Blockwise Based Detection of Local Defects*

Xiaoyu Xiang[a] , Renee Jessome[b] , Eric Maggard[b] , Yousun Bang[c] , Minki Cho[c] , Jan Allebach[a]
[a] *School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA*
[b] *HP Inc. Boise, ID 83714, USA*
[c] *HPPK (HP Printing Korea), Suwon City, Gyeonggi, 16677, Republic of Korea*

Abstract

Print quality is an important criterion for a printer's performance. The detection, classification, and assessment of printing defects can reflect the printer's working status and help to locate mechanical problems inside. To handle all these questions, an efficient algorithm is needed to replace the traditionally visual checking method. In this paper, we focus on pages with local defects including gray spots and solid spots. We propose a coarse-to-fine method to detect local defects in a block-wise manner, and aggregate the blockwise attributes to generate the feature vector of the whole test page for a further ranking task. In the detection part, we first select candidate regions by thresholding a single feature. Then more detailed features of candidate blocks are calculated and sent to a decision tree that is previously trained on our training dataset. The final result is given by the decision tree model to control the false alarm rate while maintaining the required miss rate. Our algorithm is proved to be effective in detecting and classifying local defects compared with previous methods.

Introduction

Laser electrophotographic (EP) printers have been widely used in past decades. As one of the most important criteria in evaluating the performance of a printer, print quality is not only of concern to customers, but also designers of printers. For the reason that print quality can reflect the current working status and reveal hidden mechanical problems inside of a printer, the assessment of print quality has continued to be an important topic in printer-related research.

The traditional way of print quality diagnosis relies on manual examination of a printed page, which is specially designed for the testing purpose. The assessment work that is usually conducted by well-trained experts, includes marking exact areas with local defects and rating the overall page. Each test page can be rated as "A" "B" "C" or "D" four ranks, in which "A" and "B" mean the page passes the print quality assessment, while "C" and "D" mean the page fails the assessment. However, given the large number of pages to be evaluated, manually examining all pages is too costly and time-consuming. To solve this problem, a print defect detection algorithm is highly desired for building a smart print quality diagnosis system.

The local defect is one of the print defects of most critical concern. Typical local defects include gray spots and solid spots. The gray spot (also called carrier spot) is a phenomenon of low density around the agglomerates, which usually happens when the toner transfer from the Organic Photoconductor (OPC) to the Intermediate Transfer Belt (ITB) is blocked by some developed car-

riers or toner agglomerates on the OPC. Thus, a poor transfer of toners occurs around the agglomerates. An obvious visual feature of gray spots is that their color is lighter than that of the surrounding content (as shown in Figure 1).

The solid spot is another type of local defect. Different from gray spots, the color of solid spots is usually darker than nearby contents (as shown in Figure 1)). This phenomenon of high density around is due to the agglomerates of toner or carriers. The solid spot is a defect that occurs during the retransfer process, and is often observed in halftone patterns. Generally, the cause of solid spots is that toner retransfer from the ITB to the OPC is blocked by some developed carriers or toner agglomerates on the ITB. When the after-image transferred on the ITB moves to the next Pod (T1 or T2), an air gap occurs by the carrier on non-image area or contamination. Since the retransfer quantity is lower in the air gap area, the area appears as a solid spot.
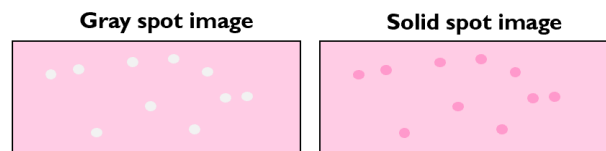

Figure 1. *Comparison of simulated gray spots and solid spots.*

There are some previous works on automatic detection of print defects. Jing et al. [2] borrowed a metric from the image quality area to print defect assessment. Ju et al. [3], Yan et al. [13] and Xiao et al. [12] proposed new algorithms to predict the visibility of fading defects. Zhang et al. [14][15] modeled periodic and aperiodic bands, and applied a histogram-based method to detect them. Nguyen et al. [5][6][7] designed a complete framework for print defect prediction based on defined intra-block and inter-block features for local and global characteristics, respectively. For local defects, Wang et al. [11] developed an algorithm to detect them and predict overall print quality with a trained support vector machine (SVM). In previous papers, we have a limited understanding of the cause, type, and severity of local defects. Different from streaks [9] or banding [10], we still lack a model describing common local defects.

In this paper, we develop a blockwise algorithm to detect and characterize local defects. This method involves a coarse-to-fine strategy in defect areas detection: first select possible regions by simple thresholding, and then apply a decision tree to exclude false alarms. In addition, our algorithm can classify different local defects according to their perceptual attributes including size, brightness, and other aspects.

Methodology

The overall workflow of our method is shown in Figure 2. The detection of local defects can be roughly divided into two stages: finding candidate areas, and verify the defect features inside each candidate block to give the final results. The two-stage detection pipeline can greatly reduce the runtime while ensuring the required miss rate.
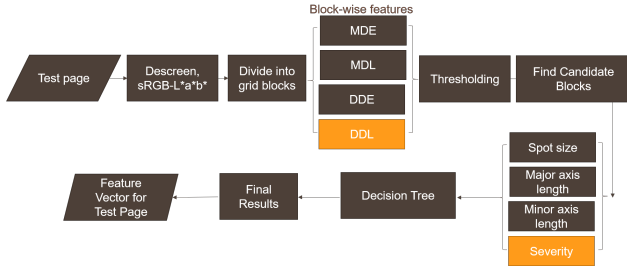

Figure 2. The pipeline of our method.

Preprocessing Test Page
Masking

The test pages are letter-size pages with at least one constant-tint area that is printed with one solid color (cyan, magenta, red, etc). These test pages are scanned at 600 dpi and stored in Portable Network Graphs (PNG format) that includes an alpha channel as a mask, which is a binary channel where 0 (black) stands for the masked part while 1 (white) is for the content. The mask channel is used to tell our algorithm which part of the test page should be processed. Because besides those constant-tone areas that our defect detection algorithm focuses on, each test page also has some other contents such as fiducial marks, and a barcode that records metadata about the master file and the original printer. Also, the unprinted areas, e.g., white edge, should also be excluded from our area of interest. Figure 3 shows an original image, and the masked image of one test page.
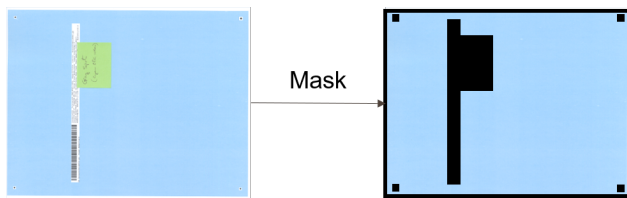

Figure 3. Test page sample (Original Page and masked page).

Descreening

Since all the pages are printed as halftones, high-resolution scanning would show the halftone patterns in our test pages. This might cause abnormal false alarms for local defect detection. In order to remove the halftone patterns without ruining the local defects, we apply a 12×12 Gaussian filter with a standard deviation of 2. Figure 4 shows a test area before and after descreening. It is clear to see that the processed region is smoothed with no visible halftone patterns while maintaining the gray spots in which we interested.

Color Space Conversion

After descreening the half-toned pages, we need to transfer the pixels from the RGB color space to the CIE L*a*b* color space, where $L*$ is the lightness and $a*$ and $b*$ are the green-red and blue-yellow color components, respectively. Compared with the RGB color space, L*a*b* is designed to be perceptually uniform according to human color vision [1]. Thus, the L*a*b* color space is widely used in color comparisons.

Blockwise Detection of Local Defects
Select Candidate Blocks

In order to detect local defects, we first divide the big region into relatively small blocks and detect local defects in each block. The advantage of blockwise detection is to lower memory consumption. We choose 75×75 as block size according to the scale of common local defects (see Figure 5), so that a block can be big enough contain a complete local defect, but not too big that contain multiple defects that might interrupt following computations.

Since the local defects are randomly located on our test page, it is very likely that a local defect falls on the boundary or even a vertex of the grid. If such a case happens, these local defects might be hard to detect. So it is necessary to search a second time for the missing defects. In the second detection, we move the grid by 35 pixels in both *x* and *y* directions from its initial location. Figure 6 shows the difference between the two grids. The local defects that cannot be detected in the first time would fall in the middle of the block. So we just need to combine the two detection results to get all local defects. However, there can be overlaps between two detection results. The connected components algorithm is applied to count the local defects accurately. The combined output is our initial estimation of areas with local defects, or the Region of Interest (ROI).

For each block, we use the metrics of graininess on the local scale that were first defined by Nguyen et al. [7]. These metrics have proved to be effective in the following works [5][11]. In general, Nguyen et al. quantized the intra-block fluctuation by computing the RMS (root mean square) difference from the mean for each pixel in a block.
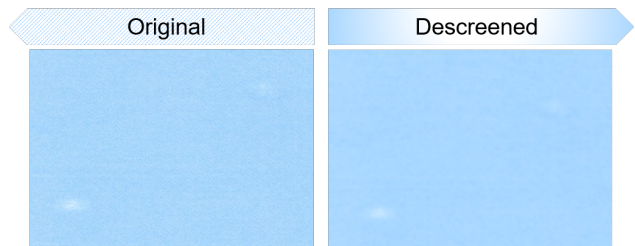

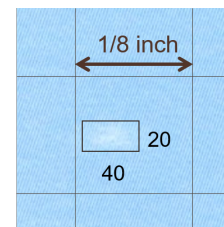**Figure 4.** Comparison of the original page and descreened page.

Figure 5. Choosing block size based on defect size.

In the pre-processing step, the input page is converted to L*a*b* color space. So for each block, we can compute the average L*, a*, and b* according to every pixel's L*a*b* value. Then we can compute the difference between the block average and a pixel i inside of block j:

$$\Delta E_{ij} = \sqrt{(L^*_{ij} - L^*_{block_j})^2 + (a^*_{ij} - a^*_{block_j})^2 + (b^*_{ij} - b^*_{block_j})^2} \quad (1)$$

where $L^*_{block_j}$, $a^*_{block_j}$, $b^*_{block_j}$ denote average values within the block j, and L^*_{ij}, a^*_{ij}, b^*_{ij} stand for the pixel values. After finishing the calculations all the pixels, the mean ΔE (MDE) for a block j can be computed:
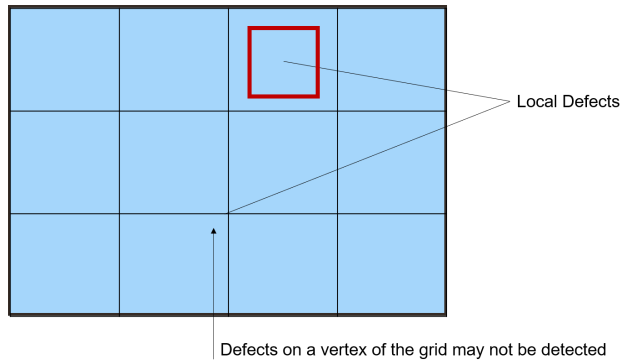
$$MDE_j = \frac{1}{75^2} \sum_{i=1}^{75^2} \Delta E_{ij} \quad (2)$$

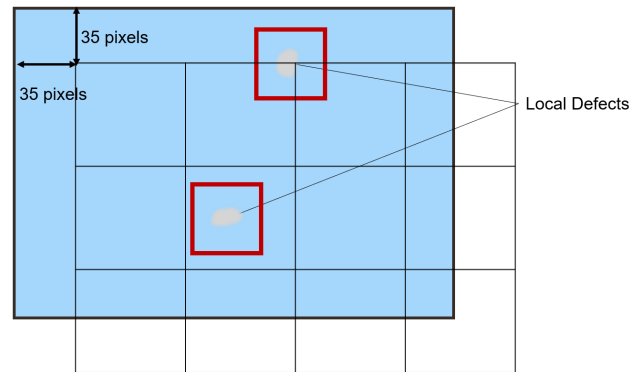The standard deviation of ΔE (DDE) is given by:

$$DDE_j = \sqrt{\frac{1}{75^2 - 1} \sum_{i=1}^{75^2} (\Delta E_{ij} - MDE_j)^2} \quad (3)$$

In a similar manner, we can define $L*$ related metrics ΔL, MDL, and DDL:

$$\Delta L^*_{ij} = |L^*_{ij} - L^*_{block_j}| \quad (4)$$

(a) Initial grid

(b) Move the grid by 35 pixels in both directions

Figure 6. *Move the grid to detect defects in all possible positions. After performing the detection twice, we combine the detected blocks as the Region of Interest (ROI)*
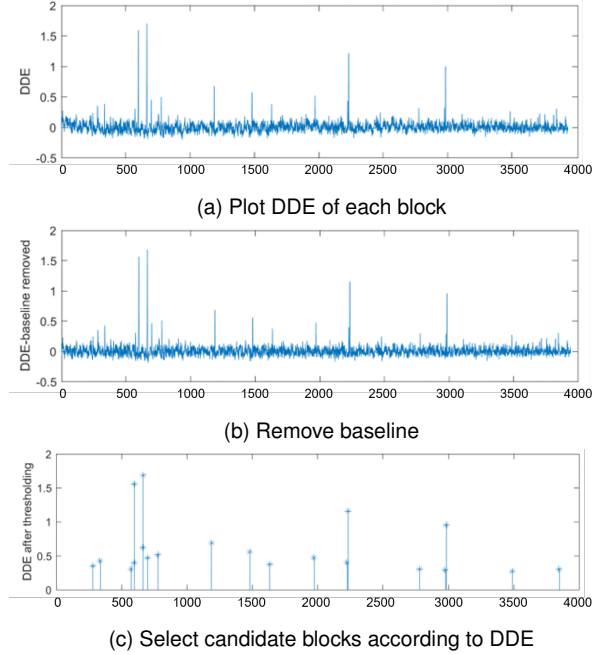

(a) Plot DDE of each block

(b) Remove baseline

(c) Select candidate blocks according to DDE

Figure 7. *Select candidate blocks according to their DDE*

$$MDL_j = \frac{1}{75^2} \sum_{i=1}^{75^2} \Delta L^*_{ij} \quad (5)$$

$$DDL_j = \sqrt{\frac{1}{75^2 - 1} \sum_{i=1}^{75^2} (\Delta L^*_{ij} - MDL_j)^2} \quad (6)$$

Repeat the calculations above until we go over all blocks. Higher *DDE* is related to a block with more fluctuations, thus it is more likely to have local defects, We take the *DDE* as the metric to get the candidate ROI. As shown in Figure 7, first we remove the baseline of DDE to make our algorithm less sensitive to local noise. Then we need to filter out the blocks with less fluctuation by thresholding. The peaks remaining in the last graph identify the blocks that comprise the ROI.

Features of Local Defects

In this section, we will introduce how to find the visible defects in the candidate blocks. Usually, local defects are small spots that are lighter or darker than the background. So we adopted valley-emphasis algorithm to mark the distinctive pixels in a candidate block. Ostu's algorithm [8] is a popular method, where the preferred threshold t is chosen automatically by maximizing the between-class variance:

$$t^* = \arg\max_{0 \le t \le L} [\omega_1(t)\mu_1(t)^2 + \omega^2(t)\mu_2(t)^2] \quad (7)$$

where t denotes the input (e.g. ΔE, L^*) value, L is the number of distinct gray levels, ω_1, ω_2 are percentages of pixels that belong to the background and defects, respectively, and μ_1 and μ_2 are the average values of background and defect pixels.

Ng et al. [4] proposed a new form of valley-emphasis algorithm based on Otsu's method by adding a new term in the maximization to emphasize the "valley" in the histogram. It is based on the assumption that the correct threshold should be located at the "valley" of histogram so that the foreground and background can be separated properly. Ng et al's modification can be expressed as:

$$t^* = \arg\max_{0 \leq t \leq L} [(1 - p(t))(\omega_1(t)\mu_1(t)^2 + \omega^2(t)\mu_2(t)^2)] \quad (8)$$

The additional term $p(t)$ is the percentage of pixels at level t. It serves as a weighting term, so that the lower the percentage is, the higher the weight is.

In Figure 8, we compare the results of applying the valley-emphasis algorithm followed by thresholding, to the descreened image expressed in either ΔE, L^* units. Figure 8 shows the results of different inputs after thresholding. The left-most image is the descreened input block, which includes a gray spot on the upper-left corner and some dispersed dark agglomerates. The result of L^* seems to be more focused on the gray spot region, while the ΔE result marks out both the gray spot the and dark agglomerates. For comparison, we also show the result that is directly acquired from the FWHM (Full width at half maximum) of ΔL, which is more close to the ΔE results. The reason for this difference lies in the operation to get the "Δ values", which is computing the absolute difference from average. In this manner, both dark and light regions strongly deviate from average, so that they are both marked out by the valley-emphasis algorithm. According to more comparisons on our test pages, the L^* inputs tend to give more gray spot results while fewer dark spots. Depending on our actual needs, we can choose either input to maximize accuracy.
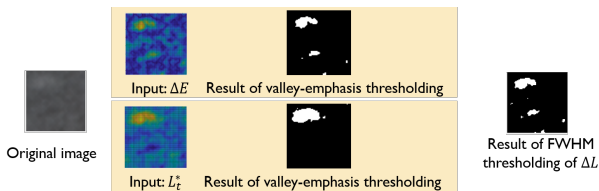

Figure 8. Results of Valley-emphasis Algorithm

After we get the mask of defects within a block, we can conduct the analysis for its attributes:

- Size (pixels): number of pixels in the defect area selected by valley-emphasis algorithm;
- Light / Dark: if the defect area is lighter or darker than the block average. This attribute can help us identify the type of local defect;
- Major and minor axis lengths: major and minor axis lengths of the equivalent eclipse of the defect area;
- Severity: the contrast of the defect area versus the background, defined by $\frac{\sum_{defect} \Delta E}{\sum_{background} \Delta E}$, which can also be regarded as the ratio of "defect volume" to the "background volume".

By inspecting each ROI according to the process shown in Figure 9, we can get the attributes above. Along with DDE, MDL, DDL, these features will be used in the following steps to exclude false positives (invisible defects). To better visualize the detected

visible defects, we draw a white bounding box around the block with gray spots, and a black box around the block with dark spots. We use the Connected Components algorithm to combine bounding boxes of adjacent blocks with the same type defect into one bigger box.
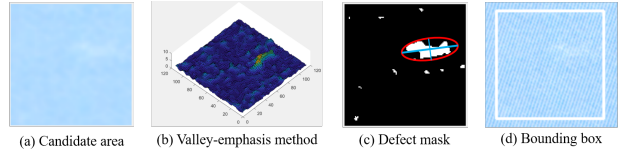

Figure 9. The processing of a candidate area in each step.

If we mark blocks with visible local defects as 1, and the rest as 0, we can plot their distribution versus each feature as shown in Figure 10. According to these plots, we can tell that *MDL*, *DDL*, and *DDE* are all good metrics for choosing candidate blocks in the initial step. The severity can exclude abnormal cases (e.g. infinite values). With defect size, we can exclude cases that are too small to see. Similarly, the major and minor axis lengths can help us exclude thin lines that are imperceptible and non-local. Although all these features obviously correlate with the existence of visible local defects, there is no single threshold that divides blocks with/without defects.
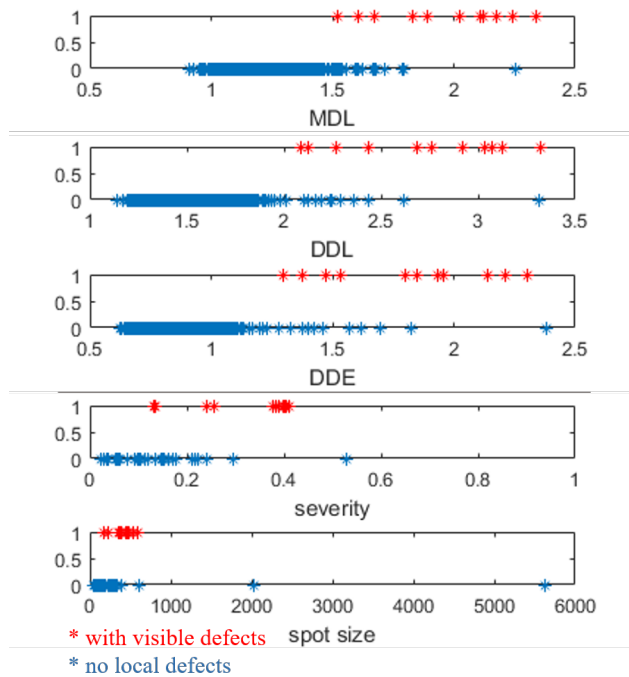

Figure 10. Distribution of each feature. The blocks with defects are marked as 1, and the other blocks as 0, so different blocks appear on the top and bottom of the graph, respectively.

Blockwise Dataset

We create a blockwise dataset that includes several types of local defects including gray spots, pinholes, etc. for further refinement models. This dataset is from 15 test pages with 66 uniform color regions including 12 colors. These test pages are in A4

size, scanned with 600 dpi. Taking out margins and barcode areas, 67,465 blocks are sampled from all test pages, among which 1,502 blocks are marked as blocks with local defects. After initial computation, 5,043 blocks are selected as ROIs by our algorithm.

Each block sample contains the following three types of metadata: 1) glocal features, including filename, block index, and color, which are related to the whole page; 2) blockwise features, including the block's x and y coordinate ranges, average L^*, a^*, b^* values, DDE, MDL, DDL, and the ground truth of local defects. These features can help us locate the block among the raw pages, and are only related to the block itself; 3) local defect features: light/dark, defect size, equivalent eclipse's major and minor axis length, and severity. Only the 5,043 blocks that passed initial selection have the third type of features.

Based on our blockwise dataset, we trained a decision tree model with 7 blockwise features as decision nodes: MDL, DDL, DDE, defect size, major axis length, minor axis length, and severity. The selection of which feature to use and the specific split is chosen using information entropy and gain. Since in our training set, the number of blocks with defects is smaller comparing to normal blocks, we train the decision tree model with a 2×2 cost matrix, where element $C(i, j)$ of this matrix is the cost of classifying an observation into class j if the true class is i. By changing the miss-classification cost, we can get models with different miss rates and false alarms that are calculated by:

$$MissRate = \frac{FN}{TP + FN} \qquad (9)$$

$$FalseAlarm = \frac{FP}{FP + TN} \qquad (10)$$

where FN denotes "False Negative", TP is "True Positive", FN is "False Negative", FP is "False Positive", and TN is "True Negative". The performance of our model is shown in Figure 11, where the best result of our model is obtained when $cost = 2$, the False Alarm rate is 0.088, while the Miss rate is 0.266.
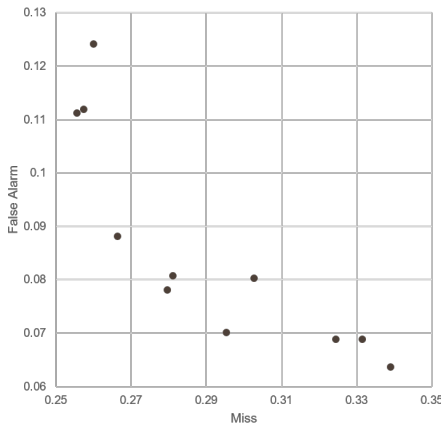

Figure 11. *ROC (Receiver operating characteristic) plot*

Results

Taking the final output blocks of the decision tree, we can get the final detection output as shown in Figure 12. By aggregating their information we can generate a feature vector for the whole test page. There are several features that we care about: number of all types of defects on the test page, number of each type of defects, local defects' average size, and maximal and minimal size, and standard deviation of the size, average severity, maximal and minimal severity. In addition, we also want to know the average location of all local defects to determine whether or not their distribution is random. The output feature vector for a test page is listed in Table 1.
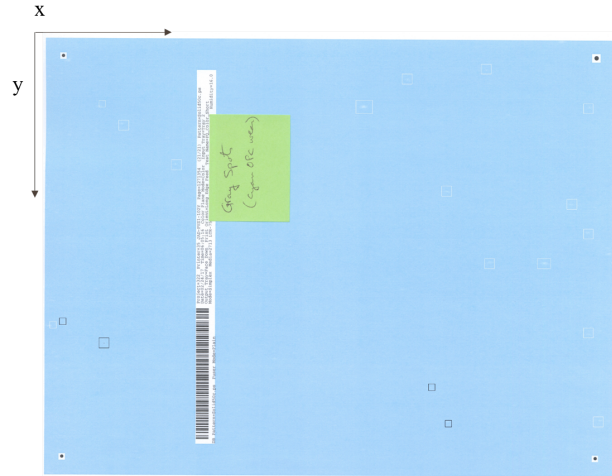

Figure 12. *Detection output of a test page. The boxes indicate the defects that have been identified (White boxes stand for gray spots, black boxes represents solid spots).*

Table 1: The feature vector for a test page

Number of defects on this page	19
Number of Gray Spots	15
Number of Solid Spots	4
Average size (mm^2)	0.53
Max size (mm^2)	1.24
Min size (mm^2)	0.08
The standard deviation of size (mm^2)	0.30
Average severity	0.16
Max severity	0.16
Min severity	0.16
Average y coordinate from the center of the page (mm)	-5.95
Average x coordinate from the center of the page (mm)	24.27

Our algorithm can not only be applied on print quality assessment, but also on detecting the scratches and contamination in the manufacturing of glass touchpads. Figure 13 shows that our method is robust to background noise that is generated from the matte surface and uneven lighting.

Conclusion

In this paper, we develop a coarse-to-fine method to automatically detect local defects, including the initial detection by thresholding, and the secondary refinement by a trained model.
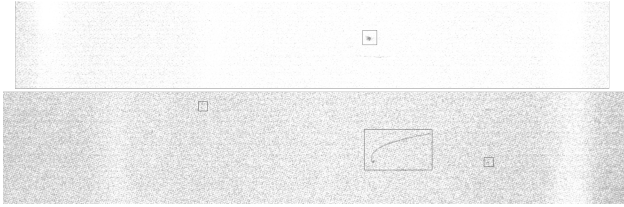
Figure 13. *The detection output of touchpad products. The boxes indicate the defects that have been identified.*

Different from previous works, we propose block-wise features to describe attributes of visible defects in the candidate area, which can help us determine the exact defect type. With these proposed features, we build a block-wise dataset of local defects for future training. A decision tree model is applied to produce more accurate results for visible local defects. Finally, we agglomerate block-wise results to generate a feature vector for each test page, which can be used for further assignment of page rank.

Acknowledgments

References

[1] Mark D Fairchild. *Color appearance models*. John Wiley & Sons, 2013.

[2] Xiaochen Jing, Steve Astling, Renee Jessome, Eric Maggard, Terry Nelson, Mark Shaw, and Jan P Allebach. A general approach for assessment of print quality. In *Image Quality and System Performance X*, volume 8653, page 86530L. International Society for Optics and Photonics, 2013.

[3] Yanling Ju, Eric Maggard, Renee Jessome, and Jan Allebach. Autonomous detection of text fade point with color laser printers. In *Image Quality and System Performance XII*, volume 9396, page 93960G. International Society for Optics and Photonics, 2015.

[4] Hui-Fuang Ng. Automatic thresholding for defect detection. *Pattern recognition letters*, 27(14):1644–1649, 2006.

[5] Minh Q Nguyen and Jan P Allebach. Controlling misses and false alarms in a machine learning framework for predicting uniformity of printed pages. In *Image Quality and System Performance XII*, volume 9396, page 93960I. International Society for Optics and Photonics, 2015.

[6] Minh Q Nguyen and Jan P Allebach. Feature ranking and selection used in a machine learning framework for predicting uniformity of printed pages. In *Image Quality and System Performance XIV*, number 12, pages 166–173. Society for Imaging Science and Technology, 2017.

[7] Minh Q Nguyen, Renee Jessome, Steve Astling, Eric Maggard, Terry Nelson, Mark Shaw, and Jan P Allebach. Perceptual metrics and visualization tools for evaluation of page uniformity. In *Image Quality and System Performance XI*, volume 9016, page 901608. International Society for Optics and Photonics, 2014.

[8] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[9] Zhang Runzhe, Maggard Eric, Jessome Renee, Bang Yousun, Choi Minki, and Allebach Jan. Detection of streaks on printed pages. In *Image Quality and System Performance XVI*. Society for Imaging Science and Technology, 2019.

[10] Huang Wan-Eih, Maggard Eric, Jessome Renee, Bang Yousun, Choi Minki, and Allebach Jan. Banding estimation for print quality. In *Image Quality and System Performance XVI*. Society for Imaging Science and Technology, 2019.

[11] Jianyu Wang, Terry Nelson, Renee Jessome, Steve Astling, Eric Maggard, Mark Shaw, and Jan P Allebach. Local defect detection and print quality assessment. In *Image Quality and System Performance XIII*, number 13, pages 1–7. Society for Imaging Science and Technology, 2016.

[12] Zuguang Xiao, Shaoyuan Xu, Eric Maggard, Katie Morse, Mark Shaw, and Jan Allebach. Detection of color fading in printed customer content. In *Color Imaging XXIII: Displaying, Processing, Hardcopy, and Applications*, number 16, pages 297–1–297–6. Society for Imaging Science and Technology, 2018.

[13] Ni Yan, Eric Maggard, Roberta Fothergill, Renee J Jessome, and Jan P Allebach. Autonomous detection of iso fade point with color laser printers. In *Image Quality and System Performance XII*, volume 9396, page 93960F. International Society for Optics and Photonics, 2015.

[14] Jia Zhang and Jan P Allebach. Estimation of repetitive interval of periodic bands in laser electrophotographic printer output. In *Image Quality and System Performance XII*, volume 9396, page 93960J. International Society for Optics and Photonics, 2015.

[15] Jia Zhang, Stephen Astling, Renee Jessome, Eric Maggard, Terry Nelson, Mark Shaw, and Jan P Allebach. Assessment of presence of isolated periodic and aperiodic bands in laser electrophotographic printer output. In *Image Quality and System Performance X*, volume 8653, page 86530N. International Society for Optics and Photonics, 2013.

Author Biography

Xiaoyu Xiang is a Ph.D. student at School of Electrical and Computer Engineering, Purdue University, supervised by Prof. Jan P. Allebach. She received her B.S. in Engineering Physics from Tsinghua University (2011), Beijing, China. Her research interests include deep learning, image quality assessment and enhancement, face alignment and recognition.

Renee Jessome received a B.S. in Mechanical Engineering from the Rochester Institute of Technology, Rochester, NY, in 1992. She has worked on a variety of printing devices over the past 14 years from the smallest LaserJet printers to Indigo Printing presses. She currently leads a Product Development team focused on high-end LaserJet Printers and MultiFunction Printers at HP Inc. in Boise, Idaho. During her early career at HP, she was developing storage devices such as disk drives, tape drives, and disk arrays.

Eric Maggard received his B.S. degree in Physics from Northwest Nazarene University, Nampa, Idaho in 1991 and the M.S. degree in Computer Science specializing in image analysis and processing from Walden University in 2006. He is an Expert Imaging Scientist in the LaserJet Hardware Division and has developed programs and image quality algorithms for the last 15 years that are used in the testing of LaserJet print and scan image quality. His interests include machine vision, object recognition, machine learning, robot control, and navigation.

Yousun Bang is a manager of Image Quality Part in Imaging Lab at HP Printing Korea Co. Ltd. She received her B.S. and M.S. in mathematics from Ewha Womans University, Seoul, Korea in 1994 and 1996, and

her Ph.D. in the school of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana in 2005. She worked for Samsung Advanced Institute of Technology and Samsung Electronics Company from 2004 to 2017. Her current research interests include image quality diagnosis and metrics and ML/DL based prediction for smart device services.

Minki Cho *is the engineer of HP Printing Korea. He received B.S.(1997) and M.S.(1999) in electrical engineering from Inha University, Korea. From 2003 to 2017, he worked for Samsung Electronics and Samsung Advanced Institute of Technology. His research areas are print image processing, print image quality diagnosis calibration. Recently, he is researching above interesting using ML DL.*

Jan P. Allebach *is Hewlett-Packard Distinguished Professor of Electrical and Computer Engineering at Purdue University. Allebach is a Fellow of the IEEE, the National Academy of Inventors, the Society for Imaging Science and Technology (IS&T), and SPIE. He was named Electronic Imaging Scientist of the Year by IS&T and SPIE and was named Honorary Member of IS&T, the highest award that IS&T bestows. He has received the IEEE Daniel E. Noble Award and is a member of the National Academy of Engineering. He currently serves as an IEEE Signal Processing Society Distinguished Lecturer (2016-2017).*