Automatic Counterfeit Protection System Code Classification

Joost van Beusekom[a, b], Marco Schreyer[a], Thomas M. Breuel[b]

[a]German Research Center for Artificial Intelligence (DFKI) GmbH
D-67663 Kaiserslautern, Germany
[b]Department of Computer Science, Technical University of Kaiserslautern
D-67663 Kaiserslautern, Germany
{joost.van_beusekom,marco.schreyer}@dfki.de, tmb@cs.uni-kl.de

ABSTRACT

Wide availability of cheap high-quality printing techniques make document forgery an easy task that can easily be done by most people using standard computer and printing hardware. To prevent the use of color laser printers or color copiers for counterfeiting e.g. money or other valuable documents, many of these machines print *Counterfeit Protection System* (CPS) codes on the page. These small yellow dots encode information about the specific printer and allow the questioned document examiner in cooperation with the manufacturers to track down the printer that was used to generate the document. However, the access to the methods to decode the tracking dots pattern is restricted. The exact decoding of a tracking pattern is often not necessary, as tracking the pattern down to the printer class may be enough. In this paper we present a method that detects what CPS pattern class was used in a given document. This can be used to specify the printer class that the document was printed on. Evaluation proved an accuracy of up to 91%.

1. INTRODUCTION

The advent of digital printing and imaging technologies had a tremendous impact on the way we generate, publish and store information nowadays. As with many new technologies, the opportunity to create printed documents in high quality has resulted in a more extensive usage of these novelties.

This technological progress is not only used for legitimate purposes but also for illegal activities. This especially holds since the quality of color laser copiers and printers increased in the end of the 1980's. Recent cases reported to the American Society of Questioned Document Examiners (ASQDE) revealed their increasing involvement in the production of counterfeited banknotes[1, 2] and forged documents.[3]

In an effort to stop the use of color laser copiers and printers as a counterfeiting tool, a counterfeit protection system code is printed on the documents. The code consists of a pattern of yellow dots that is nearly invisible for the unaided human eye. An example of such a pattern can found in Figure 1. The pattern is distributed on the entire surface of a document but may be overprinted by text or other document content.

Sending such document to a licensed laboratory allows the examiners to extract the so called *inspection code*. This is transmitted to the manufacturer of the printer. The manufacturer will then check his database to find the owner of that specific printer.

However, in public only little is known about this technology. Also, it is not clear who has access to the decoding technology and for what purposes it is being used. The Electronic Frontier Foundation attracted the wider public's attention to privacy problems related to these codes, also called tracking dots or yellow dots. Their work on detecting and decoding tracking dot patterns can be found on their web page.[4] An interesting set of slides[5] from a talk held at the *Chaos Communication Camp 2007* on this subject gives a good overview over the few publicly known details about the tracking dot patterns. Also, the *Computer Counter Culture Group* at the MIT Media Lab started a campaign to stop the use of CPS codes.[6]

If techniques of detecting and decoding these codes would be available, these could be used for legitimate applications, e.g. in large scale invoice processing, where many invoices from repeating sources occur, this method could be used to detect if one source suddenly changes the printer class. This might be a hint that one invoice could be forged. It can also be used as a first step to verify if the questioned document has possibly been printed on a given printer. If the CPS class is different, no further analysis is necessary. Finally, it could also be used as a tool, to allow people to check their print outs for the presence of these tracking dots.

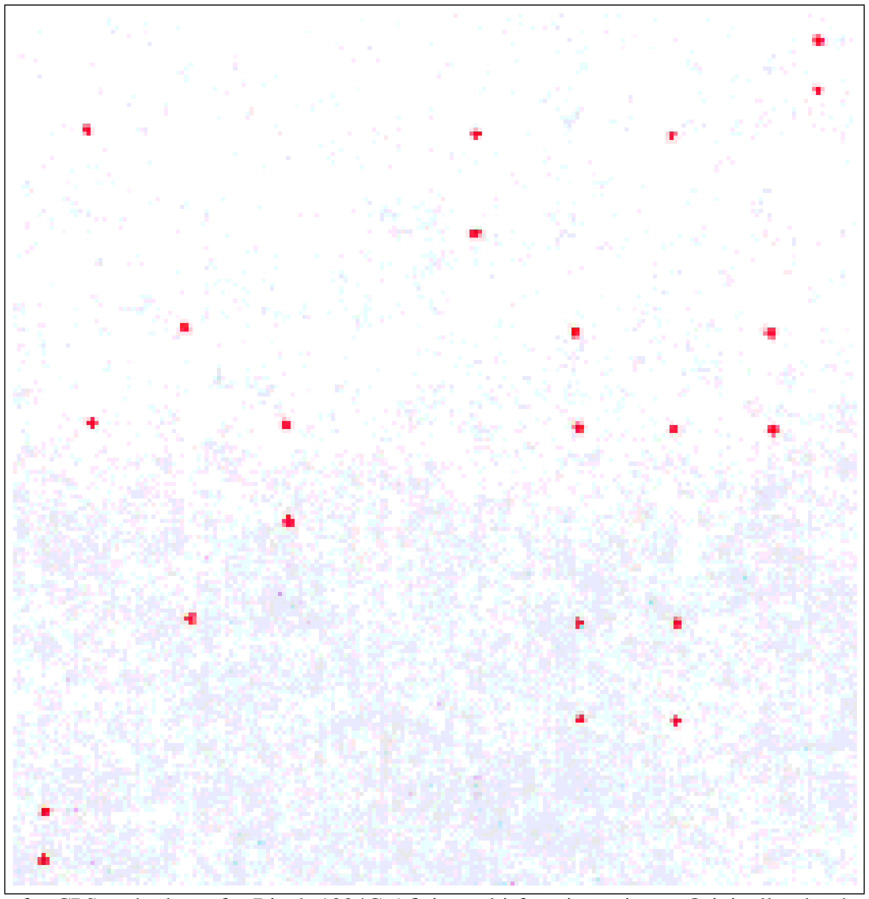
Figure 1. An example of a CPS code dots of a Ricoh 1224C Aficio multi function printer. Originally, the dots have been printed in yellow. The colors have been altered to make the dots visible.

In this work we present a automated method for detecting which printer class a document was printed on. The classification follows the methodology of Janis S. Tweedy[7] [*]. The vertical pattern separation (VPS) distances are used to distinguish 13 different classes of CPS codes. Each class can be assigned to several printer types or manufacturers.

The proposed method automatically processes the questioned document by extracting the tracking dots and estimating the VPS distance. Moreover, we propose to use 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 $90°$. Thus, we propose a method to extract the HPS and the VPS distances from a print out.

The remaining parts of this paper are organized as follows: Section 2 describes the proposed approach. Section 3 described the evaluation setup. In Section 4 results are presented. Finally, Section 5 concludes the paper.

2. DESCRIPTION OF THE APPROACH

The proposed methods first extracts the tracking dot candidate. These mainly consist of all connected components smaller than a given threshold that is fixed empirically. This is explained in Section 2.1.

Then, the HPS and VPS distances are estimated. This is done by matching a sub pattern, the so called *search pattern* on the extracted dots itself and to measure the translation from the search pattern to the matched patterns. These measurements result in a histogram showing peaks at positions that are multiples of the HPS or VPS distance respectively. The matching and the extraction of the HPS and VPS distances are explained in Section 2.2.

[*]The authors would like to thank Janis Tweedy for sending them a copy of the updated version of her work

2.1 Extraction of the tracking dots

The extraction of the tracking dots is done in several steps. The tracking dots are small yellow dots printed on the document. Their size is about 0.006inch (approximately 4px in a 600dpi scan). As the color of the dots is yellow, the first step in extracting the tracking dots is to extract the blue color channel from the RGB image. This image is considered from now on as a gray scale image.

Next, in order to separate background and foreground, the image has to be binarized. Several binarization algorithms exist.[8,9] They can be divided into two categories, global thresholding methods and local adaptive thresholding methods. As we assume scans in flatbed quality, a global thresholding method seems applicable. However, due to noisy print outs, a local adaptive method is more appropriate. A computationally fast implementation of Sauvola's local adaptive binarization was used.[10]

Problem occur if areas with dithered colors are present in the image. After binarization, these areas tend to show significant amounts of pixel noise, very similar to the ones form the CPS code. As these can be very numerous 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.[11] The width of the mask is fixed to half of the median distance between the neighboring connected components.

From the resulting image connected component are extracted. The filtering is done on a connected component basis: 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 showing the intermediate results for each step can be found in Figure 2. It should be noted that the resulting set of dots is not perfect, e.g. there may be false positives (due to noise or printer artefacts) or missed dots (due to overprinting or due to the filtering).

2.2 Horizontal and Vertical Pattern Separation Distance Computation

Now that the dots have been extracted, the next step consists of computing the vertical and horizontal pattern separation distances (VPS and HPS respectively). These distances are used for subsequent the classification scheme. For reading simplicity, in the following, only the case of the HPS distance is discussed. The VPS distance follows the analogous scheme, just switching the directions.

The approach we follow to compute the HPS is to take a random local subset of the tracking dots and match this subset to the remaining dots at the same y position. For each obtained match we get a translation value t_y in y direction. Statistics on these values can be used to estimate the size of a base pattern.

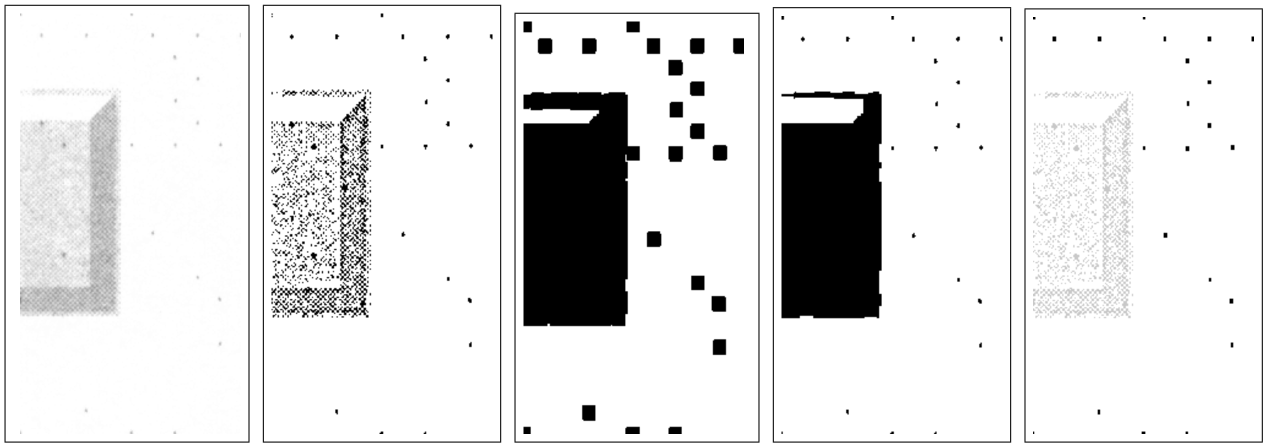
(a) (b) (c) (d) (e)

Figure 2. Visualization of the CPS dot extraction: (a) shows the extracted blue channel from the RGB image in gray scales. This is then binarized (see (b)). After dilation image (c) is obtained. Erosion of (c) leads to (d). Finally, using connected component based filtering the final image (e) is obtained: black pixels represent the extracted dots, the gray pixels represent the ignored parts.

The reason for only considering matches on the same y position lies in the fact, that not all CPS pattern repetitions are on a regular grid. Some CPS code repeat their base pattern shifted in x direction in neighboring columns. An example can be found in Figure 3. Without this restriction, the measurements would return distances that represent only a fraction of the real distance.

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 can 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 sub pattern (*search pattern*) is used for the matching: starting from a randomly selected point, an area around this point is defined such that in the direction of measurement its size is smaller than the smallest known pattern separation distance. 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 can be found in Figure 5.

2.2.1 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. Visually speaking, we try to find all repetitions of the horizontal search pattern on the right or left side of the search pattern. Similarly the vertical repetitions of the vertical search patterns are computed. This is done using the technique described in.[12] It uses an optimal branch-and-bound search algorithm, called RAST[12] (Recognition by Adaptive Subdivision of Transformation Space). This method allows robust and accurate finding of the positions of the repetitions of the search patterns.

The RAST algorithm performs a branch-and-bound search on the parameter space. For each direction, a separate search is done. The transformation parameter space can thus be reduced to parameter $t_x \in [-W, W]$ (in case of VPS: $t_y \in [-H, H]$) where W and H are the page width and height. To be more robust against small distortions of the page, t_x is set to allow for small variations, too. Examples of matching results can be found in Figure 5.

2.2.2 Estimating HPS and VPS

The idea we followed is to measure reoccurring translation values. If a pattern has a certain HPS, the translation values returned by the search are most likely to be multiples of the HPS distance. Using several iterations of the search, a histogram of translation values can be generated that shows characteristic peaks having a distance approximately equal to the HPS distance.

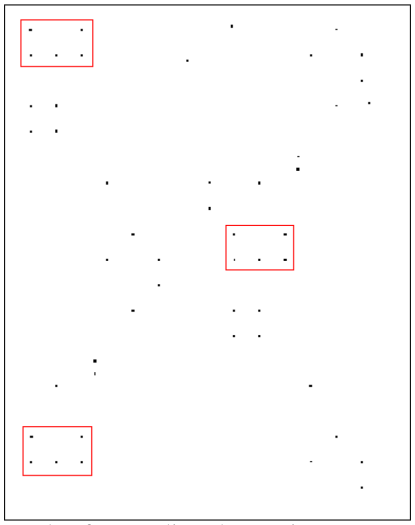
Figure 3. Example of a not aligned repeating pattern. A characteristic subset of the dot pattern is shown in the boxes. It can be seen that on the right side it is shifted in y direction in comparison to the pattern on the left side.

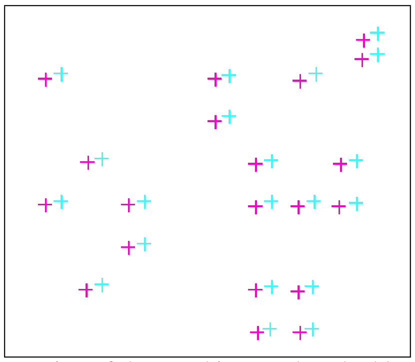
Figure 4. Illustration of the matching result. The blue (light gray) crosses represent the sub pattern to be searched for, the red (dark gray) one represent the repeating patterns above or below the subset pattern. The alignment has been slightly modified in order to be able to see the different crosses.

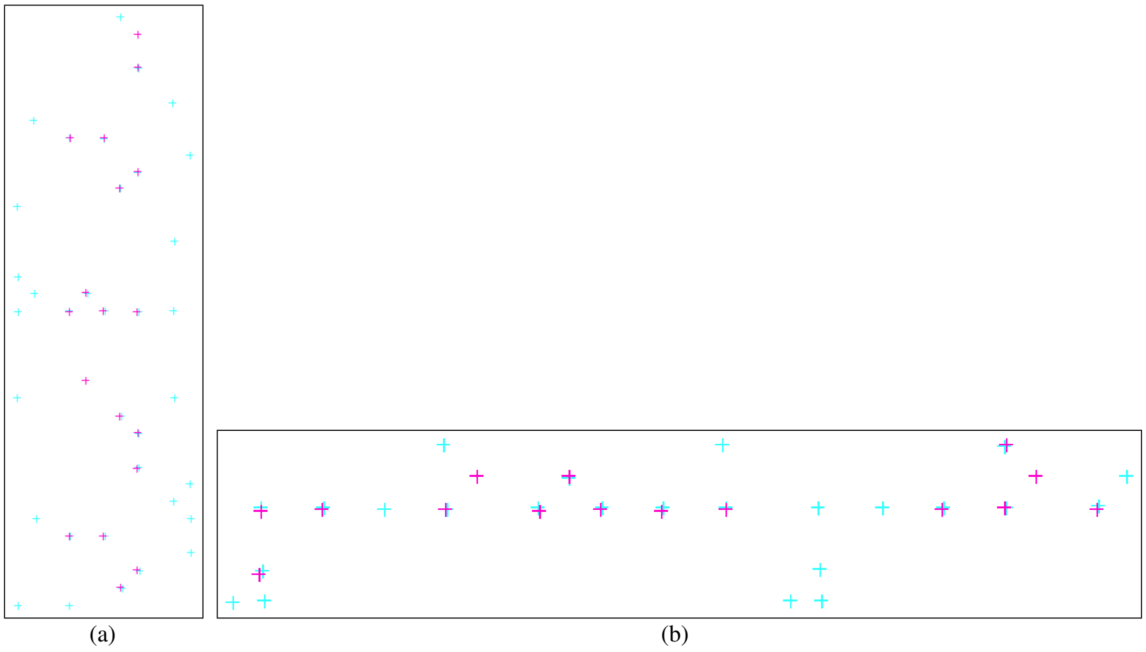

Figure 5. Examples of the horizontal and vertical search patterns. The red (dark gray) dots represent the sub pattern that is used to search for matches in horizontal (left image) and vertical (right image) direction. The blue dots (light gray) represent the dots where the sub pattern was matched to.

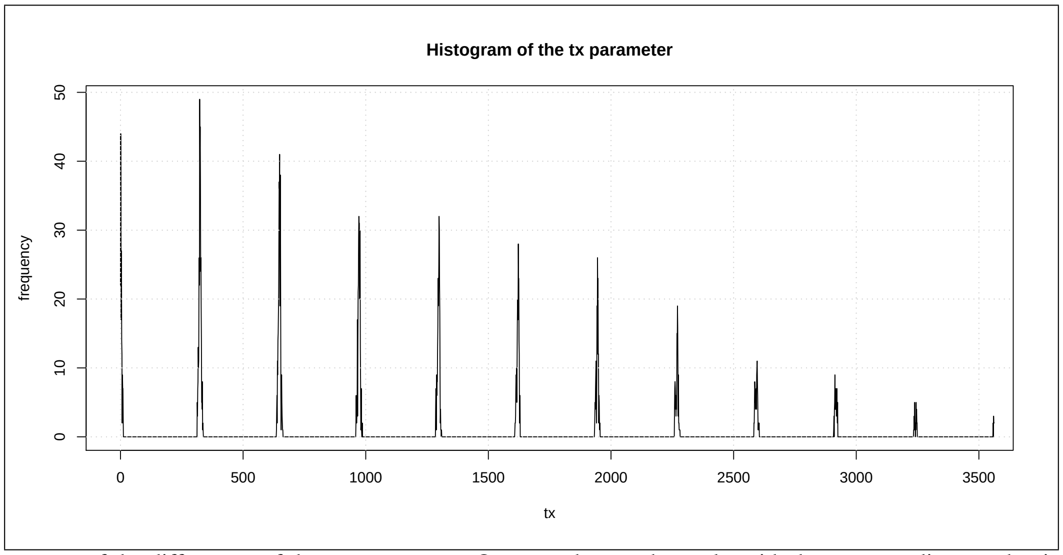

Figure 6. Histogram of the differences of the parameters t_x. One can observe the peaks with the constant distance that is equal to the HPS distance in pixel.

As the selection of the search pattern is a random process, it may happen that the selected search patterns do not represent any pattern but e.g. only a noise pattern. It may also happen that a search pattern is chosen that is repeated inside the base pattern (the main pattern that is printed multiple times on the page). 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 $R = \{(t_{x,1}, q_1) \ldots (t_{x,n}, q_n)\}$, where q_i is the quality of the match i best match returned by the RAST algorithm and n is the total number of results returned by all iterations. The quality q_i is the number of dots from the search pattern that could be matched to the remaining dots. A low quality thus means, that only a few dots from the search pattern could be matched. This quality is used to select the best m matches for generating the statistics, as matches with higher quality achieve a higher degree of robustness.

Using the m best results, a histogram of the translation values t_x is generated by computing all pairwise distances. An example of such a histogram can be found in Figure 6. Using this histogram, the HPS distance can be computed. This is done by histogram comparison: for all different possible HPS distances, a so called *ideal* histogram is generated. It is generated by distributing equally high peaks in the histograms at all different distances for HPS. Then all the *ideal* histograms are compared to the measured histogram using Jenson-Shannon-Divergence (JSD).[13] The ideal histogram with the smallest JSD gives us the HPS distance.

3. EVALUATION

Distinguishing different classes of printers and copiers on the basis of the different pattern types has been proposed by Tweedy.[7] Thirteen different classes of VPS distances have been defined.[7] For the newly defined HPS distances, the same classes have been used. The classification of the extracted HPS and VPS distance is done using fixed thresholds: if the computed HPS is in the range of ± 0.1inch around the ground truth HPS distance, it is considered as correct. The same holds for the VPS distance. Ten different classes are considered. In,[7] 13 classes were considered, as for two VPS classes different characteristics could be identified. As we only measure the HPS and VPS distances without distinguishing on the basis of the regularity of the pattern, we merged classes with identical VPS and different characteristics into one VPS class.

The following classes based on the VPS and HPS distances are used: no CPS pattern, 0.16, 0.32, 0.48, 0.50, 0.54, 0.64, 0.69, 0.96, 1.20 and 1.28inch distances. Using these classes and a database of which printer uses what pattern type can be used to identify the printer class a document was printed with.

Normally, if no tracking dots are present, no reasonable matches can be found. In this case, the document is considered to be CPS code free.

The documents we used to generate the dataset have been collected by the Electronic Frontier Foundation (EFF) to get an overview which printers actually generate tracking dots and which on do not. A first sample set of 68 sets of test print outs has been scanned in color using a resolution of 600dpi. Each set consists of 8 pages from the EFF printer test set sheets[†]. An example of such a sample set of 8 images can be found in Figure 7. On a set level, the ground truth has been generated manually. It 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 Janis Tweedy.[7]
- Miscellaneous information

Test document 6 (Figure 7(g)) of each sample contains the most realistic document type for our scenario, namely a page containing text at the top and some colored blocks at the bottom. As not all samples are complete. in total 67 images have been used.

[†]http://www.eff.org/wp/investigating-machine-identification-code-technology-color-laser-printers#testsheets

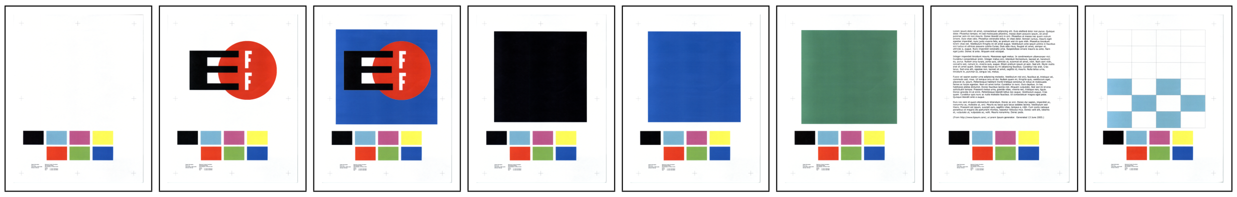
(a) Sheet 0 (b) Sheet 1 (c) Sheet 2 (d) Sheet 3 (e) Sheet 4 (f) Sheet 5 (g) Sheet 6 (h) Sheet 7

Figure 7. Test sheets from the Electronic Frontier Foundation.

4. RESULTS

The results for the test on the 67 images containing text and graphics are shown in Table 4. It can be seen, that the results for the VPS distance are slightly better than those for the HPS distance. An analysis of the errors showed the following problems:

- Diffuse patterns: some patterns 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 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 0.32inch patterns.
- Bad print quality: some documents present printing defects that can be observed when using an old drum: toner spreading all over the page. This leads to many small dots everywhere that cannot be distinguished from the CPS dots.
- Sparse pattern: some printers seem to print the pattern only around printed areas. Thus the large white areas where the dots are easily identifiable do not contain any of these. This significantly reduces the number of dots that can be used for reliable matching, leading to some misdetections.

Tot. num. of doc. images	**VPS & HPS correct**	**HPS correct**	**VPS correct**
67	57	58	61
Accuracy	85.0%	86.6%	91.0%

Table 1. Results on the 67 document images of type 6 from the EFF-DFKI dataset. The VPS distance is detected correctly in 91% of the documents.

5. CONCLUSION AND FUTURE WORK

In this work we presented a method for automatically extracting and classifying the counterfeit protection system codes for color laser printer and copiers. The classes used in[7] base on the vertical pattern separation (VPS) distance. This has been extended to also use the horizontal pattern separation (HPS) distance. First, the tracking dots are extracted. Using geometric matching of a search pattern, statistics on the translation values between the search pattern and the match can be computed. From these, the HPS and VPS distances can be extracted.

REFERENCES

[1] Chim, J. L. C., Li, C.-K., Poon, N. L., and Leung, S.-C., "Examination of counterfeit banknotes printed by all-in-one color inkjet printers," *Journal of the American Society of Questioned Document Examiners* **7**(2), 69–75 (2004).

[2] Li, C. K. and Leung, S. C., "The identification of color photocopiers: A case study," *Journal of the American Society of Questioned Document Examiners* **1**(1), 8–11 (1998).

[3] Parker, J. L., "An instance of inkjet printer identification," *Journal of the American Society of Questioned Document Examiners* **5**(1), 5–10 (2002).

[4] Electronic Frontier Foundation, "Is your printer spying on you?." http://www.eff.org/issues/printers, accessed on 18.12.2009.

[5] Schoen, S., “Decoding identifying printer information.” http://events.ccc.de/camp/2007/Fahrplan/events/1976.en.html, accessed on 18.12.2009.

[6] MIT Media Lab, “Initiative to stop the use of tracking dots.” http://www.seeingyellow.com, accessed on 18.12.2009.

[7] Tweedy, J. S., “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).

[8] Otsu, N., “A threshold selection method from gray-level histograms,” *IEEE Trans. Systems, Man, and Cybernetics* **9**(1), 62–66 (1979).

[9] Sauvola, J. and Pietikainen, M., “Adaptive document image binarization,” *Pattern Recognition* **33**(2), 225–236 (2000).

[10] Shafait, F., Keysers, D., and Breuel, T. M., “Efficient implementation of local adaptive thresholding techniques using integral images,” in [*Proc. of SPIE Document Recognition and Retrieval XV*], **6815**, 681510–681510 (January 2008).

[11] Breuel, T. M., “Binary morphology and related operations on run-length representations,” in [*Proc. of the 3rd Int. Conf. on Computer Vision Theory and Applications*], 159–166 (Januaray 2008).

[12] Breuel, T. M., “A practical, globally optimal algorithm for geometric matching under uncertainty,” *Electronic Notes in Theoretical Computer Science* **46**, 1–15 (2001).

[13] Shannon, C. E., “A mathematical theory of communication,” *Bell System Technical Journal* **27**, 379–423,623–656 (1948).