Automatic Authentication of Color Laser Print-Outs Using Machine Identification Codes

Joost van Beusekom \cdot Faisal Shafait \cdot Thomas M. Breuel

Received: date / Accepted: date

Abstract Authentication of documents can be done by detecting the printing device used to generate the print-out. Many manufacturers of color laser printers and copiers designed their devices in a way to integrate a unique tracking pattern in each print-out. This pattern is used to identify the exact device the print-out originates from. In this paper we present an important extension of our previous work for (a) detecting the class of printer that was used to generate a print-out, namely automatic methods for (b) comparing two base patterns from two different print-outs to verify if two print-outs come from the same printer and for (c) automatic decoding of the base pattern to extract the serial number and, if available, the time and the date the document was printed. Finally, we present (d) the first public dataset on tracking patterns (also called machine identification codes) containing 1264 images from 132 different printers. Evaluation on this dataset resulted in accuracies of up to 93.0% for detecting the printer class. Comparison and decoding of the tracking patterns achieved accuracies of 91.3% and 98.3% respectively.

Joost van Beusekom German Research Center for Artificial Intelligence (DFKI) D-67663 Kaiserslautern, Germany Corresponding author: Tel: +49-631-20575-4182 Fax: +49-631-20575-4020 E-mail: Joost.van_Beusekom@dfki.de

Faisal Shafait German Research Center for Artificial Intelligence (DFKI) D-67663 Kaiserslautern, Germany E-mail: Faisal.Shafait@dfki.de

Thomas M. Breuel Department of Computer Science, University of Kaiserslautern Image Understanding and Pattern Recognition Group D-67663 Kaiserslautern, Germany E-mail: tmb@informatik.uni-kl.de **Keywords** machine identification code, counterfeit protection, document authentication, color laser printer identification

1 Introduction & Related Work

J. Hails [10] defines authentication in the context of documents as "showing that writing is what it is claimed to be". In the context of printed documents, the focus is to make sure that the print-out has been created by the person or company that it claims and that all the contents are genuine.

In every-day life, people heavily rely on the genuineness of documents of many kinds: e.g. insurance companies need to make sure that the invoices they process are genuine and have not been altered; employers have to assure that the doctor's note the employee handed in is original; companies need to make sure that the diplomas that the candidate presents are genuine, etc. While in many cases this authentication is done by *manual* visual inspection, this approach cannot be followed in cases where the volume of documents to be verified is just too high as in the case of automated invoice processing systems.

Authentication is often ensured by various types of signatures: these represent a hard to forge feature that only the creator may have added to the document. It can thus be used to prove the document's originality.

Signatures have always been a critical issue, even in ancient times, where the number of paper documents was limited, compared to their tremendously wide-spread use nowadays. The signet rings from the monarchs that were used to sign the documents in ancient times have nowadays been replaced with all kinds of modern security features. A good overview over signature features for document security is presented by van Renesse [31]. Several categories of features are distinguished, depending on the effort to detect or verify them. Three different levels effort, called *level of inspection* are commonly being distinguished:

- first line inspection: this groups all the inspection methods that can be done using only the unaided human senses. Examples of signatures that can be used in a first line inspection are watermarks and holograms.
- second line inspection: inspections requiring additional tools to verify a feature or a document fall into this category. Examples are inks that are only visible under ultra violet light or bar codes that are only readable using a bar code scanner.
- third line inspection: this category covers the more sophisticated analysis that is normally done by experts, e.g. questioned document examiners. Examples are physical and chemical analysis of the ink composition to date a document.

There is a vast variety of different features available to make documents relatively secure even through first line inspection, e.g. using special sort of paper, eventually integrating a watermark, by adding holographic images [26], specialized printing techniques [2] and other physical and chemical signatures [11]. Many other types of features can be found in literature ([30,33,32]). Having holograms, bar codes or other extra security features is certainly a reasonable way to make documents more robust against tampering. They require, however, extra steps before or during the creation of a document, which will dramatically increase the costs of producing these. A more practical solution has to be found for assuring the authenticity of documents of every day life without adding extra features.

Therefore, the focus is on using features "inside" the document as a signature, the so called *intrinsic* features. In contrast to *extrinsic* features that are added only for document security purposes, intrinsic features are byproducts of the normal document generating procedure. This has the advantage that the creators of the documents can continue to use their usual technique to generate the documents while achieving a certain degree of tampering resistance.

Several uses of intrinsic features have been presented in previous publications: printer identification, the process of assigning a print-out to a unique printer or a printer type, has been intensively be studied by different groups. Mikkilineni et al. [1,14,13] present features that can used to determine the model of laser printer that was used to print a document.

Choi et al. [8] present an approach for identification of single printers using statistics as e.g. skewness, kurtosis and correlation to train a support vector machine (SVM) that is than used to assign a print-out to its originating printer. Results are given on a non-public dataset containing print-outs of only 9 different printers.

Schreyer et al. [22,21,23] worked on detecting the printing technique used to print a document. Also classification between printed documents and copied versions of printed documents has been analyzed. Using discrete cosine transform (DCT) features, good performance could be shown even when scanning with moderate resolutions of 400 dpi.

Jiang et al. [12] presented an approach also using DCT features to identify the printer model. Multi-Size block DCT coefficients are extracted to train an SVM, that is used in a second step for classification. Evaluation results are given on a very small dataset, consisting of 6 printers only.

Another frequent intrinsic document feature is handwriting. Many printed documents contain handwritten parts as notices or signatures. Two related questions can be distinguished: off-line writer identification and off-line signature verification. In this context, off-line means that only the image of the signature or the handwriting is available, in contrast to online data, where stroke information is also used. The first problem consists in identifying the writer of a document in question using a previously trained model of the writers handwriting [19,20,3]. In signature verification the question is whether a signature on a document has been generated by the person claimed by the signature or if someone else forged the signature [7,17].

The introduction of color laser printers and color copiers has made it very easy even for unskilled people to generate high-quality forgeries. Therefore, manufacturers designed their devices in a way that they print a tracking pattern on every print-out. This tracking pattern, also called counterfeit protection system (CPS) or machine identification code (MIC), is unique for each printer and can thus be used to identify the device that was used to generate the print-out.

The CPS codes consist of small yellow dots (see Figure 1 for an example) that are invisible for the unaided human eye. These dots form a pattern that is repeated many times on a page with a fixed horizontal and vertical spacing. The exact amount of spacing is manufacturer specific and can be used to identify the latter. This repeating pattern is defined as the *base pattern*. It contains information about the printer that was used to generate the print-out, e.g. the serial number of the device. Each printer generates a different base pattern and for some printers even the date and the time when the print-out was generated is encoded [9].

As these CPS codes have been introduced by purpose by the manufacturers, decoding these codes is also possible if the decoding scheme is known. In general, the procedure that a person with the necessary authorization level has to follow in order to find out the exact printer used for generating the questioned print-out is the following: the document is send to a licensed laboratory that extracts the so called *inspection code*. This inspection code can be used by the printer manufacturer to identify the printer that was used and, by comparison to the customer data base, the presumed owner of the printer can be identified.

For privacy reasons, the customer data base is not available for a "public" setup. Furthermore, the information about how to decode the base patterns is classified. In many applications, however, this is not needed: if authentication of a document is needed, e.g. if one needs to make sure that the origin of the document is what the document claims to be, then it may be enough to identify the printer class or to compare the questioned document to a previously known genuine document: if their patterns are identical the questioned document comes from the same device. If they differ, it can be concluded that the



Fig. 1 Examples of CPS dots on a print-out. The center and the right image have been enhanced to make the dots visible.

document comes from a different source¹. A modification of this approach can also be used for automatic decoding of the base pattern, which is demonstrated on the Xerox-type patterns for which the decoding scheme is known [9].

In this paper we considerably extend our previously presented work [28] on (a) detecting the class² of printer that was used to generate a print-out by methods for (b) comparing two base patterns from two different print-outs and (c) automatic decoding of the base pattern, if the decoding scheme is known. These extensions are important, because they are needed to use the CPS codes for authenticating documents on a printer level, instead of only on a printer class level. The the authors' best knowledge, this is the first approach to use automatic processing of the CPS codes for identifying the originating printer. The main novelty of both approaches lies in the fact, that by our methods, the CPS codes can be used in a productive setup by everyone for authenticating documents. To the authors' best knowledge, no such a method has been published before. Also, the paper presents solutions to solve the comparison and decoding tasks, given the noisy nature of the extracted dots.

Finally we also present (d) the first public dataset on machine identification codes containing 1264 images from 132 different printers.

- (a) Detection of the manufacturer is done by automating and extending the methodology proposed by Tweedy [27]. By detecting the distance between reoccurring patterns, the so-called horizontal and vertical pattern separating (HPS and VPS) distances can be computed. An approximate assignment to the printer manufacturers can be done on the basis of these two measurements.
- (b) By exploiting the redundancy of the repeating base pattern, a method is developed to extract representative candidates of the pattern for comparison to the pattern of a different document. Significant differences in the frequencies of appearing dots can be used to decide if both patterns are identical or not.
- (c) Finally this approach is extended to extract the base pattern which can be used for decoding the pattern, if the decoding scheme is known.
- (d) To evaluate the proposed methods a new dataset has been generated in cooperation with the Electronic Frontier Foundation (EFF). The printouts collected by the EFF have been scanned and manually ground-truthed resulting in a dataset with 1264 print-outs from 132 different printers.

The remaining parts of this paper are organized as follows: Section 2 describes the extraction of the CPS dots. Printer class identification is described in Section 3. Section 4 presents the method for comparison of the CPS codes of two pages. In Section 5 the approach for automatic decoding of the CPS base pattern is presented. Details about the newly generated dataset can be found in Section 6. Evaluation and results are given in Sections 7, 7.5 and 7.6. Section 8 concludes the paper.

 $^{^1\,}$ If one source uses more than one color laser printers, the questioned document has to be compared to several base patterns instead of just one.

² Roughly speaking, these classes can be assigned to different manufacturers

2 Extraction of the CPS Dots

In a first step, the CPS dots have to be extracted. The dots, also called *yellow* or *tracking* dots are small yellow dots printed on the document. Their size is about 0.007 inch (approximately 4 pixels in a 600 dpi scan). Examples are shown in Figures 1 and 2.

Several ways are possible to extract the tracking dots: MIT Seeing Yellow [15] proposes to extract the blue channel. Conversion to CYMK and using the Y channel is also possible. A subsequent binarization step [16,18,24] could be used to decide between foreground (dot candidate) and background. This approach, however, leads to many false candidates.

Therefore, a manually tuned extraction was chosen: examining the color values of the dots on different pages showed that in RGB color space the R and G values are close to 255, while having a slightly lower B value. The following binarization method was deduced:

$$I(x,y) = \begin{cases} 0, & \text{if } \min(I_R(x,y), I_G(x,y)) - I_B(x,y) < T_1 \\ & \text{and } I_R(x,y) > T_2 \\ & \text{and } I_G(x,y) > T_2 \\ 255, & else \end{cases}$$
(1)

where $I_R(x, y)$, $I_G(x, y)$ and $I_B(x, y)$ are the intensities of the red, green and blue channel of image I at position (x, y), 0 being black and 255 being white. A good choice for the thresholds is $T_1 = 20$ and $T_2 = 240$. These thresholds have been manually set by observing the RGB values of the yellow dots from different documents. The result of this step is a binary image where the black pixels represent dot candidates.

Problems occur if areas with dithered colors are present in the image. After binarization, these areas tend to show significant amounts of pixel noise similar in size to the ones form the CPS code. As these may be very numerous



Fig. 2 Examples of CPS dots on a print-out. The left picture shows the dots by making them visible using a blue LED light source. The right image shows a detail of a high resolution image taken from a part of the page. From the visual appearance of this pattern, it is likely, that an HP color LaserJet was used to print this document [27].

it is preferable to remove these dots to make the system more robust. This filtering is done by morphological closing (dilation followed by erosion) using a quadratic mask. The effect is that dots closer than half of the width of the mask are being connected together, whereas singular dots remain unchanged. A computationally fast implementation using run length encoding for binary morphology was used [6]. The width of the mask is fixed to half of the median distance between the neighboring connected components, which is normally about 12 px.

From the resulting image, the connected components are extracted and filtered: big connected components (width or height bigger than 4 px) are being ignored. This also removes the dithered parts of the image, as these remain connected after the filtering.

Example images in Figure 3 show the intermediate results for each step. It should be noted that the resulting set of dots is not perfect, in the sense that there may be noise dots and that dots might also be missing due to the following two reasons:

- overprinted dots: CPS dots are frequently overprinted by text or other document content portions. This leads to a significant number of dots that are not detectable and thus leads to an overall incomplete pattern.
- noise dots: some extracted dots may not represent CPS dots. These come from binarization, page contents in yellow color and noisy print-outs due to e.g. old drums.
- base pattern ambiguity: extracting a unique base pattern is less trivial than it may seem: the separating distances do not give any information about where the base pattern starts and ends, thus leading to ambiguities when extracting a base pattern. Moreover, the exact size of the base pattern is unknown. An example for this ambiguity is shown in Figure 7.

This will lead to increased requirements concerning the robustness for the analysis steps.



Fig. 3 Visualization of the CPS dot extraction: (a) shows binarized image using the task specific binarization. After dilation image (b) is obtained. Erosion of (b) leads to (c). Finally, using connected component based filtering the final image (d) is obtained.

3 Printer Class Identification

The proposed method for printer class identification uses the the vertical pattern separating distances feature and a classification scheme proposed by Tweedy [27]. This idea is extended to use also the horizontal pattern separation (HPS) distance to cope with the problem that depending on the paper feed inside the printer, the CPS codes may be rotated by 90°.

3.1 Computation of Horizontal and Vertical Pattern Separation Distances

For reading simplicity, in the following, only the computation of the HPS distance is discussed. The VPS distance follows an analogous scheme, just switching the directions x (horizontal) and y (vertical).

The approach for computing the HPS takes a random local subset of the tracking dots and matches this subset to the remaining dots at approximately the same y position. For each obtained match the translation parameter t_x . As random selection of a local subset may lead to noise patterns being matched, the whole process is run several times. Statistics on all the obtained translation parameter values are used to estimate the HPS and VPS distance of a pattern. A visualization of the main idea is given in Figure 4.

The reason for only considering matches on approximately the same y position lies in the fact, that not all CPS pattern repetitions are on a regular grid. Some CPS codes repeat their *base pattern* shifted in x direction in neighboring columns. A base pattern is defined as the smallest set of dots that can not be split into equal sub patterns and that explains all the tracking dots in



Fig. 4 Computation of HPS and VPS distances: first, a sub pattern is selected. This is matched at different positions in the same column or row respectively. The computed translation parameters in x and y direction are used to extract the HPS and VPS distance of the pattern.

the image by just translating it in x and y directions. An example of such a repetitive pattern with offset is shown in Figure 5(a). Without this restriction, the matching would return distances that represent only a fraction of the real distance.

Small variations of t_y have to be allowed to cope with small skew angles α of the document page. Since $\alpha = atan \frac{t_y}{t_x}$, it becomes clear that the method cannot work, if the skew angle becomes to high, relative to t_x , which is a priori unknown. Therefore, t_y is allowed to vary only little, $\pm 0.75 \times \theta$, where θ is the median distance between closest neighboring point.

The local pattern subset has to be chosen carefully: it must be assured, that its width is not bigger than the smallest known HPS distance. Elsewise, only multiples of the distance will be found. Another constraint is that the pattern should be big enough to allow for robust matches. If only a few (e.g. two or three) dots are to be matched, many matches on random positions will be found. Therefore, for each direction a different search pattern is used: starting from a randomly selected dot, an area around this dot is defined such that in the direction of measurement its size is smaller than the smallest known pattern separation distance (currently 0.16 inch). In the other direction, it is extended to a wider area to cover more points. This allows the method to find more robust matches. A visualization of such search patterns is given in Figures 5(c) and 5(d).

3.2 Matching the Search Pattern

After choosing the search patterns, the positions where the dot pattern in the same geometric relation can be found have to be computed. This is done using the technique described by Breuel in [4]. It uses an optimal branch-andbound search algorithm, called RAST (Recognition by Adaptive Subdivision of Transformation Space). This method allows robust and accurate finding of the positions of repetitions of the search patterns. For completeness, a short overview of the algorithm is given here.

RAST optimizes a quality function that is defined as the number of model points (dots from the search pattern) matching an image point (remaining dots) under the error bound ϵ . The RAST algorithm uses a branch-and-bound search for quickly finding a global optimum for the given quality function. The method uses a priority queue containing parameter subspaces ordered by their upper bound quality. The highest upper bound quality subspace is divided into two new subspaces, by splitting it into two parts of equal size. For each part, the new upper bound quality is determined and both subspaces are added into the priority queue. These steps are repeated until a stopping criterion is met.

The RAST algorithm performs a branch-and-bound search on the parameter space (also called transformation space). As for each direction, a separate search is done, the transformation space can thus be reduced to parameter $t_x \in [-W, W]$ for the HPS distance (in the case of the VPS distance: $t_y \in [-H, H]$) where W and H are the page width and height and t_x is the



Fig. 5 Example of different patterns: Image (c) and (d) show an example of a horizontal and a vertical search pattern. Image (a) shows an example of a non-aligned pattern. Image (b) shows the result of the matching: the blue crosses represent the sub pattern to be searched for, the red ones represent the repeating patterns above or below the subset pattern. The alignment has been slightly displaced in order to be able to see the different crosses.

translation parameter in x direction. To be more robust against small distortions of the page when matching in horizontal direction, t_y is set to allow for small variations, too. Examples of the matching results can be found in Figure 5(b).

3.3 Estimating HPS and VPS Distances

The HPS and VPS distances are measured using recurring translation values: if a pattern has a certain HPS distance, the translation values returned by the search are most likely multiples of the HPS distance, apart from a few "noise" matches. Using several iterations of the search, a histogram of translation values is generated that shows characteristic peaks having a distance approximately equal to the HPS distance.

As the selection of the search pattern is a random process, the selected search pattern may not represent any pattern but e.g. only noise dots. It may also happen that a search pattern is chosen that is repeated inside the base pattern. Therefore, several iterations of search pattern selection and matching are run, each returning a set of matching parameters. All these results are collected in one priority queue $R = \{(t_{x,1}, q_1) \dots (t_{x,n}, q_n)\}$, where q_i is the quality of the match i and n is the total number of results returned by all



Fig. 6 Histogram of the values of the parameters t_x . One can observe the peaks with the constant distance that is equal to the HPS distance in pixel.

iterations. The quality q_i is the number of dots from the search pattern that could be matched to the remaining dots. This quality is used to select the best m matches for generating the statistics, as matches with higher quality lead to a higher degree of robustness.

A histogram of the translation values t_x of the *m* best results is generated by computing all pairwise distances. An example of such a histogram is shown in Figure 6. From this, the HPS distance is computed by histogram comparison: for all possible HPS distances, a reference histogram is generated. It is generated by distributing equally high peaks in the histograms at all different distances for HPS. All the reference histograms are compared with the measured histogram using Jenson-Shannon-Divergence (JSD) [25]. The reference histogram with the smallest JSD gives us the HPS distance.

4 Comparison of CPS Patterns

After extracting the CPS dots of both pages for which the CPS patterns will be compared, a robust method of comparison of these patterns has to be defined, since the set of extracted dots is not perfect (see Section 2).

The straightforward comparison of both sets of points is difficult, since the set of dots on both pages largely depends on the document's content. To avoid this problem, an indirect approach of comparison is chosen: the redundancy of the base pattern is used to generate a prototype pattern model that not only contains the dots on their positions but also their frequencies of appearance.

The idea is that noise points are randomly distributed and will be present in very different positions throughout all the repetitions of the base pattern. The dots belonging to the pattern, however, will be present at always the same position of the base pattern. By this, an implicit distinction between



Fig. 7 Example demonstrating the prototype ambiguity: from the dot image in the center, several possible prototypes might be extracted if the start and end points are not known.

dots belonging to the pattern and noise dots can be done: if the two extracted prototypes significantly differ in one dot, e.g. a dot being very frequent in the first prototype pattern but being very rare in the second prototype pattern, the two patterns are considered to be different. The next section explains how the prototype model is computed. Section 4.2 explains how the comparison of two prototype patterns is done.

4.1 Computation of the Prototype Pattern Model

Determining the prototype model is done by using the redundancy of the basepattern inside the page. The first step consists of computing the approximate size of the base-pattern. This can be done using an approach presented in Section 3. Using the HPS and VPS values as an approximate size of the base pattern, a *prototype pattern* is extracted at a random position on the page. This pattern contains all dots inside the window of width VPS and height HPS around a randomly selected dot.

To get a robust prototype pattern, the initial one is used as a model that is matched on the remaining dots of the page. Again, RAST [5] is used for this step in the following way:

```
00 randomly select prototype pattern S
01 match S to the remaining dots
02 for each of the 10% best matches:
03
       compute target area of matched prototype
04
       for each point p in target area:
05
           select the closest point c from the prototype pattern
           if d(c, p) < D: // Manh. distance
06
07
               increase frequency count for p
08
           else:
09
               add p to prototype pattern
10 Repeat for I iterations from line 01 on
```

In line 09, the dot p from the target area is added to the search pattern, if the distance is higher than a predefined threshold (normally set to few

pixels). This step is necessary to add dots that could belong to the prototype pattern but that where not present in the first, randomly selected prototype set. If these points would not be added, a missing dot in the initial prototype pattern could not be compensated.

The approach is robust to an initially bad choice of the prototype pattern: as the noise dots are randomly distributed on the page, even a noisy prototype pattern is likely to be matched to genuine CPS dots. This will continuously increase the quality of the prototype pattern. Examples of two prototype patterns can be seen in Figure 9.

4.2 Prototype Comparison

Due to the afore mentioned ambiguity of the base-pattern, comparing two prototype patterns from two images directly is not a trivial task.

To overcome this problem, the prototype from one image is matched to all the dots of the other image. For each dot from the prototype model the frequencies with which they can be matched to the dots of the second image are computed. If a dot is frequently present in the prototype model, it should also be frequent in the other image. If not, both patterns differ. If noise dots are present in the prototype pattern, they will have a low frequency. After matching the prototype to the second image, noise dots should still have a low frequency.

A threshold has to be fixed defining what differences are considered as significant. Consider a normal CPS pattern on a standard page: it should be easily possible to find multiple occurrences of the pattern on that page. One or the other dot might be missing but it is unlikely that by chance always the same dot is missing. On the other side, if noise dots appear, it is unlikely that they will appear in the same position for different repetitions of the base pattern. Therefore, the threshold is set to a high value: infrequent noise will be discarded and for the relevant pattern dots, minor variations in frequency



Fig. 8 Examples of extracted prototypes. The darker the cross is the higher its frequency of occurring in a match.

Fig. 9 Examples of the fully occupied and the extracted Xeroxtype base patterns.

will not influence the outcome. A pattern is defined as different if there is at least one significant difference.

In order to make sure that the second page does not contain any frequent dots that are not present in the first prototype pattern, the comparison has to be done in both ways. Two pages are considered to be different if at least one significant difference in dots frequencies can be found. The threshold defining a significant difference is set to 0.9.

5 Extracting & Decoding the Xerox-Type Base Pattern

Some color laser printers and copiers produce CPS patterns that also encode the date and the time a print-out was generated into the pattern. In this case the CPS pattern of two print-outs from the same printer will differ, which will result in a wrong output of the CPS comparison step.

In this part we present a solution to this problem: we use the previously presented approach to extract the base pattern. This base pattern is then decoded and the comparison can be done on the level of the decoded string. As currently only the decoding for the Xerox DocuLaser is publicly available and as this seems to be the only printer type encoding the time and the date, we explain the method for the application on Xerox-type CPS patterns.

The extraction of the base pattern follows the same approach as the extraction of the prototype model: the sole difference lies in the selection of the initialized search pattern: instead of taking a randomly positioned pattern, a fully occupied base pattern for the CPS pattern type is used. This is a pattern where each crossing on the regular grid of the Xerox pattern is occupied by a dot. An example of a fully occupied Xerox pattern can be found in Figure 9.

After the prototype matching each point is associated with a certain frequency. Based on this frequency it is decided whether to set each dot as either *on* or *off*. The threshold for this operation can be set at a relatively low level. Without any tuning, 0.1 showed to give suitable results. An example of the fully occupied and an extracted base pattern can be found in Figure 9.

The decoding of the pattern is done by applying the scheme presented by the Electronic Frontier Foundation [9].

6 The Machine Code Identification Dataset

No public dataset was available for testing the proposed method. In cooperation with the Electronic Frontier Foundation (EFF), the first public dataset for evaluating methods analyzing CPS codes was generated³: in an attempt to gain more information about the CPS codes, the EFF invited the visitors of their web site to send print-outs of the provided test documents, together with additional information as e.g. the manufacturer, the printer model and its serial number. Those documents were scanned and manually ground-truthed.

³ The dataset can be downloaded from: https://madm.dfki.de/downloads-ds-mic

The scanner used for this task is a Fujitsu fi-4120C2 automatic document feeding scanner with a maximum optical scanning resolution of 600 dpi and a color depth of 24 bit. All documents were scanned using 600 dpi in full color mode.

A set of 132 sets of test print-outs has been scanned. Each set consists of 8 pages⁴ from the EFF printer test set sheets⁵. An example of such a sample set of 8 images is given in Figure 10. On a set level, the ground truth has been generated manually by using the information that was handed in on the cover sheet or by supplemental print-outs of e.g. configuration pages of the printer. The ground truth contains the following information:

- manufacturer of the printer / copier
- serial number of the printer / copier used to generate the print-outs
- presence or absence of dots
- manually measured horizontal pattern separation distance
- manually measured vertical pattern separation distance
- vertical pattern separation distance according to the classification given in the paper by Tweedy [27].
- miscellaneous information

7 Evaluation & Results

Different evaluation procedures have been used to measure the performance of the different approaches presented in this paper. All methods use the previously introduced dataset.Detailed information about the evaluation setups for the different methods are given in the following sections.

Despite the existence of related methods in the area, a comparative analysis could not be done, since none of the datasets used by other researchers has been made public. Furthermore, most of these datasets only contained very few samples from few printers.

 4 Some sets are not complete, others have extra pages, e.g. printer configuration pages.

 $^{^5}$ http://www.eff.org/wp/investigating-machine-identification-code-technology-color-laser-printers#testsheets



Fig. 10 Test sheets from the Electronic Frontier Foundation.

7.1 Evaluation of CPS Code Classification

For the evaluation of the classification based on the HPS and VPS distances, the different classes have to be defined: Tweedy [27] proposed thirteen different classes of VPS distances. For the 0.32 and 0.64 inch distances, Tweedy distinguished two different subclasses based on different visual characteristics. As these subclasses have the same VPS distance, these are merged into the same class for the current experiments. Thus, eleven different classes are considered: no CPS pattern, 0.16, 0.32, 0.48, 0.50, 0.54, 0.64, 0.69, 0.96, 1.20 and 1.28 inch distances. A list of printers corresponding to the different VPS distances is given in [27]. A list of HPS and VPS pairs and the associated printers can be found in Figure 11.

As two different measures are computed per page, three different accuracy rates are being computed: the accuracy of computing the correct HPS distance, the accuracy of computing the correct VPS distance and the accuracy of correctly computing both VPS and HPS distances. A list of the different pairs of HPS and VPS distances with more than five occurrences is presented in Table 1. In total 19 different pairs could be identified in the dataset.

Sheet 6 (Figure 10(g)) of each sample contains the most realistic document type for the proposed scenario, namely a page containing mainly text regions. As not all samples are complete, in total 128 images have been used.

The computation of the HPS and VPS distances is done using the method described in Section 3. The classification is done using a simple threshold: if the difference between the ground truth distance and the computed distance

0.48 0.96	0.64 1.28
Epson Aculaser C1900	Epson Aculaser C1100
Epson AcuLaser C900	Epson Aculaser C3000
Epson AcuLaser C1500	Dell 3110CN
Konica Minolta Magiccolor 2300 DL	Dell 5100CN
Minolta QMS Magicolor 2210	Xerox Docucolor 1632
Minolta QMS Magicolor 2300DL	1.28 0.64
Konica Minolta Magicolor 2300 DL	Xerox Phaser 790
	Xerox DocuColor 6060
0.54 0.69	Xerox DocuColor 2045
HP Color Laserjet 3700DN	Xerox DocuColor 12
HP Color LaserJet 4700	0.96 0.48
HP Color LaserJet 3700DN	Minolta DialtaColor CF 2001
HP Color LaserJet 2840	Minolta-QMS Magicolor 7300
HP Color LaserJet 2600N	Konica Bizhub C252
HP Color LaserJet 2605DN	Konica Minolta Bizhub 350C
HP Color LaserJet 4600	-1 -1
	Xerox Phaser 8400N
0.69 0.54	Xerox Tektronix Phaser 7750DN
Lexmark C910	OKI C9200 Model N31061A
Lexmark C912	Samsung CLP-500/XAA
Kyocera C2630D	Brother HL-2700 CN
HP Color LaserJet 5500	OKI C5150n
HP Color LaserJet 5500DTN	
HP Color LaserJet 5500DIN	

Fig. 11 Excerpt of the list of printer classes in the evaluation dataset. The HPS and VPS class is given at the top, followed by the printers in this class. An other list of printers can be found in [27].

HPS [inch]	VPS [inch]	# Occurrences
0.54	0.69	34
-1.00	-1.00	31
0.48	0.96	17
0.69	0.54	11
0.64	1.28	7
1.28	0.64	7
0.96	0.48	6

Table 1 List of pairs of HPS and VPS distances occurring more than five times in the dataset. Documents without dots are represented with the pair (-1.0, -1.0). A -1.0 in any other row means that during ground truth generation, the respective HPS or VPS distance was not measurable due to a sparse pattern (see Figure 13 for examples of sparse patterns).

is less than 0.02 inch, it is considered to be in the same class. If no tracking dots are present, no reasonable matches for the search patterns can be found. The document is then considered to be CPS code free.

To test the effect of lower resolutions, the tests have been run in different resolutions of 600 dpi (original resolution), 400 dpi, 300 dpi and 200 dpi. The lower resolutions have been obtained by downscaling the images.

As the documents in the dataset were not deskewed after scanning, some show considerable amounts of skew leading to possibly wrong measurements of the HPS and VPS distances. Therefore the tests have been run on the original images as well as on the deskewed images. A previously published automatic deskewing method [29] has been used to generate the deskewed images.

7.2 Evaluation of CPS Code Comparison

The test setup for the evaluation of the CPS code comparison is as follows: a set of image pairs was defined to compare print-outs with same HPS and VPS distances. For each sample set, Sheet 3 was compared with Sheet 6 and Sheet 6 was compared with Sheet 3 of the numerically next sample set. If Sheet 3 is not present for a sample set, the next Sheet in numerical ascending order is taken (Sheet 4, etc.). By this method a set of 184 comparisons was generated. The ground truth HPS and VPS distances were given as input.

However, as some Xerox printers encode time-variable information into the CPS codes, the comparisons of Xerox print-outs from the same printer have been manually analyzed to see if the patterns vary over time. If this is the case, the expected outcome has been changed to "different", although the patterns come from the same printer.

The evaluation measure is the number of correctly computed outcomes of the comparison. The threshold which decides if a dot significantly differs from one pattern to the other is set to 0.9.

7.3 Evaluation of CPS Pattern Decoding

For testing the decoding method for the Xerox-type CPS pattern, all the test sheets including the cover page (if available), from samples with Xerox-like patterns were used. Extra sheets have been ignored as some contain blackand-white only print-outs. The extracted information was compared to the ground truth information. If the ground truth information reflected parts of the decoded information, and the decoded information is consistent, the decoding is considered to be correct. This, in most cases, is expressed by parts of the serial number that could be extracted from the CPS pattern. As this criterion of success is a fuzzy one, the extracted information and the relevant ground truth information for each file is given in the result section in order to show the success of the method.

7.4 Results for CPS Code Classification

The results for the accuracy of the CPS code classification on the original images for the different resolutions can be found in Table 2 and the results on the deskewed images in Table 3.

First of all, it can be seen that the accuracies decrease with the resolution. This is not surprising as more and more dots will be missed, the smaller the resolution will be.

Second, it can be noted that the accuracies on the deskewed images are higher than on the skewed ones. Due to the slight rotation, the distances between repeating patterns will be slightly increased but also the matching of the search pattern will return less results leading to a less robust estimation of the HPS and VPS distances. From this, we can conclude that the skew angle decreases the performance of the method. This problem, however, would be rare in real world high-volume scanning scenarios, since deskewing is in most cases a part of the processing pipeline.

The reasons for error can be divided into the following categories:

- diffuse patterns: some patterns (generated mostly by Canon machines) have an irregular appearance that shows a clearly distinguishable VPS distance but a much less clearer HPS distance. For these patterns, repetitions in horizontal direction are slightly shifted from one repetition to the next. An exact horizontal repetition was not detectable. But due to the error

[dpi]	HPS corr.	VPS corr.	HPS & VPS corr.
600	87.5%	89.1%	82.8%
400	82.8%	78.1%	75.0%
300	80.5%	73.4%	72.6%
200	71.1%	65.6%	63.3%

 Table 2
 Accuracies for the CPS code classification on the original (skewed) images for different resolutions. It can be seen that for 600 dpi good accuracies can be obtained, especially for the VPS distance measurement.

[dpi]	HPS corr.	VPS corr.	HPS & VPS corr.
600	90.6%	93.0%	88.3%
400	85.9%	78.9%	77.3%
300	82.8%	75.0%	75.0%
200	74.2%	67.2%	66.4%

Table 3 Accuracies for the CPS code classification on the deskewed images for different resolutions. It can be seen that for 600 dpi good accuracies can be obtained, especially for the VPS distance measurement.

margin given for the matching, it will find a HPS distance that is actually not the correct one when sticking to the definition. This is mostly observed for the 0.16 and 0.32 inch patterns. An example can be found in Figure 12.

- bad print quality: a few documents present printing defects that can be observed when using an old drum or when the paper path of the printer is dirty: this leads to toner spreading all over the page, resulting in many small dots everywhere that cannot be easily distinguished from the CPS dots.
- sparse patterns: on several print-outs the pattern only appears around printed areas. Thus the large white areas where the dots are easily identifiable do not contain any dots. This significantly reduces the number of dots for reliable matching, leading to mis-detection.

In only few cases, neither the VPS nor the HPS distance could be correctly extracted. In all other cases at least one of the distances was correctly computed, while the other distance was returned either as "no dots present" or as a distance that does not fit any of the classes and that can thus be easily intercepted.

The problem of diffuse patterns could be solved by accurately deskewing the scanned image before processing and reducing the allowed variation during horizontal matching of the search pattern. This would lead to a more accurate estimation of the HPS distance. Another approach would imply verifying if the diffuse patterns appear only in combination with the 0.16 and 0.32 inch VPS distances. In these cases, the HPS distance information could be discarded for printer class detection.

Sparse patterns represent the main problem: depending on the document content, e.g. in case of text, there may be enough dots left to detect a repetitive pattern. For some sample sets, the ground truth had even to be updated as the automatic method correctly found a pattern that had not been detected while manually generating the ground truth. In most cases, however, there



Fig. 12 Example of a diffuse pattern. It can be seen that the pattern is repeating in horizontal direction but that the repetitions have a small offset. A characteristic set of dots is colored in red to make the repetitions more easily detectable.

are only very few dots left which makes it hard for an automated method to detect a recurring pattern. During manual ground truth generation knowledge about the printer type often helped in steering the search for the HPS and VPS distances into the right direction. Unfortunately this information is not available in a real world scenario. Concerning the bad print quality, a more sophisticated document cleaning could help.

7.5 Results for CPS Code Comparison

The results for the evaluation of the accuracy of the CPS code comparison can be found in Table 4. For the test on the 600 dpi images 168 correct decisions and 16 wrong decisions were returned, thus an accuracy of 91.3%.

A detailed analysis of the errors in the case of the 600 dpi scans showed again that the main issue are *sparse patterns*: for these patterns the dots seem

[dpi]	Correct	Error	Accuracy [%]
600	168	16	91.3%
400	160	24	86.9%
300	151	33	82.1%
200	112	72	60.9%

Table 4 Accuracies for the CPS code comparison for different resolutions. It can be seen that for 600 dpi good accuracies can be obtained, whereas for 200 dpi results are only slightly better than guessing.



Fig. 13 Example of sparse patterns. It should be noticed, that for Sheet 0057-0004 only very few dots are present. For the and image 0057-0006 containing Sheet 6, more dots can be observed, especially in the text area.

only to be present around page content areas, so not in large white spaces. As only few dots are present and this mostly in areas where dots may be missing due to overlaid text, it is hard to compute a good prototype estimate. Examples of sparse patterns can be seen in Figure 13.

The problem of the sparse patterns is hard to solve: if only few dots are present, no stable prototype pattern can be generated. In this case the system could try to extract the prototype patterns and display these to an operator who could then decide if the document should be analyzed further. Only comparing the HPS and VPS distance could also be done, as the HPS and VPS distance computation is slightly less sensitive to sparse dot patterns.

A few errors are due to variations in the skew angle of the pages. This can be overcome by initially deskewing the page. As the image 0005 does not contain many text-lines, automatic and accurate deskewing as for the CPS code classification was not possible.

7.6 Results for Decoding the Xerox-type Pattern

Table 5 contains the list of the images that have been decoded together with the decoded information and the ground truth. It can be seen that from the 129 images that were decoded, 119 were correctly or at least consistently decoded. Sample set 0041 could not be decoded correctly due to sparse patterns. Discarding this sample set from the accuracy measurement, only two errors were made in the decoding, leading to an accuracy of 98.3% (119 out of 121 remaining patterns were correctly decoded).

One issue is related to the threshold defining when a dot is considered *on* or *off*. The default value of 0.5 showed to lead in rare cases to dots that are falsely set to *on*. This problem could be solved by more carefully choosing the threshold. One could also use a more sophisticated method to decide if a dot is on or off, e.g. using a classification based on the computed frequencies.

Another issue is related to the matching of the fully populated search pattern: missing dots in the image at the border of the base pattern may



Fig. 14 Examples of differing patterns from the same Xerox printer. It can be seen that there is a considerable overlap in both patterns but some parts differ. The red rectangle shows the region where the time and date information is encoded. This region shows differences between the two patterns. The green region shows the serial number information.

lead to wrongly matched search pattern, which in consequence will lead to erroneous frequency counts for the base pattern dots.

GT serial / type	sam.	img	serial	date	time
D5W0012766	0004	0001	127666	00.00.00	00.00
Epson C4000 Aculaser			127666	00.00.00	00.00
		0009	127666	00.00.00	00.00
VF6004222	0015	0001	42222	00.00.00	00.00
Xerox Phaser 790			42222	00.00.00	00.00
		0009	42222	00.00.00	00.00
207768	0018	0001	207768	18.10.05	10:54
Xerox DocuColor 1632		0002	207768	18.10.05	11:04
		0003	207768	18.10.05	11:17
		0004	207768	18.10.05	11:46
		0005	207768	18.10.05	12:03
		0006	207768	18.10.05	11:49
		0007	207768	18.10.05	12:08
		0008	207768	18.10.05	12:24
		0009	207768	18.10.05	12:15
FZ20000126	0021	0001	126	00.00.00	00.00
Epson C3000 Aculaser			126	00.00.00	00.00
		0009	126	00.00.00	00.00
CN-0PF019-73240-6CF-	0034	0001	1114002	32.00.00	64.00
4002					
Dell 3110CN			1114002	32.00.00	64.00
		0009	1114002	32.00.00	64.00
NA	0039	0002	29246	20.06.05	08:41
Xerox DocuColor 6060		0003	29246	20.06.05	08:41
		0004	29246	20.06.05	08:40
		0005	29246	20.06.05	08:40
		0006	29246	20.06.05	08:39
		0007	29246	20.06.05	08:39
		0008	29246	20.06.05	08:38
		0009	29246	20.06.05	08:37
NA	0040	0002	324928	24.06.05	18:44
Xerox DocuColor 2045			324928	24.06.05	18:44
		0009	324928	24.06.05	18:44
NA, Xerox DocuColor 12	0041	all	sparse	sparse	sparse
CN-0J6508-71971-49I-B055	0044	0001	1322	00.00.00	00.00
Dell 5100CN			1322	00.00.00	00.00
		0009	1322	00.00.00	00.00
F2PZ119714	0053	0001	119714	00.00.00	00.00
Epson C1100 Aculaser			119714	00.00.00	00.00
		0009	119714	00.00.00	00.00
NA	0072	0002	1071	25.06.05	10:38
Xerox DocuColor12			1071	25.06.05	10:38
		0009	1071	25.06.05	10:38
RLU000972	0108	0001	972	27.12.04	21:39
Xerox Workcentre M24		0002	972	27.12.04	21:32
		0003	972	27.12.04	21:33
		0004	05	80.00.04	64.55
		0005	972	27.12.04	05:36
		0006	972	27.12.04	21:36
		0007	972	27.12.04	21:37
		0008	972	27.12.04	21:38
		0009	972	27.12.04	21:38

					-
GT serial / type	sam.	\mathbf{img}	serial	date	time
3116491345	0109^{6}	0001	649134	29.07.05	12:32
Xerox Workcentre Pro		0002	649134	29.07.05	13:29
C2636					
		0003	649134	29.07.05	13:30
		0004	649134	29.07.05	13:29
		0005	649134	29.07.05	13:31
		0006	649134	29.07.05	13:00
		0007	649134	29.07.05	13:00
		0008	9034	29.05.05	13:00
		0009	649134	29.07.05	13:00
NA	0115	0001	19178	00.00.00	00.00
Dell Laser Printer 5100CN			19178	00.00.00	00.00
		0009	19178	00.00.00	00.00
NA	0115	0001	34880	00.00.00	00.00
Dell Laser Printer 5100CN			34880	00.00.00	00.00
		0009	34880	00.00.00	00.00

Table 5: Results of decoding the Xerox-type patterns. The three rightmost columns show the extracted serial number, date and time. It can be seen that not all Xerox-type pattern actually contain date and time information. Also, different printer manufacturers use the same pattern type but seem to encode different information, as can be seen on the Dell 3110CN.

8 Conclusion

In this paper we presented considerable extension of our previous work on counterfeit protection system code classification. The presented methods allow automatic comparison and decoding of counterfeit protection system codes. This can be effectively used for authenticating print-outs by identifying the source of the document. For evaluation purposes, the MIC dataset has been generated and made publicly available. It consists of sample sets of print-outs of the EFF sample documents is presented. The accuracy for pattern classification is shown to be up to 93%. For CPS comparison an accuracy of 91.3% is obtained. The decoding resulted in 98.3% of the cases in a meaningful result.

Despite the advances of this work, unanswered questions about the CPS codes remain: on the one hand it would be important to know the reason for the generation of the sparse patterns. On the other hand, evaluation would benefit if also the decoding of the patterns other than the Xerox type would be known. To achieve these goals, more community effort is needed.

⁶ The decoding method did not recognize the orientation of the pattern correctly due to a different scheme of the constant pattern parts. Therefore the orientation was manually corrected as the evaluation focuses on the correct detection of the base pattern and not on the correctness of the decoding scheme.

References

- G. N. Ali, P.-J. Chiang, A. K. Mikkilineni, J. P. Allebach, G. T. C. Chiu, and E. J. Delp. Intrinsic and extrinsic signatures for information hiding and secure printing with electrophotographic devices. In *Proc. of the Int. Conf. on Digital Printing Technologies*, pages 511–515, New Orleans, LA, USA, September 2003.
- I. Amidror. A new print-based security strategy for the protection of valuable documents and products using moire intensity profiles. In *Proc. of SPIE Optical Security and Counterfeit Deterrence Techniques IV*, volume 4677, pages 89–100, San Jose, CA, USA, January 2002.
- G. R. Ball, R. Stittmeyer, and S. N. Srihari. Writer verification in historical documents. In Proc. of SPIE Document Recognition and Retrieval XVII, volume 7543, pages 1–8, San Jose, CA, USA, January 2010.
- T. M. Breuel. A practical, globally optimal algorithm for geometric matching under uncertainty. *Electronic Notes in Theoretical Computer Science*, 46:1–15, 2001.
- 5. T. M. Breuel. Implementation techniques for geometric branch-and-bound matching methods. *Computer Vision and Image Understanding*, 90(3):258–294, 2003.
- T. M. Breuel. Binary morphology and related operations on run-length representations. In Proc. of the 3rd Int. Conf. on Computer Vision Theory and Applications, pages 159–166, Funchal, Madeira, Portugal, January 2008.
- S. Chen and S. Srihari. A new off-line signature verification method based on graph. In Proc. of the 18th Int. Conf. on Pattern Recognition, pages 869–872, Hong Kong, China, August 2006.
- J.-H. Choi, D.-H. Im, L. H.-Y., J.-T. Oh, J.-H. Ryu, and H.-K. Lee. Color laser printer identification by analyzing statistical features on discrete wavelet transform. In *Proc.* of the 16th Int. Conf. on Image Processing, pages 1505–1508, Cairo, Egypt, November 2009.
- 9. Electronic Frontier Foundation. Docucolor tracking dot decoding guide. http://w2.eff.org/Privacy/printers/docucolor/, accessed on 14.05.2010.
- 10. J. Hails. Criminal Evidence, page 150. Thomson Learning, 2004.
- N. A. Hampp, M. Neebe, T. Juchem, M. Wolperdinger, M. Geiger, and A. Schmuck. Multifunctional optical security features based on bacteriorhodopsin. In *Proc. of SPIE Optical Security and Counterfeit Deterrence Techniques V*, volume 5310, pages 117– 124, San Jose, CA, USA, January 2004.
- W. Jiang, A. T. S. Ho, H. Treharne, and Y. Q. Shi. A novel multi-size block benford's law scheme for printer identification. pages 643–652, Shanghai, China, September 2010.
- N. Khanna, A. K. Mikkilineni, G. T. C. Chiu, J. P. Allebach, and E. J. Delp. Survey of scanner and printer forensics at purdue university. In *Proc. of the 2nd Int. Workshop* on *Computational Forensics*, volume 5158 of *Lecture Notes in Computer Science*, pages 22–34, Washington, DC, USA, August 2008.
- 14. A. K. Mikkilineni, P.-J. Chiang, G. N. Ali, G. T. C. Chiu, J. P. Allebach, and E. J. Delp. Printer identification based on graylevel co-occurrence features for security and forensic applications. In *Proc. of SPIE Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 430–440, San Jose, CA, USA, February 2005.
- 15. MIT Media Lab. Initiative to stop the use of tracking dots. http://www.seeingyellow.com, accessed on 18.12.2009.
- N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Systems, Man, and Cybernetics*, 9(1):62–66, 1979.
- D. Pu, G. R. Ball, and S. N. Srihari. A machine learning approach to off-line signature verification using Bayesian inference. In Proc. of the 3rd Int. Workshop on Computational Forensics, volume 5718 of Lecture Notes in Computer Science, pages 125–136, The Hague, The Netherlands, August 2009.
- J. Sauvola and M. Pietikainen. Adaptive document image binarization. Pattern Recognition, 33(2):225–236, 2000.
- L. Schomaker and M. Bulacu. Automatic writer identification using connectedcomponent contours and edge-based features of uppercase western script. *IEEE Trans. Pattern Analysis Machine Intelligence*, 26(6):787–798, 2004.

- L. Schomaker, M. Bulacu, and K. Franke. Automatic writer identification using fragmented connected-component contours. In Proc. of the 9th Int. Workshop on Frontiers in Handwriting Recognition, pages 185–190, Tokyo, Japan, October 2004.
- M. Schreyer. Intelligent printing technique recognition and photocopy detection for forensic document examination. In *Proc. of Informatiktage 2009*, volume S-8, pages 39–42, Bonn, Germany, 2009.
- 22. C. Schulze, M. Schreyer, A. Stahl, and T. M. Breuel. Evaluation of graylevel-features for printing technique classification in high-throughput document management systems. In Proc. of the 2nd Int. Workshop on Computational Forensics, volume 5158 of Lecture Notes in Computer Science, pages 35–46, Washington, DC, USA, August 2008.
- C. Schulze, M. Schreyer, A. Stahl, and T. M. Breuel. Using DCT features for printing technique and copy detection. In *Proc. of the 5th Int. Conf. on Digital Forensics*, pages 95–106, Orlando, FL, USA, January 2009.
- 24. F. Shafait, D. Keysers, and T. M. Breuel. Efficient implementation of local adaptive thresholding techniques using integral images. In *Proc. of SPIE Document Recognition* and *Retrieval XV*, volume 6815, pages 681510–681510, San Jose, CA, USA, January 2008.
- C. E. Shannon. A mathematical theory of communication. Bell System Technical Journal, 27:379–423,623–656, 1948.
- P. J. Smith, P. O'Doherty, C. Luna, and S. McCarthy. Commercial anticounterfeit products using machine vision. In *Proc. of SPIE Optical Security and Counterfeit Deterrence Techniques V*, volume 5310, pages 237–243, San Jose, CA, USA, January 2004.
- J. S. Tweedy. Class characteristics of counterfeit protection system codes of color laser copiers. Journal of the American Society of Questioned Document Examiners, 4(2):53– 66, 2001.
- J. van Beusekom, M. Schreyer, and T. M. Breuel. Automatic counterfeit protection system code classification. In Proc. of SPIE Media Forensics and Security XII, San Jose, CA, USA, January 2010.
- J. van Beusekom, F. Shafait, and T. M. Breuel. Combined orientation and skew detection using geometric text-line modeling. Int. Jour. on Document Analysis and Recognition, 13(2):79–92, 2010.
- R. van Renesse. Ordering the order, a survey of optical document security features. In Proc. of SPIE Conference on Practical Holography IX, pages 268–275, San Jose, CA, USA, February 1995.
- R. van Renesse. Paper based document security a review. In European Conf. on Security and Detection, pages 75–80, London, UK, April 1997.
- R. van Renesse. Hidden and scambled images a review. In Proc. of SPIE Optical Security and Counterfeit Deterrence Techniques IV, volume 4677, pages 333–348, San Jose, CA, USA, January 2002.
- 33. R. van Renesse. Protection of high security documents developments in holography to secure the future market and serve the public. In *Proc. of Holo-Pack.Holo-Print*, Vienna, Austria, November 2006.