

Text-line examination for document forgery detection

Joost van Beusekom · Faisal Shafait ·
Thomas M. Breuel

Received: 15 July 2010 / Revised: 20 July 2011 / Accepted: 28 December 2011
© Springer-Verlag 2012

Abstract In this paper, an approach for forgery detection using text-line information is presented. In questioned document examination, text-line rotation and alignment can be important clues for detecting tampered documents. Measuring and detecting such mis-rotations and mis-alignments are a cumbersome task. Therefore, an automated approach for verification of documents based on these two text-line features is proposed in this paper. An in-depth evaluation of the proposed methods shows its usefulness in the context of document security with an area under the ROC curve (AUC) score of $AUC=0.89$. The automatic nature of the approach allows the presented methods to be used in high-volume environments.

Keywords Document security · Text-line alignment · Text-line orientation

1 Introduction and related work

In every day life, document verification is an important task as many documents present a potential value. A typical example is bank notes. When handling bank notes, most people quickly control their genuineness by verifying easy to detect

security signs as, e.g., holograms, structured print, and special inks. This is seamlessly done by many people as they have got used to the bank notes and their security features.

A person handling an unseen bank note will not know exactly what features to look for. This person might either just trust the source or look for features it might know from previously encountered bank notes: it might, e.g., look for a watermark or a metallic stripe inside the paper. Thus, even unknown bank notes can be checked for signs of forgery.

For less secured documents, a verification is only feasible if the characteristic features of the document are known. Assume, e.g., the following scenario: a perpetrator has signed a contract that he wants to modify in order to gain advantages over the other contracting party. He therefore adds an additional clause in some whitespace area of the document. This will have as effect that the security features as, e.g., the genuine signatures and the genuine paper are genuine. However, this type of forgery can be detected by detecting verifying the consistency of the text-lines: small variations in the rotation and the alignment of the added text-lines compared to the previously printed, genuine text-lines can often be seen.

This forging approach is only one among others. Unfortunately, no public statistical data are available that shows how forgers generate their documents, especially for this scenario. However, text-line features are used in questioned document examination [1]. The authors also conducted a user study to find out what forgery methods are being used by people who want to defraud money from the insurance company. This user study showed that three different approaches are used:

- **Print, Paste and Copy (PPC) forgeries:** out of 25 forgeries, three were generated by replacing a part of the invoice. This was done by printing the new text (presenting higher values or more items) on an empty sheet and pasting this part onto the genuine document. This was then copied

J. van Beusekom · T. M. Breuel
Department of Computer Science, University of Kaiserslautern,
Image Understanding and Pattern Recognition Group,
67663 Kaiserslautern, Germany
e-mail: tmb@informatik.uni-kl.de

J. van Beusekom (✉) · F. Shafait
German Research Center for Artificial Intelligence (DFKI),
67663 Kaiserslautern, Germany
e-mail: Joost.van_Beusekom@dfki.de

F. Shafait
e-mail: Faisal.Shafait@dfki.de

using a color copier. This technique was mostly used by people without a computer science background.

- **Reverse Engineered Imitations (REI) Forgeries:** three candidates generated an editable document by imitating the genuine invoice. Five samples were delivered. They scanned the genuine invoice and used it as a template to generate a new document by retyping all the text, putting the logos into place, etc.
- **Scan, Edit and Print (SEP) Forgeries:** the remaining 17 forgeries were generated by digitizing the invoice and manipulating the digital image. Here, mostly only numbers were modified, e.g. increasing the price of a tire or increasing the number of tires from two to four.

Details about this user study can be found in Appendix 6. The PPC forgeries represent the class of forgeries that the proposed method can be applied too. In this work, we consider single column documents without tables. These characteristics can often be found in contracts. Also, only the manipulation of entire text-lines is considered. Tampering single characters or digits is most likely to be done by other means, as e.g., done for the SEP forgeries class.

To the authors' best knowledge, there is no related work trying to automate text-line alignment and orientation measurement for forgery detection. Different approaches from optical document security exist. However, these cannot be used in the presented use case: as normal documents do not contain any extra security features, so called intrinsic features have to be used. Intrinsic features are features that are integrated into the printout by the normal document generation process, in contrast to extrinsic features that are added solely for the task of securing a document. This excludes the use of most of the features used in the optical document security domain, as e.g., watermarks, holographic images [2], specialized printing techniques [3], and other physical and chemical signatures [4]. Many other types of extrinsic features can be found in literature [5–7].

However, this is not the first attempt to use intrinsic document features for document security applications. Several uses of intrinsic features have been presented in previous publications: printer identification, the process of assigning a printout to a unique printer or a printer type, has been intensively studied by different groups. Mikkilineni et al. [8–10] present a set of gray scale features that is used to determine the type of laser printer that was used to print a document. Schreyer et al. [11–13] worked on detecting the printing technique used to print a document. Also, classification between printed and copied documents has been analyzed. Using discrete cosine transform features, good performance could be shown even when scanning with relatively moderate resolutions of 400 dpi.

The remaining sections of the paper are organized as follows: the automatic method for verification of printed

documents using text-line skew and alignment measures is presented. The features as well as the statistical models to perform the plausibility check are presented in Sect. 2. Evaluation and results are presented in Sects. 3 and 4. The paper concludes with Sect. 5.

2 Text-line features for document security

For checking the plausibility of a document, the following two features are used: the skew angle of text-lines and the alignment of text-lines. Other features exist that could be used for detecting anomalies in text-lines as, e.g., the text-line and word spacing. Measurements on a set of pages from the same document have however shown that the variation of these features in genuine documents is too high, and that their expected discriminative power will be very poor.

Measurement of the text-line's skew angle is straightforward using the method presented by Breuel [14]. The reason for using this text-line extraction algorithm lies first of all in the fact that it has proven good results in the task of page segmentation, as shown by Shafait et al. [15]. The second main advantage is that an open source implementation of the algorithm is available in the OCRopus¹ OCR system. For the sake of completeness, a short overview of the text-line extraction method is given in Sect. 2.1.

To measure the alignment of a text-line, the following approach is used: first, the left and right *alignment lines* are computed (Sect. 2.3.1). These lines are defined as the left and right margin lines where justified, left- and right-aligned text start or end on. A visualization of the alignment lines is depicted in Fig. 1.

After having extracted the alignment lines, the distance between the start and end point of a text-line to the respective alignment line is computed. These two distances are used as features to perform a plausibility check.

In Sect. 2.2, the use of the skew angles of text-lines for the proposed application is explained. Section 2.3 explains the details of the method using the alignment of text-lines for plausibility checks of documents. In Sect. 2.4, both features are combined into one framework.

2.1 Text-line extraction

Breuel proposed a parameterized model for a text-line with parameters (r, α, d) , where r is the distance of the baseline from the origin, α is the angle of the baseline from the horizontal axis, and d is the distance of the line of descenders from the baseline. The text-line extraction consists in finding parameter triples that maximize a quality function that is highest when all reference points are positioned exactly on

¹ <http://www.ocropus.com>.



Fig. 1 Example document with the *left* and the *right* alignment lines

the line. This is done using a branch-and-bound search. In the first step, the parameter space P is defined:

$$P = [r_{\min}, r_{\max}] \times [\alpha_{\min}, \alpha_{\max}] \times [d_{\min}, d_{\max}] \quad (1)$$

The branch-and-bound algorithm then works as follows:

- 0: initialize the search space and insert it into priority queue Q .
- 1: stop if Q is empty, else get the top search space S from Q .
- 2: if S is a solution: save S and discard all image points contributing to the line from further consideration; Stop if enough results are found. Otherwise, continue with Step 1.
- 3: split S into two subspaces S_1 and S_2 .
- 4: compute upper bounds for the quality of S_1 and S_2 .
- 5: put S_1 and S_2 on Q . Continue with Step 1.

The computation of the upper bound for the quality of the text-line defined by the parameter subspace is done using interval arithmetic on the reference points. The set of reference points $\{x_1, x_2, \dots, x_n\}$ is obtained by taking the middle of the bottom line of the bounding boxes of the connected components in a document image.

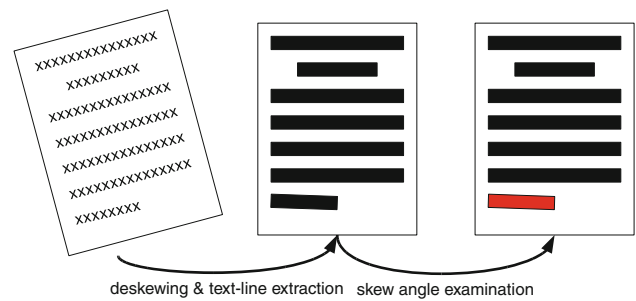


Fig. 2 Visualization of the text-line skew examination: the binarized document is deskewed. The text-lines are examined if their skew angles are abnormally high or not

A priority queue is used to keep track of the subspaces that need to be analyzed. The number of lines to be extracted is used in Step 6 to define the stopping criterion. A space is considered as a solution if it is small enough to identify the unique line in the image. The parameters of the solution represent the detected line.

This algorithm guarantees an optimal solution and is thus a good candidate for the proposed task. The necessary accuracy of the estimated rotation angle can be obtained by changing the stopping criterion to only accept solution regions that are very narrow for the angle dimension. The skew angle of the text-line is then defined as the center of the angle interval of the solution region.

2.2 Plausibility check using skew angles

Using the skew angle of text-lines for identifying suspicious documents is a well-known technique in questioned document examination [1]. Questioned document examiners do this step mostly manually using standard image manipulation software. In this section, the first approach to automate this process is presented.

The main idea of the process is sketched in Fig. 2. The binarized document is correctly oriented and deskewed using the method presented in [16]. Next, the text-line skew angles are extracted. These are checked if they are within the “natural” variation of text-line skew. Then, the text-line is considered as valid. Else, the line is reported as an “implausible” line.

In the next section, the statistical modeling of the skew angle variations and its application to the plausibility check are explained.

2.2.1 Text-line skew variation model

In order to define what “normal” and “abnormal” skew angles are, the natural variation of text-line skew angles has to be measured. Although in the electronic representation, all text-lines are exactly parallel, the transfer to the paper medium



Fig. 3 Examples showing typographic enhancements. Note that the characters “r,” “n,” “f,” and “x” are located slightly above the horizontal *red line*, which connects the *bottom points* of the characters with a *round part* at the *base-line* level, like “b” and “o”

and again back to an electronic image format adds text-line skew variations that have to be considered before being able to make a decision on the skew angle:

- **printing:** variations introduced by printing can be speckles, noise and distortions.
- **digitization:** digitization of a paper-based document leads to many variations coming from the scanning [17] or from subsequent image processing, as e.g. the regularly observed page stretching while using flatbed scanner.
- **typographic enhancements:** optical correction is often used in typography to improve the visual appearance of fonts on lines. This leads to characters starting slightly below the base-line, which will reduce the accuracy of the measurement. An example is shown in Fig. 3.

All these effects can lead to text-line skew variations that are frequently observed and that should thus not be considered as “abnormal”.

In Fig. 4, a histogram of measured skew angles is shown. It can be seen that the distribution is peaked around 0° , that its variation is quite low, and that its shape should be reasonably well approximated by a Gaussian distribution with parameters $\mathcal{N}(\mu_\alpha, \sigma_\alpha)$ being the mean and the standard deviation of the skew angle α .

Using this model, a simple threshold-based method to decide whether a text-line skew angle is suspicious or not could be applied: using the confidence intervals, the threshold could be set to $\pm 3 \times \sigma_\alpha$, so every skew angle outside the 99.7% confidence interval will be reported.

This approach has two drawbacks:

- **no prior:** due to the missing prior, forged text-lines are considered to be as frequent as genuine ones. It is, however, more likely that genuine text-lines are much more frequent than forged ones.
- **text-line length:** for longer text-lines, the skew angle can be measured much more accurately than for shorter ones. Also, the skew angles of shorter lines are more sensitive to noise. In an extreme case, for a line with a length of 100 px consisting of two connected components, one sup-

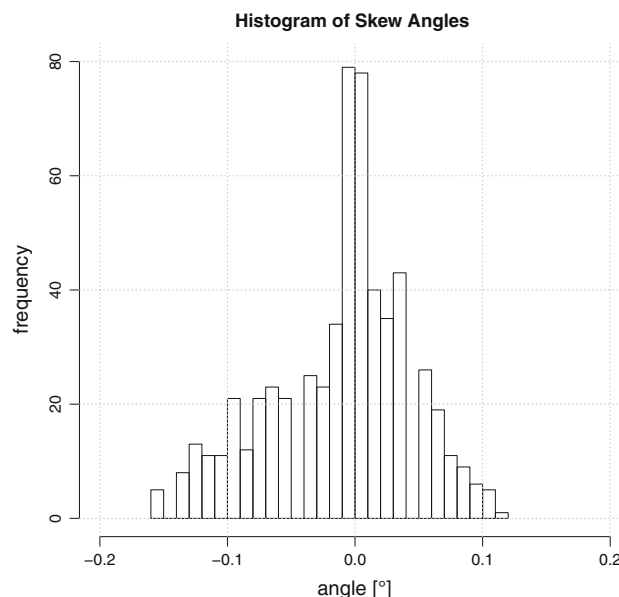


Fig. 4 Histogram of skew angles measured on deskewed pages. The vast majority of measured skew angles is close to 0° , although, small variations can be observed

plemental pixel at the bottom of one component would change the angle by at least $\pm 0.57^\circ$. For longer lines with more connected components the effect of the noise pixel will be smaller.

To solve the first problem, a Bayesian formulation of the problem is proposed:

$$P(f|\alpha) = \frac{p(\alpha|f) \times P(f)}{p(\alpha)} \quad (2)$$

where $P(f|\alpha)$ is the posterior of having a forged text-line given the observation of its skew angle α , $p(\alpha|f)$ is the likelihood of observing the skew angle α knowing it is a forged text-line. $P(f)$ is the prior for observing a forged text-line, and $p(\alpha)$ is the probability of observing the skew angle α (for forged as well as for original text-lines).

In the ideal case, statistics on the parameters could be used to extract reliable estimates. This, however, is only possible for genuine documents, as in a practical setup, no training

data for forged documents are available. Even worse, to the authors best knowledge, there is no public dataset with forged documents that could be used to extract the necessary information. For original documents, the information can partially be computed. Thus, the posterior of having a genuine text-line given the observed skew angle α is written as:

$$P(\neg f|\alpha) = \frac{p(\alpha|\neg f) \times P(\neg f)}{p(\alpha)} \quad (3)$$

In this case, $p(\alpha|\neg f)$ is estimated from training data consisting of genuine documents. As mentioned above, it is modeled as a Gaussian distribution $\mathcal{N}(\mu_\alpha, \sigma_\alpha)$. $P(\neg f)$ is considered as a sensitivity parameter of the system that can be tuned by the operator according to his needs. Still, $p(\alpha)$ is not known due to the missing data from forged documents. The information of interest is which of both posteriors are higher and thus, the normalizing factor can be ignored. Consequently, a text-line is classified as a forged one if:

$$\hat{P}(f|\alpha) > \hat{P}(\neg f|\alpha) \quad (4)$$

where $\hat{P}(\neg f|\alpha) = p(\alpha|\neg f) \times P(\neg f)$ and $\hat{P}(f|\alpha) = p(\alpha|f) \times P(f)$.

As no measurements can be taken on the skew angles of forged documents, an assumption of the likelihood term has to be made: the assumption is that the observed skew angles for forged text-lines can be well modeled using a uniform distribution in a symmetric interval around the mean of 0° . As a forging person tries to obtain perfect rotation angle, expecting the mean to be 0° is a reasonable choice. High rotation angles that can easily be detected with the bare eye are not to be expected, as in this case, the forging person will most likely try to generate a better forgery. Therefore, a uniform distribution is more representative of rotation angles for forged text-lines than a normal distribution.

To solve the second problem concerning the measurement accuracy for lines of different length, the following solution is adopted: instead of estimating the parameters μ_α and σ_α over all the text-lines, the parameters are estimated for a certain text-line length interval. Ideally, the interval would have the size of one pixel. This would require very large amounts of data to obtain reasonable estimates of the parameters. Instead, a window of size $2 \times W$ is defined for which the parameters for a given line length l are computed around in the interval $[l - W, l + W]$. The size of the interval has to be fixed by the operator. In principal, it is best to have small-sized intervals. But this is only possible if enough training data are available. If only few samples are available, a higher value has to be chosen in order to have robust model parameter estimates.

Finally, the two terms that need to be compared are:

$$P(\neg f|\alpha) = \frac{\mathcal{N}(\alpha, \mu_{\alpha,l}, \sigma_{\alpha,l}) \times P(\neg f)}{p(\alpha)} \quad (5)$$

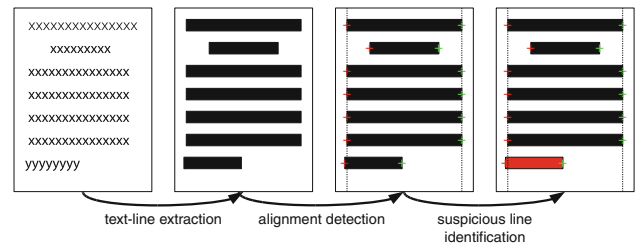


Fig. 5 Visualization of the text-line alignment examination: the text-lines are extracted from the binarized document image. Then, the *left* and *right* alignment lines are computed. Finally, each text-line is examined whether it shows normal alignment or not

and

$$P(f|\alpha) = \frac{\mathcal{U}(\alpha, -w_{\alpha,l}, +w_{\alpha,l}) \times P(f)}{p(\alpha)} \quad (6)$$

where l is the text-line length, $\mu_{\alpha,l}$, $\sigma_{\alpha,l}$ the estimated mean angle and standard deviation for the text-lines with length $[l - W, l + W]$, and where $-w_{\alpha,l}$, $w_{\alpha,l}$ are the left and right bound of the interval uniform distribution for the forged skew angles. Ignoring the normalizing factor, a text-line is classified as forged if

$$\mathcal{U}(\alpha, -w_{\alpha,l}, +w_{\alpha,l}) \times P(f) > \mathcal{N}(\alpha, \mu_{\alpha,l}, \sigma_{\alpha,l}) \times P(\neg f) \quad (7)$$

or as genuine in the other case.

2.3 Plausibility check using text-line alignment

The examination of the alignment property of text-lines has also been previously used in questioned document examination [1]. However, prior to this work, there is no method to evaluate this feature automatically. The main idea of the automated process is sketched in Fig. 5.

In the first step, the text-lines are extracted from the binarized document image. Then, the left and right alignment lines are computed. These are finally used to examine whether the text-lines are normally aligned or not. In the next section, the alignment detection is explained.

2.3.1 Alignment line computation

After the text-line extraction, the alignment lines have to be detected. Four different alignment types are commonly distinguished in typesetting:

- **left aligned:** text-lines start at the left margin
- **right aligned:** text-lines end at the right margin
- **justified:** text-lines start at the left margin and end at the right margin
- **centered:** text-lines do neither touch the right nor the left margin. The gaps between both margins are equal in size.

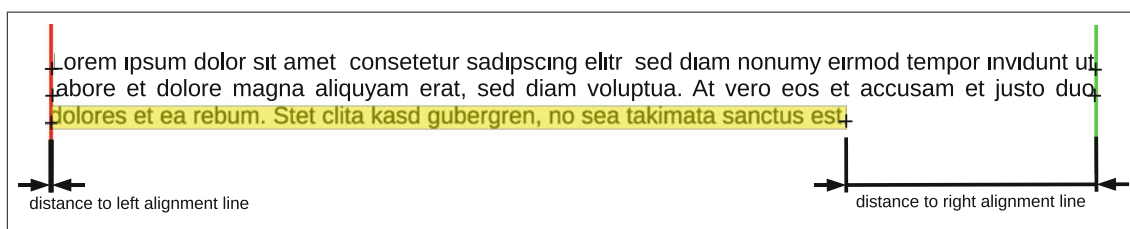


Fig. 6 Visualization of the text-line alignment measure: the crosses represent the start and end points of each line. The vertical *red* and *green* lines represent the *left* and *right* alignment lines. The distances are visualized for the last text-line only

In practical document security applications, centered lines are unlikely to be forged, as they are not frequent and they do usually not contain any valuable information. Therefore, the focus in this work lies on the left and right alignment of the text-lines.

In order to find the alignment lines, the start and the end points of the text-lines have to be analyzed. The left alignment line is defined as a vertical line where left-aligned and justified text-lines have their starting point. The right alignment line is analogously defined as the vertical line where right-aligned and justified text-lines have their end point.

Finding these lines is done using RAST line finding [18]. The polar representation $\vartheta = (|\vec{n}|, \beta)$ is adopted, where $|\vec{n}|$ is the L2-norm of the vector that is normal to the line and that points to the origin, and β is the rotation angle of \vec{n} .

Consider feature points $\{x_1, x_2, \dots, x_n\}$: the quality function that is being optimized to return the best alignment lines is written as:

$$\hat{\vartheta} := \arg \max_{\vartheta} Q_{x_1^n}(\vartheta) \tag{8}$$

where

$$Q_{x_1^n}(\vartheta) = Q_{x_1^n}(|\vec{n}|, \beta) = \sum_{i=1}^n (q_{(|\vec{n}|, \beta)}(x_i)) \tag{9}$$

and

$$q_{(|\vec{n}|, \beta)}(x) = \max\left(0, 1 - \frac{d_{(|\vec{n}|, \beta)}^2(x)}{\epsilon^2}\right) \tag{10}$$

and where $d_{(|\vec{n}|, \beta)}(x)$ is the Euclidean distance of a point x to the line defined by parameters $(|\vec{n}|, \beta)$.

The line finding is run twice, each time with different feature points: once with the starting points and once with the end points of text-lines. The left and right alignment lines are considered as the respective resulting highest quality lines, thus the lines where most start or end points of text-lines lie on.

2.3.2 Text-line alignment model

The alignment feature of a text-line is measured by the distance of the text-line's start and end point to the corresponding

alignment line. A visualization can be found in Fig. 6. For each text-line, two different distances are computed. To simplify the explanation, in the following, only the left alignment distance is considered. The modeling and computation of the right alignment distance are done analogously.

Again, just as for the text-line skew angle variance modeling, there is the problem of missing training data for forged documents. Also, the modeling of genuine documents features is not as straightforward as in case of the skew angle-based method: for the skew angle, it is clear that the angles to be encountered are close to 0° . For the alignment distances, this is not true. Indented text-lines, e.g., will lead to measurements that are significantly off from the normal variations for left-aligned text-lines and these should not be considered as potentially forged lines.

Therefore, instead of assuming a Gaussian distribution as done for the skew angles, the distribution of the alignment distances is measured from a training set of genuine text-lines. A Bayesian classification approach is chosen:

$$P(\neg f|d_l) = \frac{p(d_l|\neg f) \times P(\neg f)}{p(d_l)} \tag{11}$$

where $p(d_l|\neg f)$ is the likelihood of observing the distance d_l of the left alignment line to the left end of the text-line d_l given the information that the text-line is original. As discussed above, assuming a Gaussian distribution would be a bad choice. Therefore, the distribution is measured on a training set.

For the forged case this becomes:

$$P(f|d_l) = \frac{p(d_l|f) \times P(f)}{p(d_l)} \tag{12}$$

The problem to be solved is how to model the likelihood $p(d_l|f)$. Again, training data are not available. The assumption is that the variability around the correct alignment distance is higher for forged lines than for original ones. However, this distribution is not Gaussian, as, just as for genuine text-lines, forged text-lines may start or end somewhere further away from the corresponding alignment line.

The distances are therefore modeled as a mixture of a Gaussian and a uniform distribution:

$$P(d_l|f) = P(j) \times p(d_l|f, j) + (1 - P(j)) \times p(d_l|f, \neg j) \quad (13)$$

where $P(j)$ is the prior of observing a justified text-line, $p(d_l|f, j)$ the likelihood of observing distance d_l given the fact that the text-line is forged and justified, and $p(d_l|f, \neg j)$ the likelihood of observing distance d_l given the fact that the text-line is forged and not justified.

As argued before, $p(d_l|f, j)$ is modeled as being normally distributed $\mathcal{N}(\mu_{d_l}, \epsilon_{d_l})$. For the observed distance in a case of a non-justified text-line, a uniform distribution is assumed $\mathcal{U}(d, 0, L_{\max})$ where L_{\max} is the longest expected text-line. The prior $P(j)$ is estimated from the training set as it is expected to be the same for forged and genuine text-lines.

So far, the modeling of one of the two alignment distances has been explained. Next, these two measures have to be combined into one common framework. Two different measures are generated for each text-line: d_l and d_r , representing the distances of the left and right text-line point to the left and right alignment line respectively. In the end, one decision is needed. In order to combine these three features into one decision, the presented model is extended:

$$P(\neg f|d_l, d_r) = \frac{p(d_l, d_r|\neg f) \times P(\neg f)}{p(d_l, d_r)} \quad (14)$$

and

$$P(f|d_l, d_r) = \frac{p(d_l, d_r|f) \times P(f)}{p(d_l, d_r)} \quad (15)$$

where d_l and d_r are the distances of the left and right text-line point to the left and right alignment line, respectively, $p(d_l, d_r|\neg f)$ is the likelihood of observing d_l, d_r given that the text-line is not forged and $p(d_l, d_r|f)$ is likelihood of observing the aforementioned distances knowing that the text-line is forged.

For classification, the normalization term is discarded. Assuming independence of d_l and d_r , the two terms are rewritten as:

$$\hat{P}(\neg f|d_l, d_r) = p(d_l|\neg f) \times p(d_r|\neg f) \times P(\neg f) \quad (16)$$

and

$$\hat{P}(f|d_l, d_r) = p(d_l|f) \times p(d_r|f) \times P(f) \quad (17)$$

where now $p(d_l|\neg f)$ and $p(d_r|\neg f)$ are measured from a training set and $p(d_l|f)$ and $p(d_r|f)$ are approximated by the mixture distribution described above. The prior is used as a sensitivity parameter to tune the system.

2.4 Integration of skew and alignment

In the previous two sections, the skew angle and the alignment features were considered separately for detecting implausible text-lines. In this section, the combination of both features into a combined framework is presented.

Two different combination schemes have been tested: the first scheme is a simple voting based on the two separate classification results: a text-line is classified as forged if one of the two classifiers presented in the previous two sections “fires.” This is a restrictive approach that should prevent that many forged text-lines are missed.

The second combination scheme extends the statistical model presented in the previous section to include the text-line skew angle α and the alignment distances d_l and d_r .

$$P(\neg f|\alpha, d_l, d_r) = \frac{p(\alpha, d_l, d_r|\neg f) \times P(\neg f)}{p(\alpha, d_l, d_r)} \quad (18)$$

and

$$P(f|\alpha, d_l, d_r) = \frac{p(\alpha, d_l, d_r|f) \times P(f)}{p(\alpha, d_l, d_r)} \quad (19)$$

The normalization factor is discarded and independence of the observations is assumed, leading to the following simplification:

$$\hat{P}(\neg f|\alpha, d_l, d_r) = p(\alpha|\neg f) \times p(d_l|\neg f) \times p(d_r|\neg f) \times P(\neg f) \quad (20)$$

and

$$\hat{P}(f|\alpha, d_l, d_r) = p(\alpha|f) \times p(d_l|f) \times p(d_r|f) \times P(f) \quad (21)$$

Thus, a text-line is classified as forged if $\hat{P}(f|\alpha, d_l, d_r) > \hat{P}(\neg f|\alpha, d_l, d_r)$ and vice versa.

3 Evaluation

Evaluation of the proposed approaches showed to be a challenging task. On the one hand, no public real-world dataset could be found to do a meaningful evaluation on. On the other hand, apart from the observations made during the forgery experiment in Sect. 6, no statistics could be found on the methods used by amateur document forgers. For these reasons, new datasets had to be generated.

- the **Two-pass Print 300 dpi (TP300)** dataset contains 43 document images that were generated using a two pass printing process: in the first pass, an electronic document page was printed. Then, an electronic document was generated having extra text located in the whitespace at the end of the printed document page. This extra text was printed in a second pass on the paper. Finally, this paper was scanned using a resolution of 300 dpi. Images 001 to 015 were used as a training set for the method development. The remaining images were used as test set. Three different laser printers were used in different combinations, assuring that the second pass was always printed on the same printer as the first pass. This test set focusses

on the scenario of a forger who wants to add additional text-lines to an original by printing them onto the original page.

- the **Print, Paste and Copy 300 dpi (PPC300)** dataset contains 49 document images that were generated using a manual modification process: in the first pass, an electronic document page was printed. Then, on a new sheet, a supplementary text-line was printed. This was cut out manually and pasted over an existing text-line on the printed page. A copy of this page was made. The copy has been scanned using a resolution of 300 dpi.
- the **Two-pass Print LaserJet (TPLJ)** dataset contains 64 document images that were generated using a similar two pass printing procedure as for the TP300 dataset. The first difference is that instead of deliberately adding extra text at the end of an integral document page, the page was printed in two passes, the first one printing all but the last part (e.g. leaving out the last paragraph) and the second pass exclusively prints the last paragraph. Again, the same printer was used for the two passes (a HP Laserjet 2100tn). The focus of this test set is again the two print process, but this time with much more constraint setup in order to find out what degree of exactness is possible under ideal circumstances. The only source of distortions that will be present are coming from the interaction with the printer hardware.
- the **Two-pass Print Color LaserJet (TPCLJ)** dataset contains 50 document images that were generated in exactly the same way as the TPLJ dataset, except that a different printer was used (a HP Color Laserjet 4650dn). The goal of this dataset is the same as for PTHL, but this time with a different printer.
- the **Originals** dataset contains 30 document images that were generated by printing and scanning pages from an electronic document. This dataset is used to learn the distributions of the features for genuine text-lines.
- the **Distorted Text-Lines (DTL)** consists of three main parts:
 - Training set consisting of 196 synthetic document images with 49 different fonts in 4 different font sizes (8, 10, 12, 14) *pt*.
 - Rotation test set containing 9792 synthetic document images in 49 different fonts, 4 different font sizes and the last paragraph of varying length rotated between 0° and 1.0° in steps of 0.02° .
 - Alignment test set containing 3920 synthetic document images in 49 different fonts, 4 different font sizes and the last paragraph of varying length shifted horizontally between -0.5 and 0.5 cm in steps of 0.05 cm.

This dataset is used to analyze the effect of varying fonts, font sizes, rotation angles, and line lengths on the pro-

posed method. The dataset was generated automatically using python to generate scalable vector graphic (SVG) files that were then converted into PDF files using the open source software *inkscape*. Ground truth, consisting of a map showing the area containing the distorted text-lines was also automatically generated in the same process. The dataset has been made public and can be downloaded².

All images have been binarized and deskewed before further processing. The deskewing method from [16] has been modified to use the median angle instead of the angle of the text-line with highest quality, to avoid deskewing the page by an angle of a forged text-line. It is reasonable to assume that for a given page, the number of forged text-lines is lower than the number of genuine text-lines. In the first part (Sect. 4.1), it will be shown that the preprocessing of the document (scanning, binarization, and deskewing) is stable and that it is thus likely that the distortions present in the measurements have been introduced by the forging procedure. The hypothesis about the distributions of the features for original text-lines are also verified. Therefore, the text-line alignment distances and the skew angles have been extracted automatically from the *Originals* dataset.

The second part (Sect. 4.2) consists of evaluating the skew angle of text-lines for the plausibility check. The tests were run on all four test sets. For the TP300 and the PPC300 dataset, an extensive evaluation on the influence of both parameters $\sigma_{d_\alpha} \in [0.5, 8.0]$ and $P(f) \in [0.0, 1.0]$ has been done. The method has also been evaluated on the TPLJ and TPCLJ dataset.

The third test setup (Sect. 4.3) repeats the tests from the second setup for the alignment feature. The parameters of interest are $\sigma_d \in [1.0, 30.0]$ and $P(f) \in [0.0, 1.0]$. In both tests, the prior was sampled in with higher density (step size 0.005) in the start ($[0.0, 0.1]$) and end ($[0.9, 1.0]$) regions of the interval and with a step size of 0.02 for the middle part of the interval ($]0.1, 0.9[$).

The fourth part (Sect. 4.4) deals with the evaluation of the combination of both features into a combined framework by using reasonable parameter settings found in the isolated case. Both methods of classifier combination have been tested. For the statistical approach, the parameters $\sigma_\alpha = 2.5$ and $\sigma_d = 20.0$ were fixed according to the settings giving good performance from the previous experiments and sole the parameter $P(f)$ was varied. Tests were done on all four test datasets.

For the evaluation of the alignment distances, centered text-lines are being ignored as they do not fit the proposed model. In the case of the combination of both approaches, if a centered text-line is encountered, only the skew angle is considered to make a decision. Each line that does not start or end within a fixed threshold of six pixels from an alignment

² <http://madm.dfki.de/downloads>.

line and that has its middle point on the center alignment line is being considered as a centered text-line.

As evaluation measures, the receiver operating characteristic (ROC) [19] curves and the area under the ROC curve (AUC) are used.

The last evaluation focusses on the influence of different fonts, font sizes, and line lengths on the forgery detection performance. For these tests, the DTL dataset was generated. As a large amount of data is needed for these tests, synthetic data were chosen. Also, the parameter grid was coarser to keep the computational effort reasonable. For the prior, values {0.0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0} were tested. Values for sigma were set up {1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0} in the case of the rotation measurement and {1.0, 2.0, 3.0, 5.0, 10.0, 20.0, 30.0} in the case of the alignment measure. The results of these tests are given in Sect. 4.5.

Finally, the error analysis and discussion are presented in Sect. 4.6.

4 Results

In the following section, the results of the different evaluation steps are presented. Hypothesis validation is presented in Sect. 4.1. Evaluation of the different features and the combination of both features are given in Sects. 4.2, 4.3, and 4.4.

4.1 Hypothesis validation

In this part, the hypothesis concerning the distributions of the skew and alignment text-line features are analyzed. The features were extracted on the *Originals* dataset containing 30 deskewed and genuine document images. The histogram of the extracted text-line skew angles is shown in Fig. 7. It can be seen that the vast majority of the extracted skew angles lies around 0°.

The histograms of the left and right alignment distances are shown in Fig. 8. As expected, the majority of the text-lines start at the left and end at the right alignment line. Also, peaks coming from indented text-lines are observed. It becomes clear that a frequency measurement-based statistics for computing the likelihood $p(d_{lr}|\neg f)$ is a reasonable choice.

A last test was done to test whether the variations that are being measured do not come from the scanning or the preprocessing steps. Therefore, a small sample set of 21 document pages has been printed once. These pages have been repeatedly scanned, in total 15 times on the same scanner. For each run, the preprocessing pipeline has been run to obtain deskewed images. Also, the mean and standard deviation of the text-line skew angles and alignment distances have been measured. The variance of these parameters over all runs has

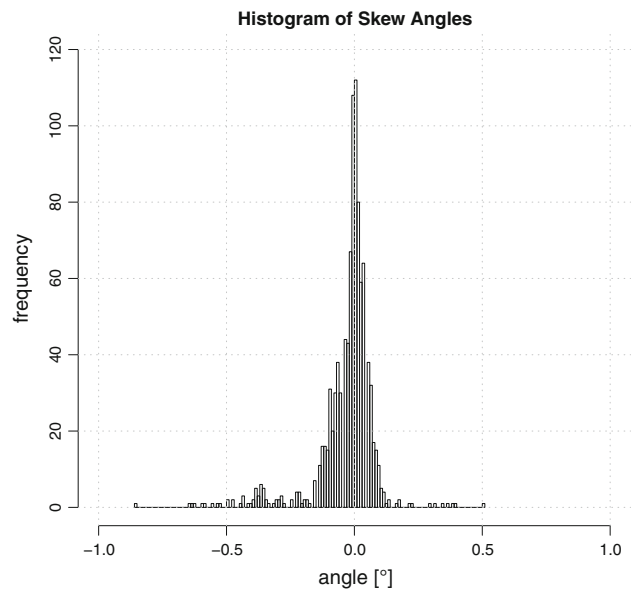


Fig. 7 Histogram of skew angles measured on deskewed pages. The vast majority of measured skew angles is close to 0°. The text-lines were extracted from 30 genuine document pages

been measured. Results can be found in Table 1. It can be seen that the variation between the different runs is quite low. It can thus be concluded that the preprocessing steps are stable and that the variations measured in the test datasets are likely to be induced by document generation process rather than by the digitization process.

4.2 Results on text-line skew angles

Results on the TP300 and the PPC300 dataset are presented in Fig. 9. It can be seen that in both cases, the best results in terms of area under the ROC curve (AUC) is found for the parameter $\sigma_\alpha = 2.5$. Also, small values of σ_α give bad performance. This is due to the high number of false positives (original text-line detected as a forged one), as the threshold for identifying a skew angle as suspicious is really low. Vice versa, high values of $\sigma_\alpha > 3$ lead to a high number of false negatives (forged text-lines detected as original ones). It can also be seen that for these high values, there are practically no points in the center and end part of the ROC curve, leading to the conclusion that for these parameters settings, the prior influences the outcome only for extreme values.

It can also be seen that the performance on the two-pass printing forgeries is slightly lower compared to the manual forgeries. This goes along with the observation that the skew angle variations on the manually forged text-lines are much higher than for the two-pass printed text-lines.

The results on the TPLJ and TPCLJ dataset are shown in Fig. 10. The overall performance is comparable to the performance obtained for the TP300 dataset. Also, the

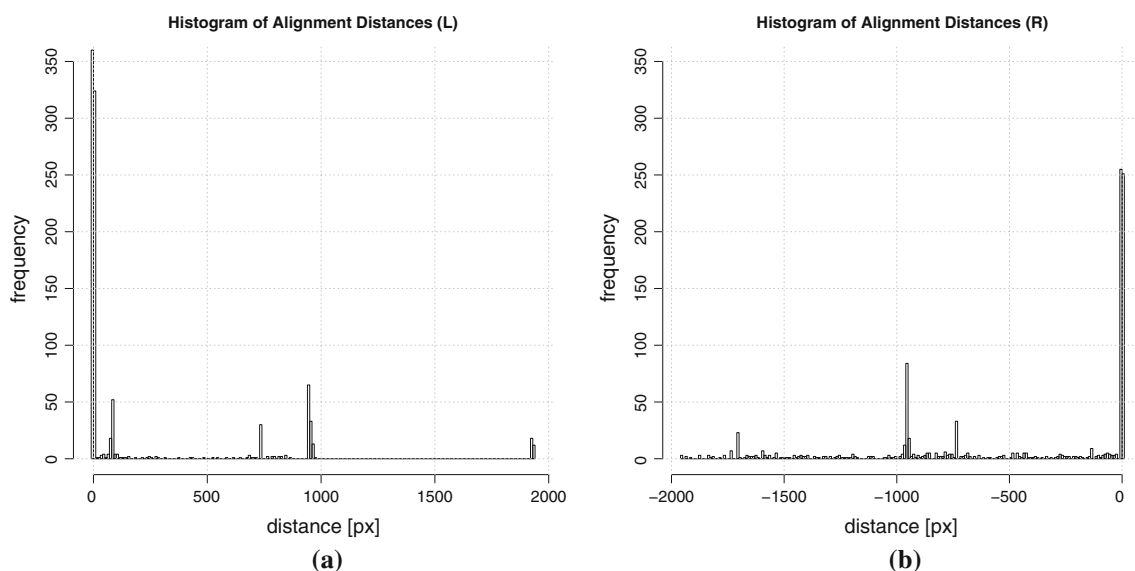


Fig. 8 Histogram of the distances of the *left* and *right* line points to the corresponding alignment lines. As expected for aligned, the majority of lines start on the *left* and end on the *right* alignment line. Also,

peaks can be identified that are likely to belong to indented lines. **a** Left alignment distances. **b** Right alignment distances

Table 1 Measurements of the features for different runs of the pre-processing pipeline consisting of scanning, binarization and deskewing

	μ_α	σ_α	μ_{d_l}	σ_{d_l}	μ_{d_r}	σ_{d_r}
Run-01	-0.004	0.112	96.7	363.3	-435.1	577.0
Run-02	-0.006	0.106	92.3	363.5	-434.8	576.9
Run-03	-0.010	0.117	96.2	362.1	-434.8	576.9
Run-04	-0.004	0.107	96.3	363.6	-435.1	577.3
Run-05	-0.010	0.115	96.5	363.5	-434.8	577.1
Run-06	-0.002	0.104	94.0	362.8	-434.3	577.5
Run-07	-0.012	0.110	92.0	364.5	-435.1	577.0
Run-08	-0.010	0.109	95.3	363.6	-434.7	576.6
Run-09	-0.007	0.116	95.2	362.5	-434.0	577.2
Run-10	-0.005	0.108	95.9	363.4	-435.5	576.7
Run-11	-0.004	0.106	92.2	363.5	-434.4	576.5
Run-12	-0.010	0.111	96.2	363.2	-435.0	576.7
Run-13	-0.012	0.123	95.5	363.5	-434.8	576.7
Run-14	-0.015	0.112	96.5	363.4	-434.8	576.8
Run-15	-0.011	0.105	90.8	364.5	-434.9	576.9
SD	0.004	0.005	1.99	0.62	0.36	0.27

The low standard deviations show that the preprocessing step does not introduce any significant variations of the measurements

characteristic gap of graph points in the middle of the curve is observed.

4.3 Results on text-line alignment

The results on the TP300 and the PPC300 dataset can be found in Fig. 11. The figures show different ROC curves for

different values of σ_d and a varying prior $P(f) \in [0.0, 1.0]$. For reasonable values of σ_d , the performance is good. For values of $\sigma_d > 10$, the performance does not improve significantly anymore.

In contrast to the previous results on the skew angle-based approach, the results on the TP300 dataset are better than for the PPC300 dataset. This is likely due to the increased amount of control over the alignment in the manual forgery process, than in the case of the two-pass print forgeries, where the alignment is only partially controllable by adjusting the document page in the paper tray accordingly.

The results on the TPLJ and TPCLJ dataset can be found in Fig. 12. Again, the performance is better than on the manual forgeries, although slight differences between the two printers can be observed.

4.4 Results on the combination of both features

The results for the combination using the voting scheme that classifies a text-line as forged if at least one of the two separate classifiers does so can be found in Fig. 13a. It can be seen that in three of the four cases, the combination outperforms the best of the single classifiers on the respective dataset, using the AUC as performance measure.

For the test on the combination using the statistical model, the parameters σ_α and σ_d were fixed with values that proved to perform well in the previous tests: $\sigma_\alpha = 2.5$ and $\sigma_d = 20.0$. The results can be found in Fig. 13b.

In comparison with the results of the single features, using the AUC as a measure, the combination of the two features

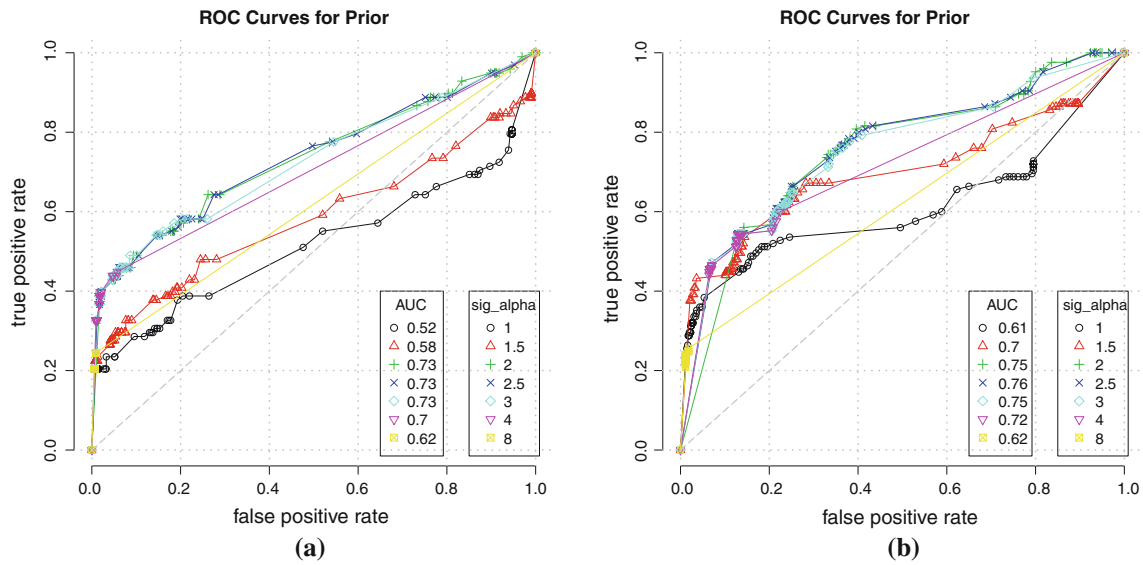


Fig. 9 Results for the test on the skew angle of text-lines on the TP300 and PPC300 datasets. The performance is best for $\sigma_\alpha = 2.5$ **a** TP300, **b** PPC300

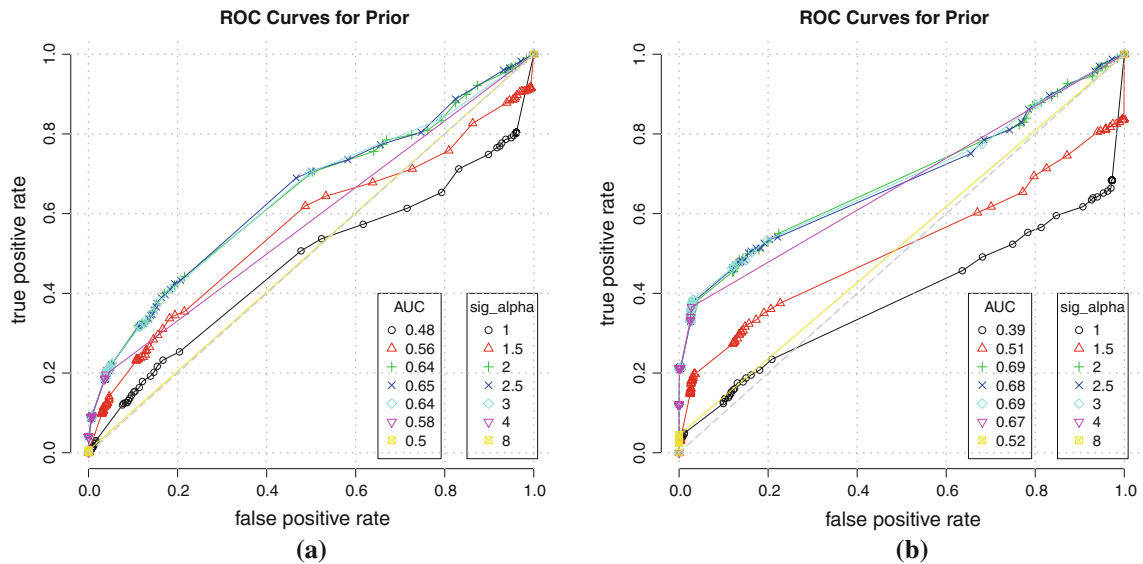


Fig. 10 Results for the test on the skew angle of text-lines on the TPLJ and TPCLJ datasets. It can be seen that the performance is best for $\sigma_\alpha = 3.0$ **a** TPLJ, **b** TPCLJ

does not significantly change the performance. One can also see that for some datasets, the combination slightly outperforms the best single feature (PPC300, TPCLJ) but for the other two cases, using only the alignment would have given a slightly better result in terms of AUC.

4.5 Results on the DTL dataset

The results for the orientation test on the DLT dataset can be found in Fig. 14. The figures show boxplots of the accuracy for different fonts (Fig. 14a), font sizes (Fig. 14b), rotation angles (Fig. 14c), and line lengths (Fig. 14d) for all differ-

ent parameter sets. The distribution of the line lengths in the dataset is shown in Fig. 15d.

It can be seen that the accuracy neither depends on the font nor on the font size. Furthermore, smaller angles of deviation lead to lower recognition rates as can be seen in Fig. 14c. Also, shorter text-lines tend to have lower accuracy, which is most likely due to the fact that short text-lines cannot be measured as accurately as long ones.

The results of the test for the text-line alignment measurement can be found in Fig. 15. The boxplot (Fig. 15a) of the accuracy compared to the font (for all different parameter pairs from the parameter sets) shows that for most fonts, the

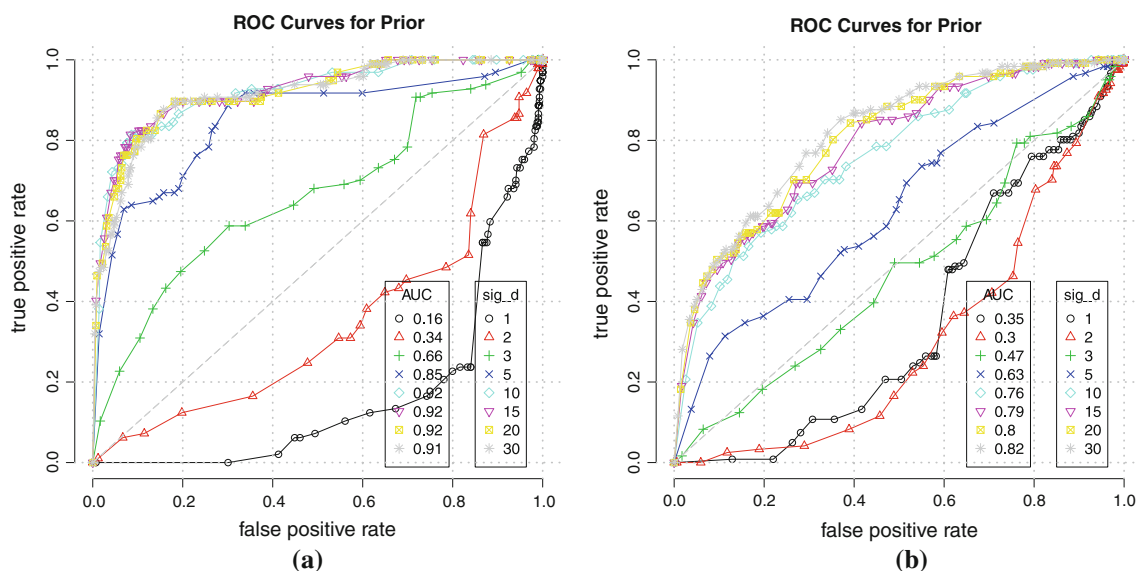


Fig. 11 Results for the test on the alignment feature. Low values of σ_d lead to a bad performance. Values of 15.0 or higher work well for the analyzed dataset **a** TP300, **b** PPC300

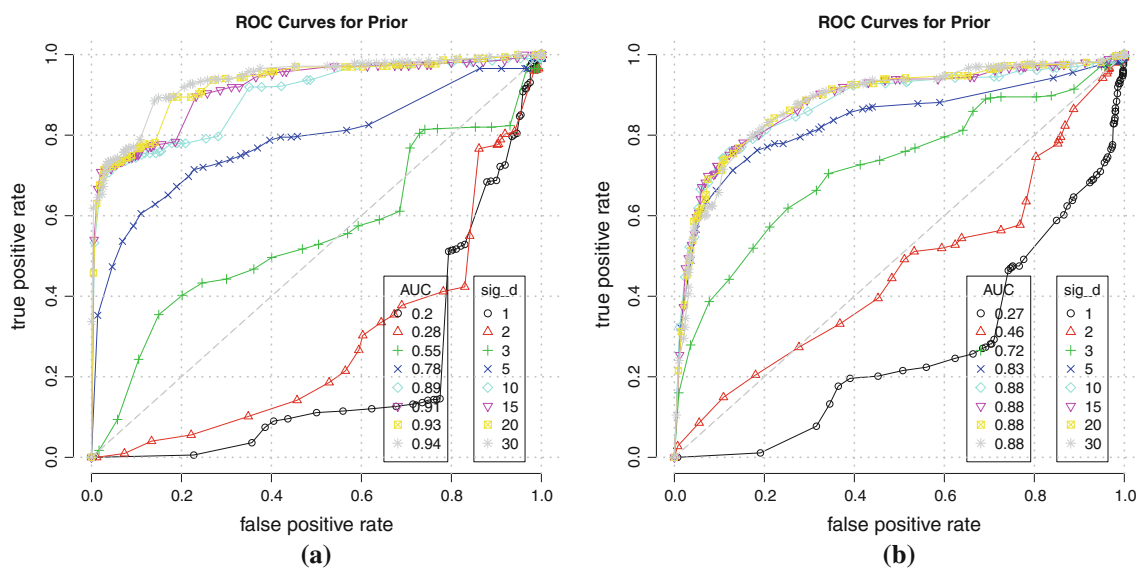


Fig. 12 Results for the test on the skew alignment of text-lines on the TPLJ and TPCLJ datasets. It can be seen that a good performance is obtained for values of $\sigma_d \geq 15.0$ approximately **a** TPLJ, **b** TPCLJ

performance of the method is similar. It can thus be concluded that the method works well on many widely used fonts.

Concerning the performance in relation to the mis-alignment, it can be seen in Fig. 15c that for small mis-alignments, the accuracy is higher. This is due to the higher true positive rate, whereas true negative rate stays relatively constant. This result coincides with our model definition that assumes that high mis-alignments are less likely to come from the forging process than small mis-alignments.

Finally, in Fig. 15b, it can be noted that the accuracy drops with increasing font size. In this case, this is due

to the decreasing true negative rate: one possible explanation could be that with increasing font sizes, the natural mis-alignments due to optical corrections increase and thus many unforged lines are classified as potentially forged ones.

4.6 Error analysis and discussion

An analysis of the errors made by the proposed text-line skew examination gave the following results:

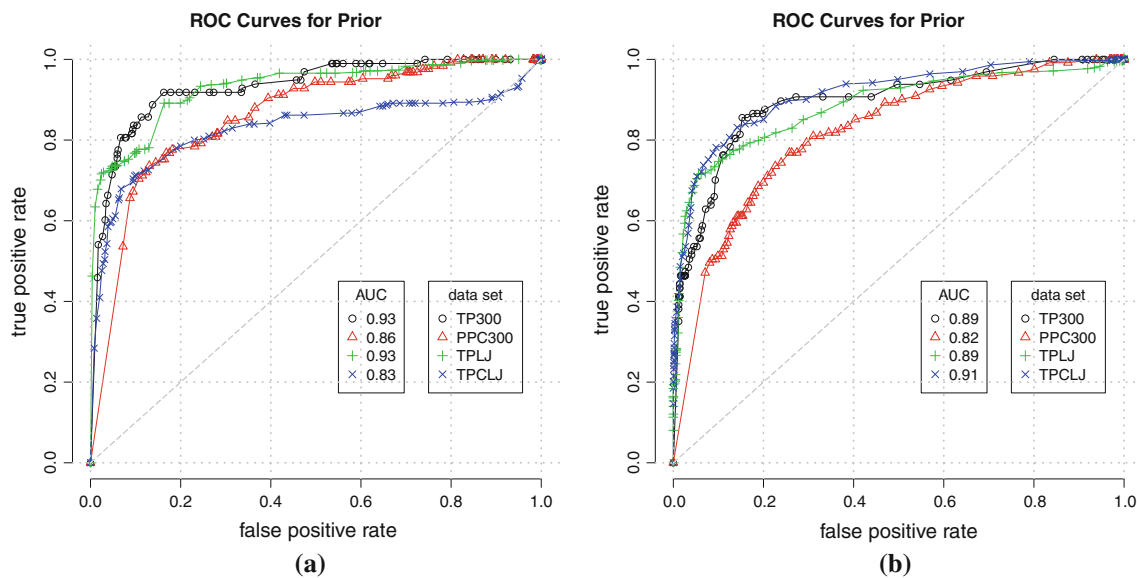


Fig. 13 Results for the test on the combination of the alignment and the skew angle feature using the voting scheme (a) and using the statistical model (b)

- **line splits:** for the manually generated forgeries of the PPC300 dataset, it may happen, that the forged text-line is split into two text-lines by the text-line finding algorithm, due to the narrow constraints set on the positional error of the points lying on the text-line. It is observed, that pasting pieces of paper containing a single text-line on an existing document is a task that is likely to introduce not only skew of the text-line but also a certain degree of curl, leading to the split text-line during extraction. An example image illustrating the problem can be seen in Fig. 16a.
- **errors in deskewing:** in rare cases, when more forged text-lines are present than original ones, the deskewing will rotate the page in a way such that the majority of the text-lines are horizontally aligned. In this case it may happen, that the forged text-lines are considered as original ones and vice versa. An illustration is shown in Fig. 16b.
- **limited discriminative power of the feature:** in several cases, there is no significant difference in skew angle of forged and original text-lines. In these cases, the method basing on skew angle measurement alone cannot detect the forgery. An example is given in Fig. 16c.

The first two error types reduce the accuracy of the proposed method in the current evaluation setup, but in a real-world scenario, these problems would be less important, as in both cases suspicious lines would be reported to the human operator who can easily see that there are issues related to that document page. For the last problem, no solution can be found relying on the skew angle alone. Other features have to be added to detect these forgeries.

An inspection of the errors made by the system in the examination of the alignment feature showed the following reasons for failure:

- **typographic enhancement:** problems, as e.g. italics at the start of the line or different characters that reach outside the line boundaries, as e.g. an “f” at the end of the line: for optical correction reasons characters may stick outside of the normal text-line borders, leading to false positives as can be seen in Fig. 17a.
- **wrong alignment line:** when more indented text-lines are present than regular ones or if the number of forged text-lines is higher than the number of original text-lines, the wrong alignment lines may be found leading to misclassification of genuine lines. This rare error leads to both false positive and false negative classifications. An example is shown in Fig. 17b.
- **errors in text-line finding:** in rare cases text-lines extend to border noise or are split into two parts.
- **indented text-lines:** text-lines from enumerations are slightly indented, leading to occasional false positives, as can be seen in Fig. 17d
- **limited discriminative power of the feature:** in some cases the left and right alignment of the forged text-lines is so accurate that a detection using the alignment feature is not feasible. This problem leads to false negative classifications. An illustrative example is given in Fig. 17c.

The problem with the optical corrections and with enumerations could be solved by integrating optical character recognition information to the process. Text-lines starting

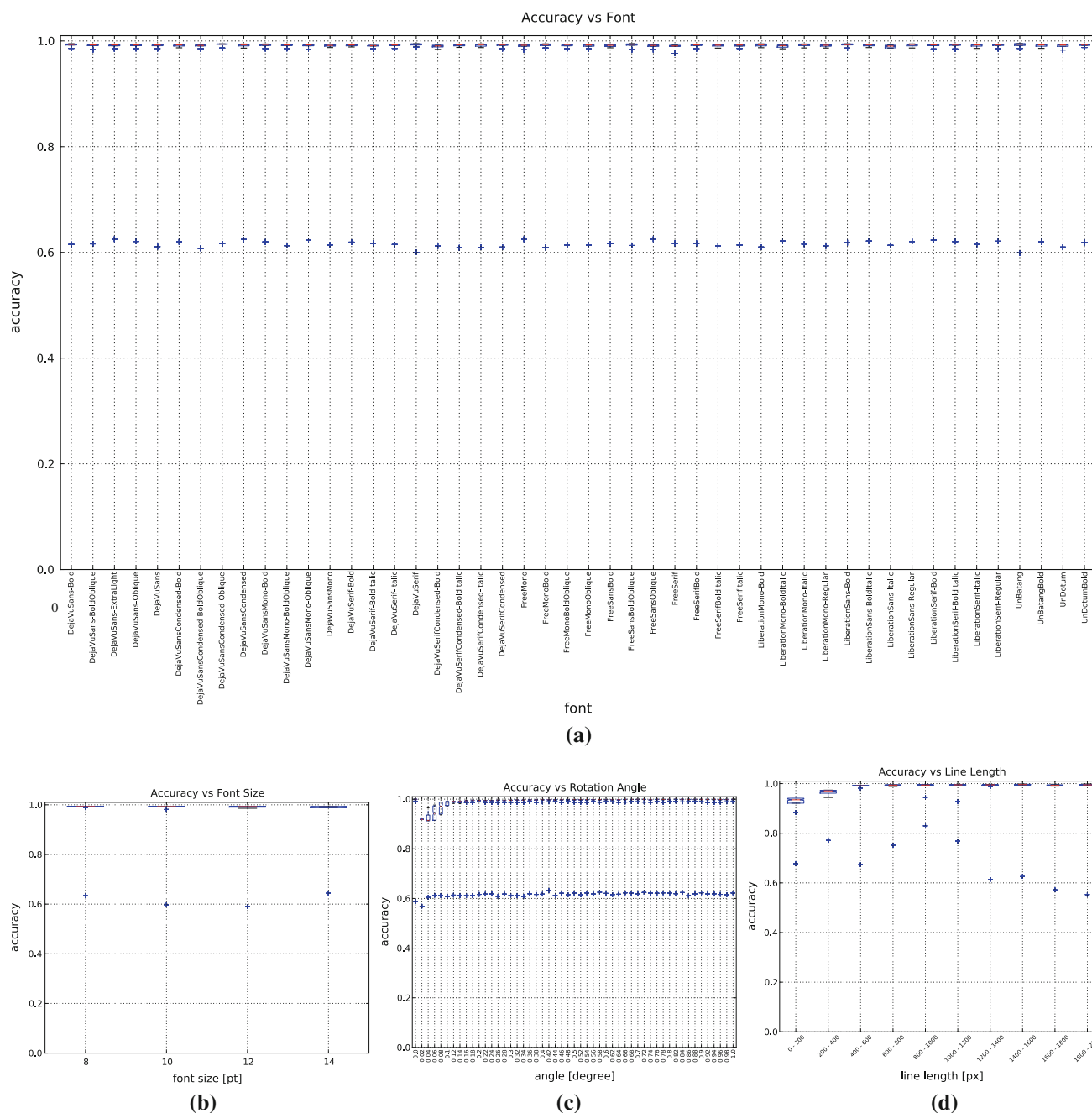


Fig. 14 Results for the orientation-based method on the DLT dataset. **a** Font. **b** Font size. **c** Mis-orientation. **d** Line length

with numbers, italic characters or ending with characters presenting these problems could be treated differently, e.g., by allowing more variation for these text-lines.

The problem of extracting the wrong alignment line could be encountered by extending the method to not only extract one alignment line but to extract more and compare them pairwise by computing their distance. This feature could then be used to detect suspicious alignment line constellations.

Taking in consideration all the results so far, one can say that for simple documents, the proposed method could be

applied on real-world documents, if such a dataset would become available. Several limitations, however, do exist: first of all, only simple document layouts have been used so far. Principally, application to more complex documents is also possible. The main challenge in this case is to extract the correct text-lines. This is a hard problem considering, e.g., multi-column and tabular content.

Another drawback is that the method can only detect forgeries if there is a mis-orientation or mis-alignment present. If the forgery method does not produce any of these distortions

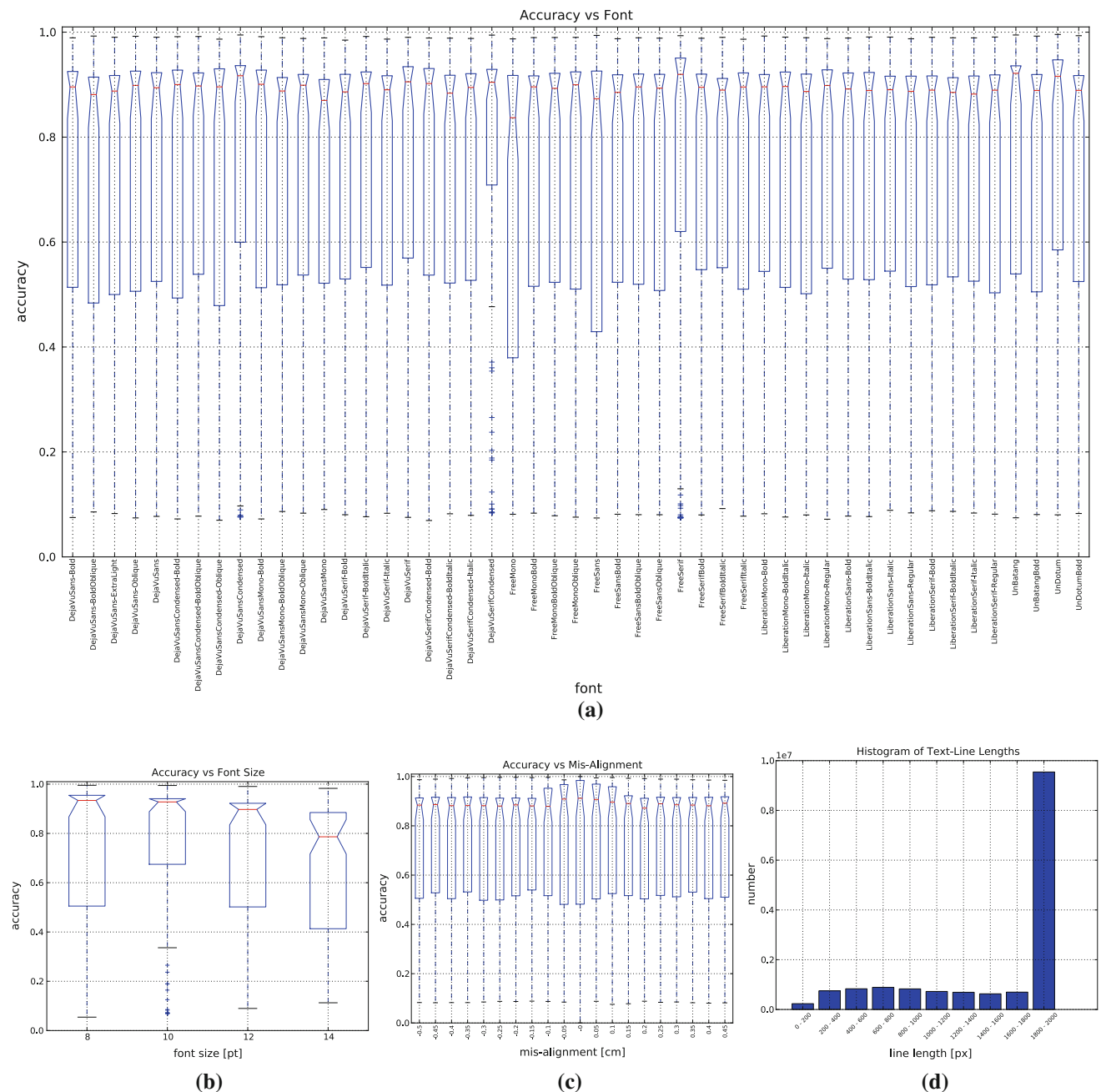


Fig. 15 Results for the alignment-based method on the DLT dataset. **a** Font. **b** Font size. **c** Mis-alignment. **d** Line length histogram

or if the forger is skilled, there will be two features that cannot be used to distinguish between forged and genuine documents.

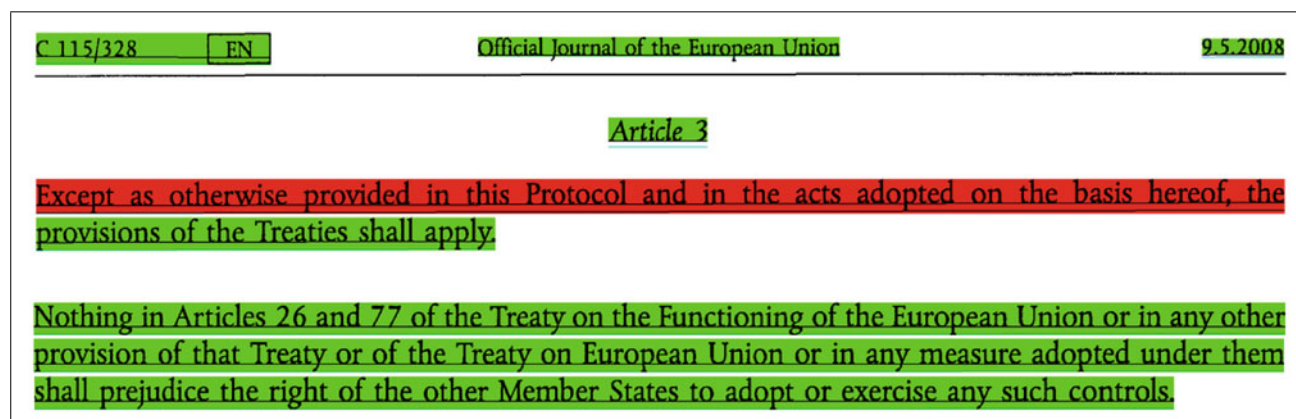
It should be noted that modifications of single characters or words cannot be detected with this approach. If the manipulated words are long enough, there is a chance that these could be detected. However, this is not the focus of this method. Modifications of single words or characters are more likely to appear in *SEP*-type forgeries. To detect these type of forgeries, other methods have to be applied as, e.g., [20].

5 Conclusion and future work

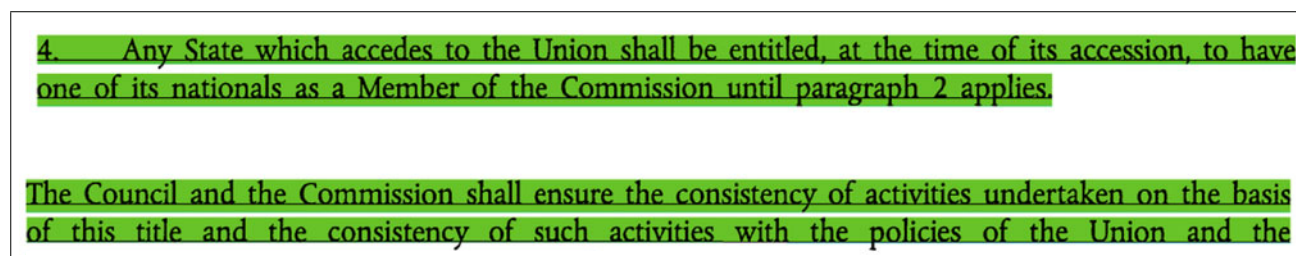
In this paper, the first approach for automatic plausibility checks for forgery detection on printed documents has been presented. The text-line skew angle and alignment features have been integrated into a statistical framework for automatically detecting implausible skew angles or alignment distances. Extensive evaluation of the proposed methods on different datasets has been done to show the usefulness of the approach.

accordance with the terms and conditions laid down by the Board of Governors, acting by a qualified

(a)



(b)



(c)

Fig. 16 Example of errors for the skew-base classification of forged text-lines. *Green lines* have been classified as valid text-lines, *red* ones have been classified as forged lines by the system. **a** Split line error: due to the curvature of the line, the text-line finding split the line into two parts, leading to a true *positive* and a false *negative* count for this text-line. **b** Deskewing error: as the forged part contains more text-lines

than the original lines, the deskewing process rotated the page such that the forged lines are perfectly horizontal. A genuine text-line is therefore classified as forged line. **c** Error due to the discriminative limitations of the feature. The last two lines have no significant difference in skew angle compared with the first two lines and are therefore classified as genuine

An important aspect for future work is the extension of the evaluation using real-world datasets or at least datasets that have been generated by a wide variety of different persons and methods, in order to measure the impact of the method in the different forging scenarios.

The overall goal of this and related work is to obtain a set of methods verifying different aspects of documents in order to find forgeries. The combination of these can hopefully help to reduce the error rate, especially the false positive rates, as these are often the most costly errors in high-volume applications.

6 Appendix: methods for generating forgeries

One major problem in the development of techniques for forgery detection is that it is not known how the common

person would forge a document. Also, no public statistical data are available giving an insight into how people forge documents. It is reasonable to assume that he will use standard hard- and software to solve this task, but even with this restriction, there are many possible scenarios leading to various kinds of defects.

In an effort to get an overview on how people would forge documents using standard hard- and software, a small experiment was set up. In this experiment, people were given two invoices and the following description:

Have a look at the invoice and consider the following scenario: a person with criminal energy, Frau Gaby Musterfrau, cannot have enough money and thinks of defrauding money from her insurance company. She recently had a technical problem with the car that is covered by her car insurance company. The idea she came up with is to somehow forge the original invoice

It shall be forwarded to the other institutions of the Community and shall be published, together with the replies of these institutions to the observations of the Court of Auditors, in the *Official Journal of the European Union*.

(a)

The Council and the Commission shall ensure the consistency of activities undertaken on the basis of this title and the consistency of such activities with the policies of the Union and the Community, and shall cooperate to that end.

(b)

Except as otherwise provided in this Protocol and in the acts adopted on the basis hereof, the provisions of the Treaties shall apply.

Nothing in Articles 26 and 77 of the Treaty on the Functioning of the European Union or in any other provision of that Treaty or of the Treaty on European Union or in any measure adopted under them shall prejudice the right of the other Member States to adopt or exercise any such controls.

(c)

HAVE AGREED UPON the following provisions, which shall be annexed to the Treaty on European Union and the Treaties establishing the European Communities,

Sole Article

(a) The European Parliament shall have its seat in Strasbourg where the 12 periods of monthly

(d)

Fig. 17 Example of errors for the alignment-based classification of forged text-lines. **a** Error leading to a false positive due to optical corrections on the italic character J. The italic “J” extends outside of the text-line boundary. **b** Error due to a wrong alignment line. The last line is falsely classified as not forged. The upper two lines are still correctly identified as forged because of the bad alignment on the right side.

c Error due to the discriminative limitations of the feature. The last three lines are not recognized as forged lines. The last three lines show forged lines with accurate alignment to genuine text-lines. **d** Error due to indentation: the line of the enumeration is slightly indented and is therefore falsely classified as a forged text-line

and increase the total amount of money stated in the invoice. She will then send the invoice to the insurance company and get more money back than she actually paid to have the car repaired.

Frau Gaby Musterfrau knows that the insurance company does not check the invoice with the repair shop. But that when something looks suspicious, the insurance company will do so. In that case, Gaby Musterfrau would be in trouble because she knows it is illegal to defraud the insurance company.

Your Task: Play the role of Frau Gaby Musterfrau and forge the invoice with the goal of increasing the total amount of money.

The candidates were not given any information about how they should or could forge the document. As the candidates did not perform the forgery under supervision, they were asked not to discuss any ideas with their colleagues in order to avoid biasing the results into one direction. However, it could not be verified if the candidates stuck to this request.

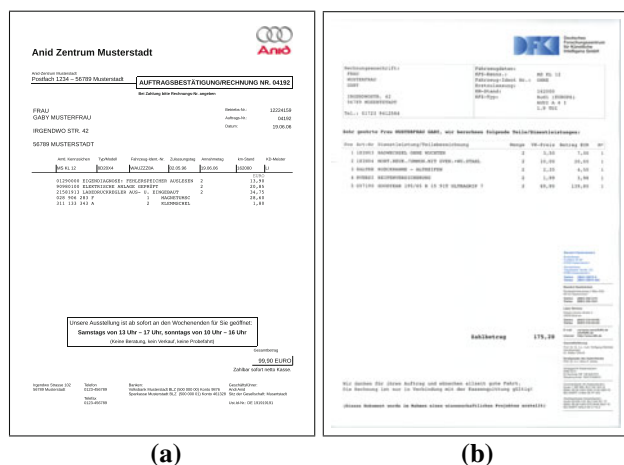


Fig. 18 Genuine invoices handed to the candidates for the forgery test. **a** Anid invoice, **b** DFKI invoice

The invoices that they were given were generated by the author using a word processing software. Genuine bills from previous car repairs were taken as a basis in order to have a realistic layout and to avoid copyright issues in case of publishing the dataset. Figure 18 shows the two genuine invoices. The “Anid” invoice was printed on normal white paper. The “DFKI” invoice was printed on old stationery from the DFKI with a logo on top and business-related information at the bottom right part of the page.

As the scenario implies that the test candidates are given a paper version of the invoice and hand back a forged paper document, diffusing the task by e-mail was not possible. Therefore, only a limited number of people could be asked for participation. Approximately 40 sample sets (cover letter, DFKI invoice and Anid invoice) have been distributed. A small award was advertised for the best³ forgery of each invoice.

Unfortunately, only 14 candidates delivered forgeries. In total, 25 forgeries were obtained, as some candidates submitted several forgeries using different approaches. The approaches can be divided into the following categories:

- **Print, Paste and Copy (PPC) Forgeries:** these forgeries are generated by replacing a part of the invoice. This is done by printing the new text (presenting higher values or more items) on an empty sheet and pasting this part onto the genuine bill. This is then copied using a color copier.
- **Reverse Engineered Imitations (REI) Forgeries:** this approach follows the idea of generating an editable document by imitating the genuine invoice. The genuine invoice is scanned and used as a template to generate

³ “Best” has not been concretely defined. It was just said that the most genuine looking invoice would win.

Table 2 Distribution of the forgery types, the forgery samples, and the background of the test candidates

	PPC	REI	SEP
# of samples	3	5	17
# of candidates	3	3	9
# of candidates with			
Computer science background	0	2	8
# of candidates with other background	3	1	1

a new document by retyping all the text, putting the logos in place, etc.

- **Scan, Edit and Print (SEP) Forgeries:** in this category the forgeries are generated by digitizing the invoice and manipulating the digital image. Here, mostly only numbers were modified, e.g. increasing the price of a tire or increasing the number of tire from two to four.

An overview of the frequencies of the different approaches is given in Table 2. The first line gives the number of samples that have been received for each class of forgery. The second line contains the number of candidates that provided a forgery of that type. It can be noted that the SEP forgery approach is by far the most commonly used approach. An analysis of the background of the test candidate gives a possible explanation: out of the 14 candidates, ten are computer scientists or study computer science, with an image processing background. The other four have a less technical background. In the table, this information is broke down to the number forgery types per background. This information is given in the last two rows of the table. It can be seen that the computer scientists tend to make REI and SEP forgeries, whereas PPC forgeries are mainly produces by persons with non-technical backgrounds.

Manual inspection of the documents has been done to see what peculiarities do forgeries show. Distortions, color, and font variations could be observed. For all the PPC copies, it could be observed that the text-line skew and alignment variations were significantly higher than for the other forgery methods. Alignment inconsistencies for single characters could be seen in some SEP forgeries. Thus, it can be concluded that the text-line alignment and skew angle are valuable measures for detecting potential forgeries. This observation is also supported by the fact that these features are used by questioned document examination experts [1].

Acknowledgments This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project PaREN (01 IW 07001).

References

1. Lindblom, B.S., Gervais, R.: Scientific Examination of Questioned Documents. pp. 238–241. Taylor and Francis, Boca Raton, FL (2006)

2. Smith, P.J., O'Doherty, P., Luna, C., McCarthy, S.: Commercial anticounterfeit products using machine vision. In: Proceedings of SPIE Optical Security and Counterfeit Deterrence Techniques V, vol. 5310, pp. 237–243. San Jose, CA, USA (2004)
3. Amidror, I.: A new print-based security strategy for the protection of valuable documents and products using moire intensity profiles. In: Proceedings of SPIE Optical Security and Counterfeit Deterrence Techniques IV, vol. 4677, pp. 89–100. San Jose, CA, USA (2002)
4. Hampf, N.A., Neebe, M., Juchem T., Wolperdinger M., Geiger M., Schmuck A.: Multifunctional optical security features based on bacteriorhodopsin. In: Proceedings of SPIE Optical Security and Counterfeit Deterrence Techniques V, vol. 5310, pp. 117–124. San Jose, CA, USA (2004)
5. van Renesse, R.: Ordering the order, a survey of optical document security features. In: Proceedings of SPIE Conference on Practical Holography IX, pp. 268–275. San Jose, CA, USA (1995)
6. van Renesse, R.: Protection of high security documents—developments in holography to secure the future market and serve the public. In: Proceedings of Holo-Pack, Holo-Print, Vienna, Austria (2006)
7. van Renesse, R.: Hidden and scrambled images—a review. In: Proceedings of SPIE Optical Security and Counterfeit Deterrence Techniques IV, vol. 4677, pp. 333–348. San Jose, CA, USA (2002)
8. Ali, G.N., Chiang, P.-J., Mikkilineni, A.K., Allebach, J.P., Chiu, G.T.C., Delp, E.J.: Intrinsic and extrinsic signatures for information hiding and secure printing with electrophotographic devices. In: Proceedings of the International Conference on Digital Printing Technologies, pp. 511–515. New Orleans, LA, USA (2003)
9. Mikkilineni, A.K., Chiang, P.-J., Ali, G.N., Chiu, G.T.C., Allebach, J.P., Delp, E.J.: Printer identification based on graylevel co-occurrence features for security and forensic applications. In: Proceedings of SPIE Security, Steganography, and Watermarking of Multimedia Contents VII, vol. 5681, pp. 430–440. San Jose, CA, USA (2005)
10. Khanna, N., Mikkilineni, A.K., Chiu, G.T.C., Allebach, J.P., Delp, E.J.: Survey of scanner and printer forensics at purdue university. In: Proceedings of the 2nd International Workshop on Computational Forensics. Lecture Notes in Computer Science, vol. 5158, pp. 22–34. Washington, DC, USA (2008)
11. Schulze, C., Schreyer, M., Stahl, A., Breuel, T.M.: Evaluation of graylevel-features for printing technique classification in high-throughput document management systems. In: Proceedings of the 2nd International Workshop on Computational Forensics. Lecture Notes in Computer Science, vol. 5158, pp. 35–46. Washington, DC, USA (2008)
12. Schreyer, M.: Intelligent printing technique recognition and photocopy detection for forensic document examination. In: Proceedings of Informatiktage 2009, vol. S-8, pp. 39–42. Bonn, Germany (2009)
13. Schulze, C., Schreyer, M., Stahl, A., Breuel, T.M.: Using DCT features for printing technique and copy detection. In: Proceedings of the 5th International Conference on Digital Forensics, pp. 95–106. Orlando, FL, USA (2009)
14. Breuel, T.M.: Robust least square baseline finding using a branch and bound algorithm. In: Proceedings of SPIE Document Recognition and Retrieval IX, pp. 20–27. San Jose, CA, USA (2002)
15. Shafait, F., Keysers, D., Breuel, T.M.: Performance evaluation and benchmarking of six page segmentation algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(6), 941–954 (2008)
16. van Beusekom, J., Shafait, F., Breuel, T.M.: Combined orientation and skew detection using geometric text-line modeling. *Int. J. Doc. Anal. Recognit.* **13**(2), 79–92 (2010)
17. Kanungo, T., Haralick, R.M.: An automatic closed-loop methodology for generating character groundtruth for scanned documents. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(2), 179–183 (1999)
18. Breuel, T.M.: On the use of interval arithmetic in geometric branch-and-bound algorithms. *Pattern Recognit. Lett.* **24**(9–10), 1375–1384 (2003)
19. Duda, R.O., Hart, P. E., Stork, D. G.: *Pattern Classification*, p. 49. Wiley, New York (2001)
20. van Beusekom, J., Shafait, F., Breuel, T.M.: Document signature using intrinsic features for counterfeit detection. In: Proceedings of the 2nd International Workshop on Computational Forensics. Lecture Notes in Computer Science, vol. 5158, pp. 47–57. Washington, DC, USA (2008)