human would check the integrity of a document. We describe in the following our method for forged character detection based on character shape comparison (similarity/dissimilarity of characters) and outlier character detection.

### A. Pre-processing step

First of all, characters are extracted using Tesseract OCR [**?**]. Afterwards, a hierarchical data structure is created to represent the document structure. At the top level, line objects represent each text-line of the document. Each line contains a set of word objects, composed of character objects, which are the lowest elements in the data structure. For each element, the paternity relation is conserved in order to create a tree structure to determine, for example, to which word a character belongs to. For each character, informations such as the alphabetical class, the character type (alphabetic character, number or symbol) and the bounding box are memorized.

### B. Detection of similar characters

The detection of copied and pasted or imitated characters is based on character shape comparison. The result of the comparison between two character shapes $a$ and $b$ is a distance $D_{(a,b)}$ obtained by processing a difference between their feature vector $V$ represented by a vector of $x$ elements.

Fraud detection based on a character shape descriptor distance relies on three assumptions:

1) When a distance $D_{a,b}$ between a character $a$ and character $b$ equals 0, then they are exactly the same and are suspect.
2) When $D_{a,b}$ is very low (i.e. close to 0), then probably one of the characters is a copy of the other, but slightly modified by the fraudster's intervention (adding or deleting black pixels/noise).

3) When the distances $D_{a,all}$ obtained by the comparison of a character $a$ with the rest of the characters from the same alphabetic class are really different from the mean distance of this class, then probably the character $a$ is a fake character obtained by an imitation forgery.

The first assumption is relevant for copy and paste forgery detection because of an extremely low probability to find two characters of exactly the same shape after a print and scan process. Indeed, we printed one hundred characters of the same alphabetic character class with the same font, type and size, and then scanned the resulting paper document and finally extracted all the characters in order to compare them by matching the characters pixelwise. We obtained only 0.00001% of perfectly matching characters for 100! comparisons of the characters *three*. Those differences of character shapes of the same class with the same font properties after a print and scan process are due to the subsampling in the print and scan process. They are especially visible at a low resolution. Thus, in case of a forgery detection on a document directly sent by the fraudster in a digital format, characters of identical shape will be considered as a clue for a fraud. The second assumption takes into account noise added by the fraudster to mask his manipulation. When the noise added on purpose is close to the noise generated by the print and scan process, the fraudulent character will be miss-classified. The third assumption concerns copy and paste forgery or an imitation forgery where a character doesn't share the same font properties than the others.

Lots of shape descriptors are available in the literature. Hu invariant moments [**?**] are particularly adapted for character description as shown in [**?**]. The invariance property of the Hu moments to rotation, scale and translation is a way to ensure that two characters will be detected as equivalent even if one of them was subject to fraudster's imperfections. Furthermore, they are easy to implement. Our objective is to characterize characters by a feature vector of Hu moments in order to detect character similarities, but we plan to use more sophisticated feature vectors in case of insufficient discrimination of the Hu moments. The features vectors of the characters of an alphabetical class are compared using the Euclidean distance:

$$D_E(V_a, V_b) = \sqrt{\sum_{i=1}^{7}(V_{a,i} - V_{b,i})^2} \quad (1)$$

where $V_a$ and $V_b$ are respectively the feature vectors containing the seven Hu moments for the characters $a$ and $b$.

Finally, we retain character pairs where $D_E$ satisfies one of the three previous assumptions by the use of a set of threshold definied by empiric observations. At this point, it is not possible to determine for the first two assumption the potential fake character of a pair. Thus, we measure the distortion of character features for all the characters in a second step to detect conception errors with the hope that a character detected as a copy or an imitation of another one is also affected by fraudster's inaccuracies.

### C. Detection of conception errors

In case of copy and paste or a imitation forgery, different inaccuracies can be found at character level. After detecting similar characters, we are now interested in the detection of the conception errors produced during the forgery. The goal is to detect inaccuracies in the document structure such as character misalignment in a line or different character sizes, positions or orientations in the same word.

We propose to define a set of forgery clues, regrouped in a feature vector $W$, computed for each character present in the digitized document:

- Character size: for each character, the pre-processing step provides the bounding box from which the character size can be easily derived. This property is included in the feature vector because of the assumption that a different size for two characters $a$ and $b$ of the same type and which belong to the same word $w$ is suspicious as shown in Figure **??**a.
- Character principal inertia axis: the principal inertia axis is obtained by using the Singular Value Decomposition method (SVD) on the $\eta_{20}$, $\eta_{11}$ and $\eta_{02}$ Hu invariant moments [**?**]. The character orientation is a valuable information to detect forged characters because of the assumption that a fraudster can copy and paste a character from another document, but with a different skew. Likewise he can also insert a character by using a text-box editor in a document that suffers of an orientation problem as shown in Figure **??**b.
- Character horizontal alignment: like the skew issue, the fraudster can misalign a character with respect to a word or a line as shown in Figure **??**c. A pixel distance is thus computed between a character and the horizontal alignment line to measure the potential error gap. The horizontal alignment line is computed by a linear regression with respect to the extracted text-line using [**?**].

a) Size issue b) Skew issue c) Alignment issue

Fig. 2. Example of conception errors: a) the number 6 with a different size, b) the number 4 with a different skew and c) the misaligned number 8.

The feature vector $W$ of each character is compared to a data model $M_c$ for a specific character alphabetic class using the Mahalanobis distance $D_M$. The Mahalanobis distance, based on correlations of variables, is appropriate for our case as it integrate the variability of the data distribution and as it can be seen as a dissimilarity measure. It is computed as follows:

$$D_M(W_i, M_c) = \sqrt{(W_i - M_c)^t \Sigma^{-1} (W_i - M_c)} \quad (2)$$

where $W_i$ is the feature vector corresponding to the character $i$ and $M_c$ is mean feature vector of class $c$ corresponding to the alphabetic class of the character $i$. $(W_i - M_c)$ denotes the vector $W_i$ according to the eigen values of the data model covariance matrix and $\Sigma$ is the inverse of the data model covariance matrix in a non-degenerate vectorial subspace.

The data model $M_c$ is constructed for each alphabetical character class. It is obtained from a training set in which 10% of the characters have been modified. By using a font property estimation, the data model can be generated without any a priori knowledge on the analyzed document.

Finally, a threshold is applied to the distance of each character: characters with a low distance are classified as a genuine character whereas the characters with a high distance are classified as an outlier i.e. a potential fraudulent character.

### D. Scoring system to detect fraudulent document

At this point, we got two different indicators to detect fake documents: characters that share a too similar shape or at the opposite, too dissimilar shapes and characters that break the continuity of the document features. Each time a character is detected as belonging to one of these two groups, the information is stored in the data structure. A weight is added at each potential fraudulent character depending on the situation. For example, when the character was detected as fraudulent by both methods and the word to which it belongs contains other potential fraudulent characters then the weight will be higher. On the contrary, when the character is detected by only one method, the weight will be lower.

## IV. Experiments and results

In this section, we describe the data we used and three experiments we realized for the evaluation of the proposed method.

### A. Dataset

As far as we know, there is neither a public dataset of fraudulent document available, nor a dataset of administrative files with character alignment errors or skew. Thus, to overcome the ground truth issue, we developed a software to create synthetic fraudulent document images.

A synthetic fraudulent document is defined by its content ($x$ characters distributed in $y$ words), its font properties and a set of distortions at character level corresponding to the main fraud imperfections and/or a paste and copy forgery as explained previously. Thus, simulating a forgery consists in moving a character along the y-axis, changing its shape size or rotating it by few degrees. Another operation consists in copying a character and pasting it over another one. Characters impacted by a forgery are randomly chosen in the document and can be affected by several falsifications at the same time. To simulate noise according to a print and scan process, Kanungo noise as described in [**?**] was applied.

### B. Experiments

We realized three experiments in order to evaluate our method.

*1) Shapes similarities/dissimilarities:* the first experiment concerns the detection of the fraudulent characters using the shape comparison: It refers to the method of Section **??**. We distinguish three cases:

1) The fraudster scans a document, frauds by copying and pasting a set of characters and by emailing it.
2) The fraudster scans a document, frauds by copying and pasting a set of characters and by adding some noise to mask his manipulations and emails it (or sends it per mail).
3) The fraudster scans a document, frauds by copying and pasting a set of characters that belong to another document with different font properties and emails it.

We have generated three distinct datasets containing 20,000 numeral characters in Liberation Serif 11pt corresponding to the three cases enumerated above. For each one, Kanungo noise was added to the characters (parameters $\alpha = \beta = 2.0$) before the copy and paste forgeries. In the second case, a second pass to apply noise is performed after the forgery step but with low $\alpha$ and $\beta$ parameters (randomly between 0.1 to 0.2) to simulate noise added by the fraudster. In the third case, the copied and pasted characters belong to another file where the font type is Liberation Sans 11pt.
For each fraud, 5% of the numerical characters are obtained by copying and pasting others characters. All the documents were created at low resolution (300 dpi). Results are presented in Table **??**.

*2) Outlier detection - imperfection retrieval:* the second experiment is related to the detection of the imperfection due to the manipulation of the image by the fraudster. It refers to the method described in section **??**. Actually, a fraudster can make only one imperfection when he copies and pastes a character or he can accumulate them. To evaluate the capability of the method to detect the fake characters in such cases, we generated eight datasets corresponding to the combination of the possible imperfections: alignment error, character size error, skew error, alignment error + character size error, alignment error + skew error, character size error + skew error and finally, the combination of these three errors.

Each dataset contains 20,000 numerical characters where 5% are fakes. The alignment error is randomly from 1 to 3 pixels (up or down), the skew error is randomly created by modifying the inertia axis of a character by 4 to 8 degrees (clockwise or anticlockwise) and the size error is simulated by increasing or decreasing randomly the size of a character by 5 to 10%. The documents were created with the Liberation Serif font at 11pt in a 300 dpi resolution. Results are shown in Table **??**.

*3) Fraudulent document detection:* the last experiment consists of a combination of the two previous ones: copied and pasted characters that are also affected by one or more of the three common imperfections. It refers to the evaluation of the scoring system defined in **??**. Nevertheless, the fraudulent characters belong to the same word (e.g. supposing that a complete date or money amount was falsified). One dataset containing 20,000 numerical characters was created with 5% of copied and pasted characters suffering of imperfections. The imperfections are the same that in the second experiment. The Liberation Serif 11pt font was used and the document was generated in 300 dpi. Results are presented in Table **??**.

### C. Results

The results of the three experiments described above are discussed in this section. The shape similarities and dissimilarities experiment results (Table **??**) reveals that a comparison

| | Recall | Precision |
|---|---|---|
| Copy/paste - noise free | 1.0 | 0.5 |
| Copy/paste - noise | 0.22 | 0.28 |
| Imitation | 0.35 | 0.70 |

Fig. 3. Shape similarities and dissimilarities experiment results

between a simple character shape descriptor can be sufficient to detect copy and paste forgeries without added noise. The precision of 0.5 in the first case is due to the method itself as we are not able to distinguish in a pair of characters which one is the fraudulent one. When noise is added to mask the fraud, the recall of this method decreases rapidly. When noise close to a scan and print process is added to a document where two character shapes were initially perfectly similar, they become quite different. In the third case that is imitation forgery, the precision increases due to the ability of the method to distinguish a unique character.

| | Recall | Precision |
|---|---|---|
| Alignment error | 0.26 | 0.98 |
| Size error | 0.90 | 0.80 |
| Skew error | 0.41 | 0.92 |
| Alignment + skew error | 0.27 | 0.90 |
| Alignment + size error | 0.90 | 0.89 |
| Size + skew error | 0.54 | 0.36 |
| Alignment + skew + size error | 0.59 | 0.49 |

Fig. 4. Outliers detection and errors retrieval experiment results

In the case of the imperfection retrieval results (Table **??**), the precision of the method seems to be good enough for the character outlier detection except when the imperfections involved a character skew error. For some cases, the recall is quite low. This is due to the threshold $t$ fixed at the same value for all character classes. The experiment shows that by measuring the mean and the standard deviation of all the distances obtained in a class the threshold $t$ should differ according to the alphabetic class of the characters. Thus, a adaptive the threshold $t$ with respect to the character classes could be a way to detect more true positive fakes while the number of false positive characters discovered would decrease.

| | Recall | Precision |
|---|---|---|
| Forgeries detection | 0.77 | 0.82 |

Fig. 5. Fraudulent document detection experiment results

Finally, the last experimentation results (Table **??**) show that the scoring system allows to increase the precision of the copy and paste detection method by distinguishing the fraudulent characters in a pairs of potential frauded characters. It also reveals the importance of the spatial information to detect the fraudulent parts of a document.

## V. Conclusion

We proposed and evaluated a method to automatically detect forgeries at character level in binarized low resolution documents. The method is based on a comparison of character shapes and the detection of structural irregularities of the document. We obtained encouraging results: copy/paste characters could be retrieved by measuring distance between their features vector (recall = 1.0 without added noise) and detecting fraudster imprecision allow to indicate parts of the document potentially modified (especially when the character size is involve). These result could be improved by adding further forgery indicators such as the measure of the gap between two neighboring characters. Moreover, results can be improves by working on a noise model estimation to reduce the print and scan noise issue and by using more robust shape descriptors as the Zernike moments.

## Acknowledgment

This work has been partially support by Spanish project TIN2012-37475-C02-02.