https://doi.org/10.2352/ISSN.2470-1173.2020.9.IQSP-314

A comprehensive system for analyzing the presence of print quality defects*

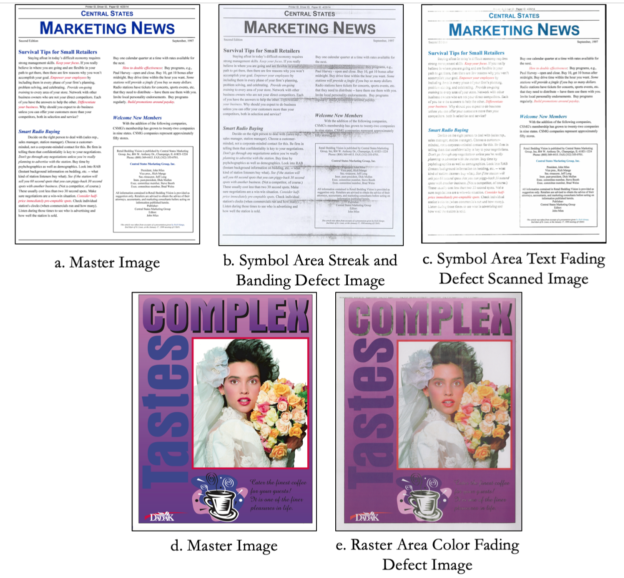
Runze Zhang [a], Yi Yang [a], Eric Maggard [b], Yousun Bang [c], Minki Cho [c],Jan Allebach [a]
[a] School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, U.S.A.
[b] HP Inc., Boise, ID 83714, U.S.A.
[c] HP Inc., Suwon City, KOREA.

Abstract

Print quality (PQ) is most important in the printing industry. It plays a role in users' satisfaction with their products. Page quality will be degraded when there are print quality defects on the printed page, which could be caused by the electrophotographic printer (EP) process and associated print mechanism. To identify the print quality issue, customers have to consult a printer user manual or contact customer service to describe the problems. In this paper, we propose a comprehensive system to analyze the printed page automatically and extract the important defect features to determine the type and severity of defects on the scanned page. This system incorporates many of our previous works. The input of this system is the master digital image and the scanned image of the printed page. The comprehensive system includes three modules: the region of interest (ROI) extraction module, the scanned image pre-processing module (image alignment and color calibration procedure), and the print defect analysis module (text fading detection, color fading detection, streak detection, and banding detection). This system analyzes the scanned images based on different ROIs, and each ROI will produce a printer defect feature vector. The final output is the whole feature vector including all the ROI feature vectors of the printed page, and this feature vector will be uploaded to customer service to analyze the printer defect.

1. Introduction

The electrophotographic (EP) process widely used in a modern laser printer is susceptible to various print defects. In the traditional method, the diagnosis of print defects requires the customer to print a professionally designed test page, and an expert to visually evaluate the printed page. This process is very costly and time-consuming [1]. On the other hand, there are many image quality analysis algorithms to detect print defects, such as streaks [2], banding [3], and color fading [4]. Each of these printer defect detection methods can only detect one printer defect. In this paper, we propose a comprehensive system that can automatically diagnose a customer's printer for many different printer defects without any human intervention. The input to this system is the digital master image and the scanned test image, shown in Figure 1, and the output of this system is the printer defect feature vector of the scanned test image. The printer defect feature vector will be uploaded to the service provider server to analyze the printer problem.



a. Master Image

b. Symbol Area Streak and Banding Defect Image

c. Symbol Area Text Fading Defect Scanned Image

d. Master Image

e. Raster Area Color Fading Defect Image

Figure 1: Four kinds print defects samples.

In this comprehensive system, we integrate many of our previous works, such as image alignment [5], an object map generation procedure [6], a region of interest (ROI) extraction method [7], and a text fading detection method [8]. We modify these image processing methods, and combine them in this image quality analysis system.

Our comprehensive image quality analysis system includes three main modules. In the first module, we use the master image to produce the object map based on the customer content and use the object map to extract four different kinds of ROI from the master image. The four kinds of ROI are symbol (text characters area), raster (image or graphics area), vector (uniform or gradient color area), and background (no color area). In the second module, we will process the scanned test image using the master image. This process includes image registration and color calibration. In the third module, we extract the defect feature vectors based on the different ROIs, because different ROIs include different printer defects. After the analysis process, each ROI will produce one feature vector. Finally, the system output is a feature vector that combines all the ROIs' feature vectors. We will cover the detail of this system in the following sections.

*Research supported by HP Inc., Boise, ID 83714

2. The comprehensive system procedure

In this section, we introduce the details of the comprehensive image quality analysis system. The overall pipeline of the proposed method is shown in Figure 2. It will include five parts: 1. Object Map Procedure; 2. ROI Extraction Method; 3. Image Alignment; 4. Color Calibration; 5. Feature Vector Extraction.

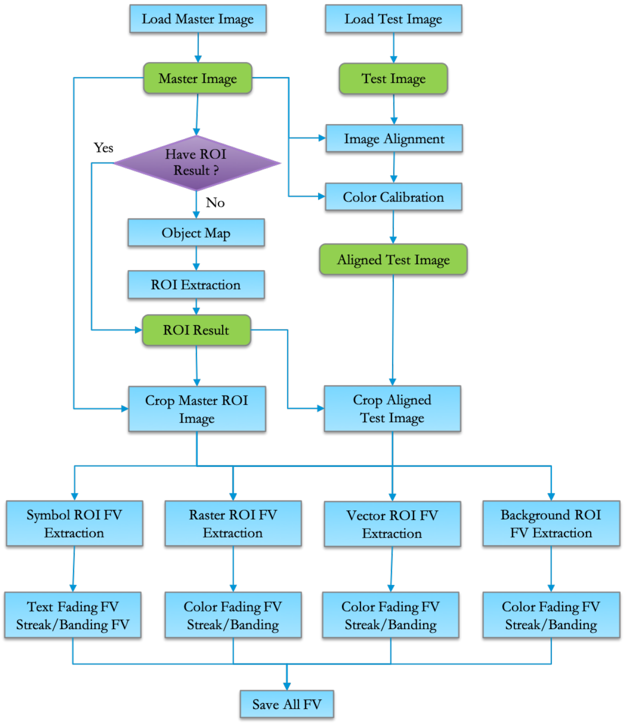
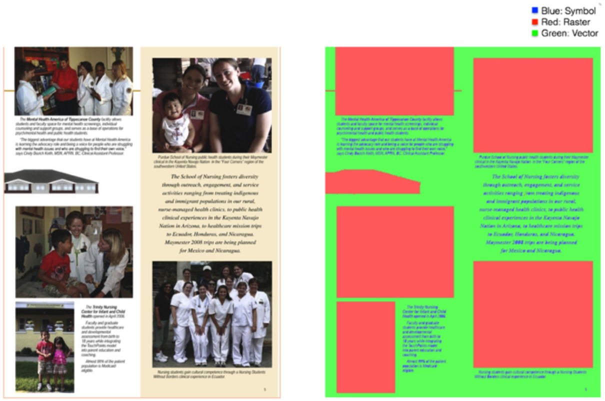


Figure 2: The overall pipeline of the comprehensive system for analyzing the presence of print quality defects.

As we introduced before, there are two input images for this comprehensive system: the master image and the corresponding scanned test image. First of all, we should check whether there are ROI extraction results of the master image stored in a file from a previous analysis of this page content. If there is no corresponding ROI extraction result, the system will process the master image to get the object map, and based on the object map produce the ROI extraction result. This ROI extraction procedure will label the rectangular area of four kinds of different ROI: symbol, raster, vector, and background. In the second step, the system will use the master image to do the image alignment and color calibration [8] for the scanned test image. Then, we can get the aligned test image. The system will use the master image ROI extraction result to crop the master image and the aligned test image to get the corresponding ROI images. There will be many corresponding ROI images, but they each belong to one of the symbol, raster, vector, and background types of ROI. Finally, the system will extract the feature vector for these four different kinds of ROI. Finally, we will combine all the feature vectors and save them. The printer can then send just this small feature vector to the service provider support center to get the print defect detection result.

2.1 Object map procedure

The first step of this analysis system produces an object map based on the master image. Note that we initially classify the background ROIs as vector objects. We will show how we separate vector ROIs and background ROIs from vector object in Sec. 2.2. The object map is an image; and each pixel of this image is labeled by the corresponding type, such as symbol, raster, vector. We show an object map sample in Figure 3. Figure 3 (a) is the input master image, and Figure 3 (b) is the output object map in which three types of object are represented by different colors: red for the raster objects, blue for the symbol objects, and green for the vector objects. The different objects in the input master image have different properties: the symbol object is small, and has sharp edges with a smooth interior (text characters); the vector object is large, and the color changes smoothly (figure or graph); the vector object is large, and the color changes very smoothly without a visible edge (uniform color area).



(a) Master image

(b) Object map

Figure 3: The input and output of the object map procedure.

In the object map procedure, we use two features: the size of an object and the edge sharpness of the object to classify the component, and we use the Sobel edge detection filter [9] to extract these two features. The Sobel edge detection filter contains two 3 × 3 mask windows, one to detect horizontal gradients and another to detect vertical gradients:

$$G_x^c = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * M^c \qquad G_y^c = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * M^c$$

The edge magnitude (EM) is defined as:

$$EM[i,j] = \frac{1}{3} \sum_{c=r,g,b} \sqrt{\left(G_x^c[i,j]\right)^2 + \left(G_y[i,j]\right)^2} \qquad (1)$$

In Equation 1, M is the master image; c is the superscript for the r, g, or b channel; G_x^c is the x direction filter processed master image c channel result; G_y^c is the y direction filter processed master image c channel result; the $*$ represents 2D convolution; and $[i,j]$ is the pixel position.

After getting the EM, we set two threshold values: Ts_edge (strong edge threshold) and Tw_edge (weak edge threshold), to separate all the pixels into three different parts and use them to produce three different maps: Strong Edge Map ($EM > Ts\_edge$), Non-Strong Edge Map ($Tw\_edge < EM < Ts\_edge$), and Non-Edge Map ($EM < Tw\_edge$). The Ts_edge threshold is higher than the Tw_edge threshold. We use the connected components algorithm [10] for these three object maps to get the symbol object edges, symbol object interiors, and vector objects. Figure 4 shows the pipeline of the object map procedure.

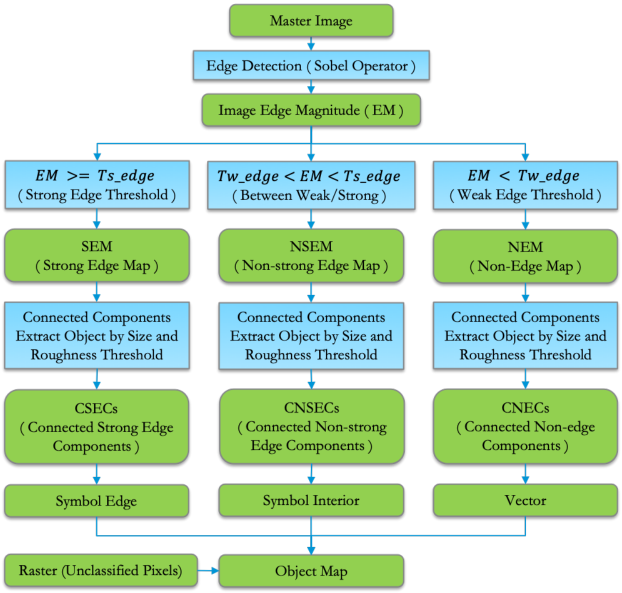
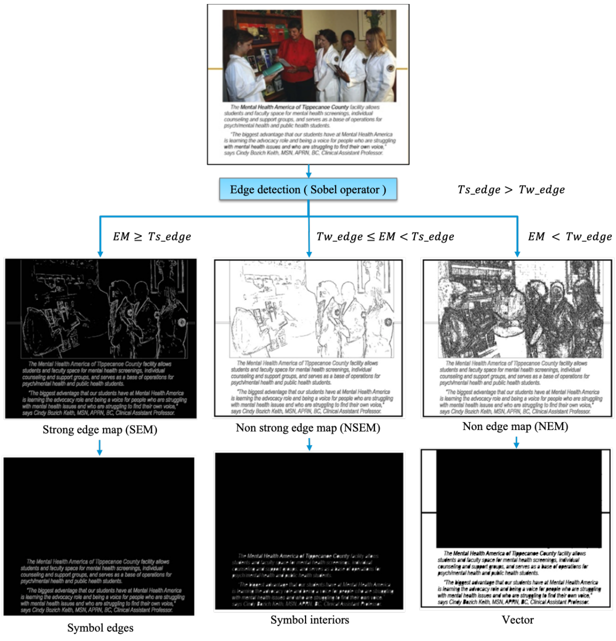


Figure 4: The pipeline of the object map procedure.



* In each binary image, only the white pixels which satisfy the threshold condition will be considered.

Figure 5: Sample images from the object map procedure.

In Figure 5, we show sample images illustrating how the object map procedure works. In the beginning, we use an RGB color image as an input master image and convert it from RGB color space to gray scale. Then, we can use the Sobel operator filter to process the gray image and get the edge magnitude (EM) result. In the second step, we use two thresholds to generate three binary images: SEM, NSEM, and NEM. The pixels we are interested in each binary image are the white pixels, which satisfy the threshold condition. If we observe these three images, we can find all the white pixels in the SEM are the edge pixels of symbol objects and raster objects. In the NSEM image, all the white pixels are the interior pixels of symbol, raster, and vector; but they are not connected. We can use the connected components algorithm to remove each big interior area and get the symbol object. In the NEM, the white pixels are also interior points. We can use it to extract the big uniform color areas. After we get the symbol objects and vector objects, the pixels that are un-labeled are raster objects. Finally, we can produce an object map.

2.2 ROI extraction procedure

In the ROI extraction module, we want to extract four types of rectangular ROIs based on the object map. There are only three different labels in the object map, so we will distinguish the background ROI and color vector ROI from the original vector object in the object map. The input to this ROI extraction module is the master image and object map, and the output is the symbol, raster, color vector, and background ROI, such as is shown in Figure 6.

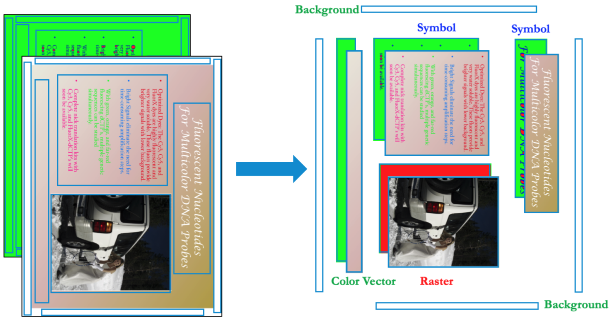


Figure 6: The input and output of the ROI extraction procedure.

In this ROI extraction procedure, we extract three different objects from the original object map at the beginning. Then, we use morphological operations and the connected components algorithm to extract the symbol and raster ROIs. For the vector object, we prefer to crop the ROI rectangular area to be as large as possible, because in a large rectangular area, it is easier to detect and analyze possible print defects. Based on this requirement, we design two methods to crop color vector and background ROIs. One method is to use the 2-D "Chessboard" distance method [12] to calculate the distance between every pixel in the interest area and its nearest edge in the area of interest. Then, we can choose and extract the deepest rectangular area. Another method is to calculate the maximum rectangular area of the area of interest. If we only extract the biggest and deepest rectangular ROI from the vector object map, there is a lot of vector object region left in the object map. To solve this problem, we use a greedy algorithm to get more ROIs for both of these two methods. In the greedy algorithm, we first extract the deepest or biggest ROI, and continue to extract the second deepest or biggest ROI, until the area of the most recently extracted ROI is less than a threshold. For the 300 dpi scanned image, we set the threshold area to be 600 $pixels$ × 600 $pixels$. Figure 7 shows the pipeline of the ROI extraction procedure.

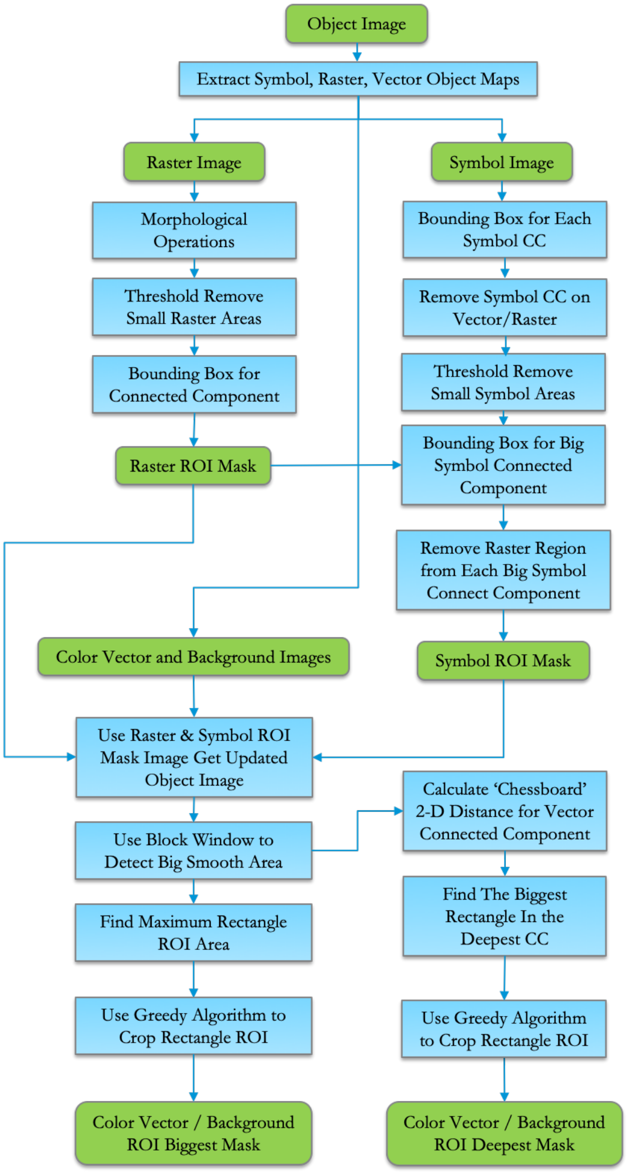
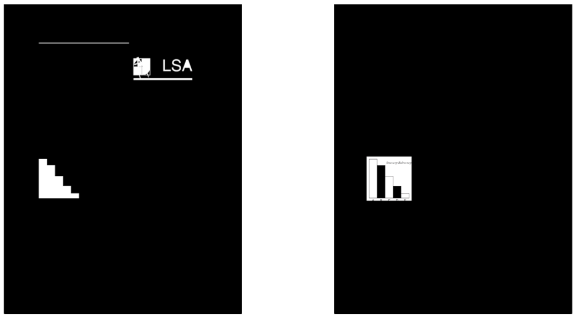

Object Image
Extract Symbol, Raster, Vector Object Maps
Raster Image
Morphological Operations
Threshold Remove Small Raster Areas
Bounding Box for Connected Component
Raster ROI Mask
Symbol Image
Bounding Box for Each Symbol CC
Remove Symbol CC on Vector/Raster
Threshold Remove Small Symbol Areas
Bounding Box for Big Symbol Connected Component
Remove Raster Region from Each Big Symbol Connect Component
Symbol ROI Mask
Color Vector and Background Images
Use Raster & Symbol ROI Mask Image Get Updated Object Image
Use Block Window to Detect Big Smooth Area
Find Maximum Rectangle ROI Area
Use Greedy Algorithm to Crop Rectangle ROI
Color Vector / Background ROI Biggest Mask
Calculate 'Chessboard' 2-D Distance for Vector Connected Component
Find The Biggest Rectangle In the Deepest CC
Use Greedy Algorithm to Crop Rectangle ROI
Color Vector / Background ROI Deepest Mask

Figure 7: The pipeline of the ROI extraction procedure.

2.2.1 Raster ROI extraction

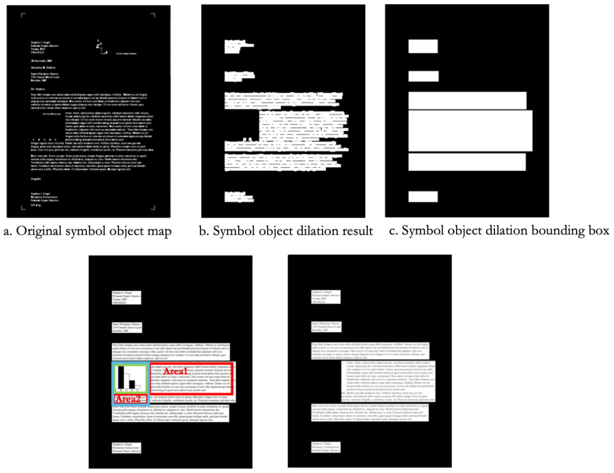
In the raster ROI extraction process, we do morphological operations [11] on the raster object map, as shown in Figure 8 (a), to remove the imperfections. Figure 8 (a) shows the raster area extracted from object map (the white color is the raster area). We do this by accounting for the form and structure of the image by using the average connected symbol component height as the morphological operation kernal size. We first do the dilation operation, and then continue to do the erosion operation. After the morphological operations, most of the raster area will connected together, and we can use the connected components algorithm to generate a new raster object map and use the bounding box [15] to label each raster connected component. If the raster area is too small, it is not useful to detect a defect. We set an area threshold to remove the small connected components. Finally, we can extract the final raster ROI. Figure 8 (b) shows the raster ROI blend image using raster ROI result as a mask for the input image.



a. Original raster object map
b. Raster ROI blend image

Figure 8: The raster object map and its ROI extraction result.

2.2.2 Symbol ROI extraction



a. Original symbol object map
b. Symbol object dilation result
c. Symbol object dilation bounding box
d. Remove raster area
e. Symbol ROI result blend image

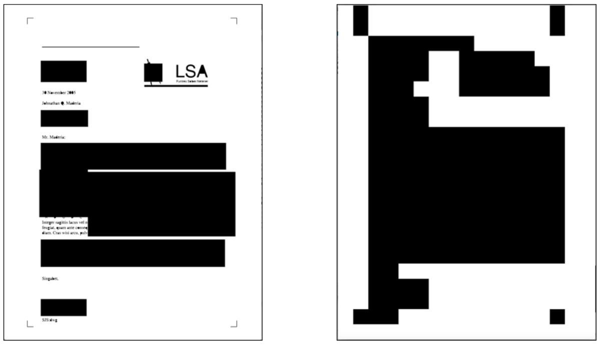
Figure 9: The symbol ROI extraction procedure.

In the symbol object map, most symbol areas are text characters. And they are not connected, as shown in Figure 9 (a). Firstly, we use the connected components algorithm for the symbol object map to label each independent text character. Based on the observed result, we use the average height of the text characters connected component as the size of the structuring element for the morphological operations to dilate the symbol object map, shown in Figure 9 (b). All the text characters in one paragraph will be connected. We use a bounding box for the new symbol object map connected component result to get the significant symbol ROI result, shown in Figure 9 (c). However, we find that it includes a raster object, which is what we do not want. In this case, we need to cut off the smallest rectangular area from the large symbol ROI shown in Figure 9 (c) to retain most of that ROI. In Figure 9 (d), we find that there are two ways to cut off the raster object: one is to cut off the red box Area 1 (horizontal cut); another is to cut off the red box Area 2 (vertical cut). In the general case, we want to cut off the small red area and to keep most of the symbol ROI. So in this condition, we will cut off the blue box. We then get the final symbol ROI result, shown in Figure 9 (e).

2.2.3 Color vector and background ROI extraction

We use the same two methods to extract the color vector and background ROIs. As most of the images include background areas, and not color vector areas, in this paper, we only show the background ROI extraction procedure.

1. Mask the symbol and raster ROI area in the background object map to produce a new background object map, shown in Figure 10 (a);
2. Use a 300 $pixels$ × 300 $pixels$ block window to label the big background map, shown in Figure 10 (b);


a. Updated background image

b. Background big smooth area image

Figure 10: The pre-process for background object map.

3. Use the 2-D "Chessboard" distance to extract the deepest rectangular area from background big smooth map;
4. Extract the maximum rectangular area of the background big smooth map [13]. In this step, we should calculate the accumulation matrix at the beginning. Then, we can extract the maximum rectangular area of the big background map;

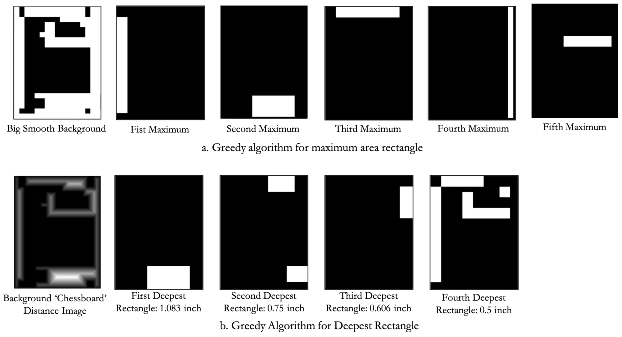
Algorithm 1 ACCUMULATION MATRIX

```
for $row\ i$ do
    for $col\ j$ do
        if $A[i,j] == 1$ then
            $B[i,j] = A[i,j] + B[i-1,j]$
        else if A[i,j]==0 then
            $B[i,j] = A[i,j]$
        end if
    end for
end for
```

5. Use the greedy algorithm to iterate the maximum area rectangular algorithm and the deepest rectangular method to find all useful background areas, shown in Figure 11.

Algorithm 2 MAXIMUM RECTANGULAR

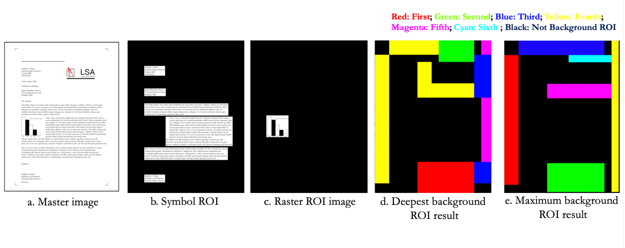
```
for $row\ i$ do
    for $col\ point\ j1$ do
        for $col\ point\ j2$ do
            Calculate the maximum area between $j1$ and $j2$
            if Current Calculation Area > Record Maximum Area then
                Update the record maximum area
            end if
        end for
    end for
end for
```


Big Smooth Background
Fist Maximum
Second Maximum
Third Maximum
Fourth Maximum
Fifth Maximum
Background 'Chessboard' Distance Image
First Deepest Rectangle: 1.083 inch
Second Deepest Rectangle: 0.75 inch
Third Deepest Rectangle: 0.606 inch
Fourth Deepest Rectangle: 0.5 inch
a. Greedy algorithm for maximum area rectangle

b. Greedy Algorithm for Deepest Rectangle

Figure 11: The greedy algorithm result.

The structure of the color vector object map is the same as the background object map, except that the color vector is color and the background is white. We can use the same method that we used to extract background ROIs to extract color vector ROIs. After we extract the background and color vector ROIs, we finish the ROI extraction process, as shown in Figure 12.


Red: First; Green: Second; Blue: Third; Yellow: Fourth; Magenta: Fifth; Cyan: Sixth ; Black: Not Background ROI
LSA
a. Master image

b. Symbol ROI

c. Raster ROI image

d. Deepest background ROI result

e. Maximum background ROI result

Figure 12: The ROI extraction result.

2.3 Image alignment of scanned test image

When the test image is printed from a printer and goes into a scan bar, it is very misaligned with the master image. An example shown in Figure 13, where Figure 13 (a) is the master image, Figure 13 (b) is the scanned image, Figure 13 (c) is the blend image of the master image and the scanned test image, and Figure 13 (d) shows the detail of the blend image between the master and test images. An image registration algorithm should correct this misalignment before we crop out the ROI from the master and test images.

The pipeline of the image registration algorithm is shown in Figure 14. Firstly, we should convert the scanned test image and the master images to grayscale value. Sec-

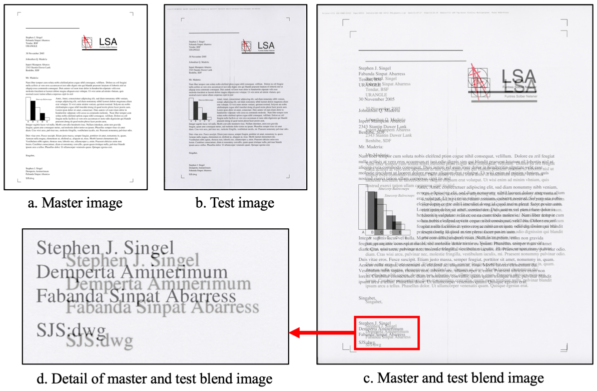


Figure 13: The blend of master and scanned test images.

ondly, we downsample both the test and master images to a lower resolution for faster computation. Thirdly, we should do a histogram matching for the scanned test image based on the master image, because there may be some intensity or color differences between the master and scanned test images. In our experiments, we found that histogram matching can significantly improve the image alignment result. Now, we can extract the interest points from the color balanced low-resolution grayscale images. In this step, we use the Harris Corner Detection method [16].

The next step is to extract the feature descriptors for each interest point. These feature descriptors are useful to find the corresponding interest points in the master image and the test image. We can establish correspondences by directly comparing the gray levels. We use an $(m+1)\times(m+1)$ window around the corner pixel in the master image with the gray levels, and use a similar window around the corresponding pixels in the scanned test image. The m value is not fixed, because it depends the downsampling rate. For the results presented in this paper, the downsampling rate is 3 and $m = 30$ (it's 240 pixels before downsampling in 300 dpi scanned image). We minimize the following function (Equation 2) the sum of squared differences (SSD) [17] to establish correspondences between the master and scanned test image. In the Equation 2, f is the feature descriptors for the interest point at position $[i, j]$ from the master or test image. The result is shown in Figure 15.

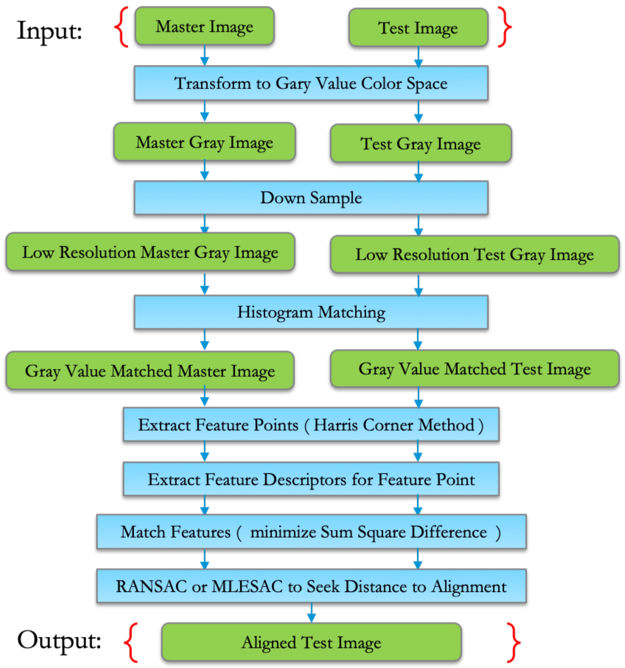

Input:
Master Image
Test Image
Transform to Gary Value Color Space
Master Gray Image
Test Gray Image
Down Sample
Low Resolution Master Gray Image
Low Resolution Test Gray Image
Histogram Matching
Gray Value Matched Master Image
Gray Value Matched Test Image
Extract Feature Points (Harris Corner Method)
Extract Feature Descriptors for Feature Point
Match Features (minimize Sum Square Difference)
RANSAC or MLESAC to Seek Distance to Alignment
Output:
Aligned Test Image

Figure 14: The pipeline of image alignment for the scanned test image.

$$SSD = \sum_i \sum_j \| f_{master}[i, j] - f_{test}[i, j] \|^2 \quad (2)$$

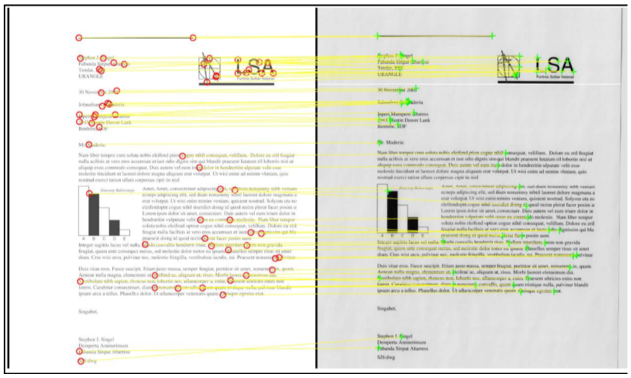


Figure 15: The corresponding interest points between the master image and the scanned test image.

After acquiring the corresponding set of interest points and their feature descriptors, we need to calculate the transformation matrix between the master image and the scanned test image. Here, we assume that the scanned test image is only misregistered with the master image by a small skew angle θ and a small translation along the x axis and the y axis. Three pairs of corresponding interest points are sufficient to solve this problem. But we usually have a lot of matched interest points. So we use the RANSAC (random sample consensus) algorithm [18] to calculate the best transformation matrix between the master image and the scanned test image. Finally, the transformation matrix needs to be adjusted by the factor of the downsampled rate before we use it for the scanned test image. The image alignment result is shown in Figure 16.

2.4 Color calibration of the aligned test image

Since there will be a difference in the colors between the digital master image and the scanned test image, we should do color calibration before we extract the defect feature vectors in the different ROIs. In the actual implementation of the image analysis process, we will use a different color calibration algorithm for each different printer/scanner pair. Here, to demonstrate the overall print quality defect diagnosis process, we will just use a simple color transformation. Firstly, we transform the scanned test image and master image from RGB color space to CIE $L^*a^*b^*$ color space. Secondly, we calculate the mean and standard deviation for these three channels. After we use the color transformation function (Equation 3) [21] for the scanned test image, we can get the color calibration result.

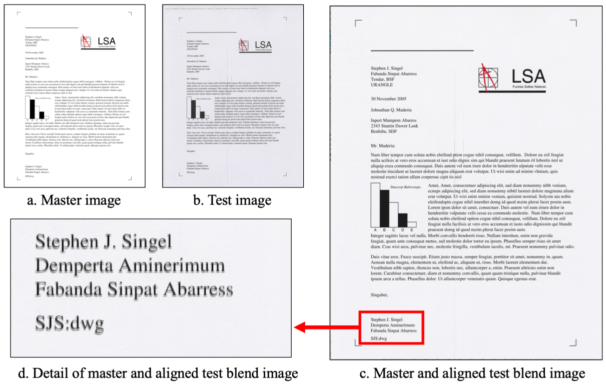
$$TC^c[i,j] = \left[\left(T^c[i,j] - T^c_{mean} \right) \cdot \frac{M^c_{stddev}}{T^c_{stddev}} \right] + M^c_{mean} \quad (3)$$

In Equation 3, TC is the target image after calibration; T is the scanned test input image; M is the master image that serves as the reference source for calibration; i, j is the pixel position in the image; c is the channel of L^*, a^*, or b^*; $mean$ is the average value of the channel; and $stddev$ is the standard deviation of the channel.

2.5 Symbol ROI defect feature vector extraction

In this section, we will describe the symbol ROI feature vector extraction process. Up to this point, we have done ROI extraction, image registration, and color calibration between the master image and the scanned test image. Now, we can extract feature vectors of printer defects from the different ROIs. As we introduced before, there are four kinds of ROIs. Let us analyze the symbol ROI first. There are three types of potential defects in the symbol ROI: streaks, banding, and text fading. Steaks are light or dark lines parallel to the printing process direction, as shown in Figure 17 (c). They occur when the ITB (Intermediate Transfer Belt), OPC (Organic Photo Conductor), or other color cartridge components have defects. Banding refers to light or dark lines orthogonal to the printing process direction, as shown in Figure 17 (c). Text fading usually results when one or more cartridges is low on toner. The text fading page has a noticeable reduction in the density of the text characters, as shown in Figure 17 (d). The critical issue of this section is to separate the text characters and background from the cropped symbol ROI. Figure 17 shows the pipeline of the extraction process for the symbol ROI defect feature vector. This pipeline illustrates the processing of a single symbol ROI. The same process would be separately applied to each symbol ROI on the page, as shown in Figure 17 (b). The details of this process will be discussed below.

Firstly, we should crop out the corresponding symbol ROIs from the master image and scanned test image based on the ROI extraction result. Figure 17 (b) shows the symbol ROI result, where only the white areas are the ROIs that we



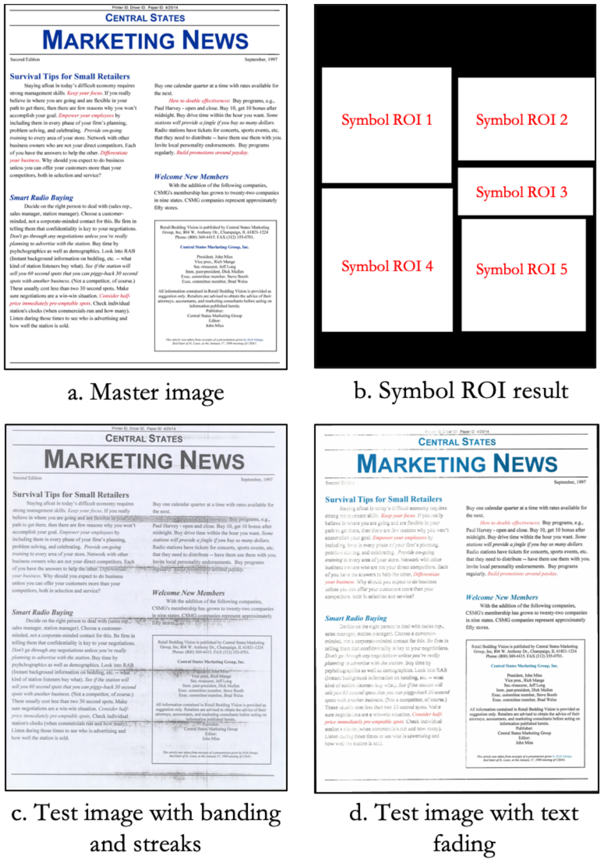
a. Master image

b. Test image

d. Detail of master and aligned test blend image

c. Master and aligned test blend image

Figure 16: The image alignment result.



a. Master image

b. Symbol ROI result

c. Test image with banding and streaks

d. Test image with text fading

Figure 17: Symbol ROI feature vector extraction input images. (a) shows the master image. (c) and (d) are the scanned images for two different prints made from this master image. The printer is a short-edge-first printer.

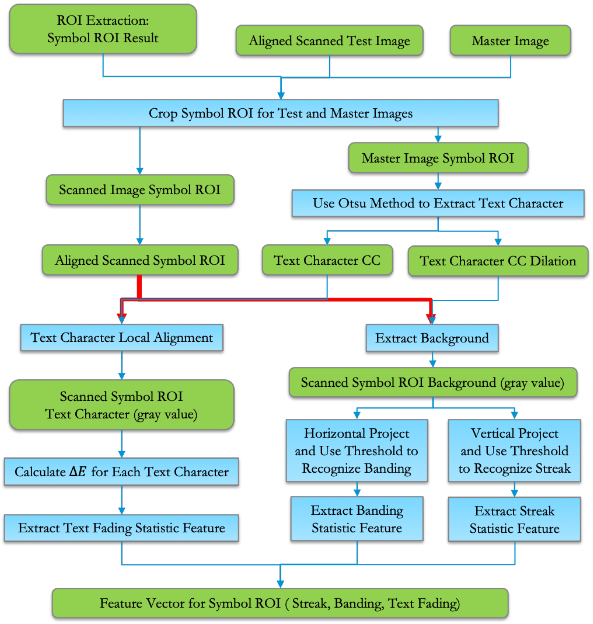

ROI Extraction: Symbol ROI Result
Aligned Scanned Test Image
Master Image
Crop Symbol ROI for Test and Master Images
Master Image Symbol ROI
Scanned Image Symbol ROI
Use Otsu Method to Extract Text Character
Aligned Scanned Symbol ROI
Text Character CC
Text Character CC Dilation
Text Character Local Alignment
Extract Background
Scanned Symbol ROI Background (gray value)
Scanned Symbol ROI Text Character (gray value)
Horizontal Project and Use Threshold to Recognize Banding
Vertical Project and Use Threshold to Recognize Streak
Calculate ΔE for Each Text Character
Extract Banding Statistic Feature
Extract Streak Statistic Feature
Extract Text Fading Statistic Feature
Feature Vector for Symbol ROI (Streak, Banding, Text Fading)

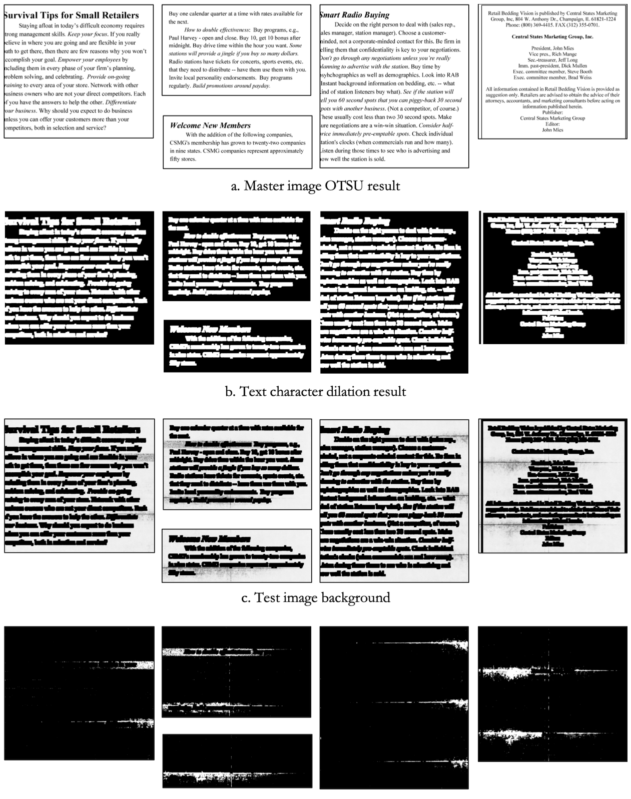
Figure 18: The pipeline for symbol ROI defect feature vector extraction for a single symbol ROI.

care about at this point. There are 5 different symbol ROIs for the sample in Figure 17.

Secondly, we need to transform each of the master symbol ROI cropped images from RGB color space to CIE $L^*a^*b^*$ color space and use the Otsu method [19] to extract the text character based on the gray value symbol master ROI image. Figure 19 (a) shows the cropped master symbol ROI Otsu result.

Thirdly, we apply a morphological operation to the cropped master symbol Otsu result. Figure 19 (b) shows the text character dilation result, where the white area is the region of interest. Based on the misalignment between the master image and the aligned test image, we set the text character dilation kernel to be 9 × 9 pixels for the 600 dpi scanned test image.

Finally, we use the text character dilation result of the master image to remove the text character area from the scanned test image, and we will get the background of the scanned image symbol ROI, shown in Figure 19 (c). we apply the Otsu method to this scanned image symbol ROI background area to extract the defect from the background area. In the next paragraph, we will introduce the details of how to use the Otsu method to extract the defect area from the symbol background area.


a. Master image OTSU result

b. Text character dilation result

c. Test image background

d. Background defect result

Figure 19: The symbol ROI background defect extraction process.

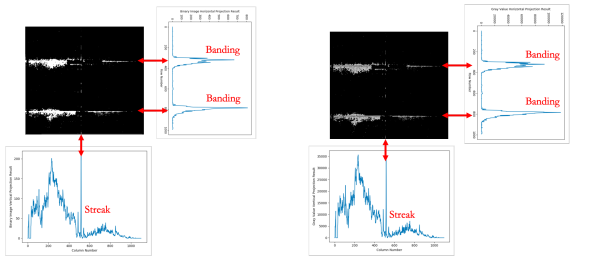
The first step to extract the background defect in the symbol ROI is to calculate the ΔE for each corresponding pixel in each master and test symbol ROI. We use Equation 4 to calculate the ΔE; $[i,\ j]$ is the position of the corresponding pixels in the master and test symbol ROI; T is the test image; M is the master image; and the superscript c is the L, a, or b channel.

$$\Delta E[i,j] = \sqrt{\sum_{c=L,a,b} (T^c[i,j] - M^c[i,j])^2} \tag{4}$$

To make the ΔE values look more intuitive, we normalize them to $0-255$, and use this normalized value to produce a gray value image. Then, we can use the Otsu method again for this ΔE gray value image to extract the symbol ROI background defect, shown in Figure 19 (d). In Figure 19 (d), only the white areas in the symbol ROIs are the symbol ROI background defects.

After we get the symbol ROI background defect result for the test image, we can extract the feature vector of these defects. There are two types of defects in the symbol ROI background: streaks and banding defects. We need to project the ΔE gray value and the number of defective pixels in the horizontal and vertical directions to extract these defect features. Figure 20 (a) shows the number of defect pixels projection results in the horizontal and vertical directions; and Figure 20 (b) shows the ΔE gray value of the defect projection results in the horizontal and vertical directions. Before extracting the streak and banding features from the projection images, we need to set a threshold to label the streaks and bands, because there are some noise defects detected in the background Otsu result. Here, we set the threshold to be the average projection value plus the standard deviation of the projection value. After we use this threshold to detect the streak and banding defects, for each symbol ROI, we can extract 33 features from the background. The description of these 33 features is provided in Table 1.

As the next step, we extract the text fading defect feature vector. Even though we have done the image alignment for the master image, there still is a little misalignment between master and alignment test image, and the misalignment is about 3 to 7 pixels for the 600 dpi scanned letter image. So, we use the connected component algorithm for the text characters (Otsu result) in the master symbol ROI. Then, we use a template matching method to find the accurate position of each text character in the test image. We use the normalized cross-correlation function to find the best template matching position [8], [20]. The template matching result is shown in Figure 21. After acquiring the accurate position of each text character in master and test images, we can separate the text



a. Background defect binary image projection result

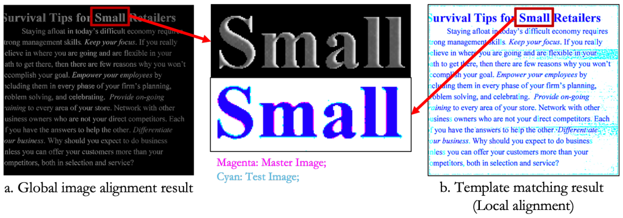
b. Background defect gray value image projection result

Figure 20: The symbol ROI background binary image and gray value image defect projection results.

Table 1: Symbol ROI Background Defect Feature Vector

1	The average of defect pixels ΔE value;
2	The number of streaks;
3	The total width of streaks;
4	The average length of streaks;
5	The average sharpness of streaks;
6	The number of bands;
7	The total width of bands;
8	The average length of bands;
9	The average sharpness of bands;
10	The width of the $1st$ streak;
11	The length of the $1st$ streak;
12	The sharpness of the $1st$ streak;
13	The severity of the $1st$ streak;
14	The width of the $2nd$ streak;
15	The length of the $2nd$ streak;
16	The sharpness of the $2nd$ streak;
17	The severity of the $2nd$ streak;
18	The width of the $3rd$ streak;
19	The length of the $3rd$ streak;
20	The sharpness of the $3rd$ streak;
21	The severity of the $3rd$ streak;
22	The width of the $1st$ band;
23	The length of the $1st$ band;
24	The sharpness of the $1st$ band;
25	The severity of the $1st$ band;
26	The width of the $2nd$ band;
27	The length of the $2nd$ band;
28	The sharpness of the $2nd$ band;
29	The severity of the $2nd$ band;
30	The width of the $3rd$ band;
31	The length of the $3rd$ band;
32	The sharpness of the $3rd$ band;
33	The severity of the $3rd$ band;

area and the background in both master and test images. For the text area, we will extract the text fading defect feature vector by calculating the ΔE value between master and test images in the text area. In the background area, we will extract the streak and bands defect feature vector by calculating the horizontal or vertical projection result. Finally, we will combine all the features in one feature vector, so there will be one feature vector for each symbol ROI.


Survival Tips for Small Retailers
Small
Small
Survival Tips for Small Retailers

a. Global image alignment result

Magenta: Master Image;
Cyan: Test Image;

b. Template matching result (Local alignment)

Figure 21: Template matching for text character.

In the template matching process, if the best template matching position is in the middle of the extended character area, we do not need to shift the character. However, if the best template matching position is not in the middle, we need to calculate the shift and move the character to match the correct position in the master image.

If the normalized cross-correlation value is very small (< 0.9), we do not extract this text character connected component.

For the text fading defect, we extract 12 features. The description of these 12 features is shown in Table 2.

Table 2: Symbol ROI Text Fading Defect Feature Vector

1	The average of L channel value of master text characters;
2	The average of a channel value of master text characters;
3	The average of b channel value of master text characters;
4	The average of L channel value of test text characters;
5	The average of a channel value of test text characters;
6	The average of b channel value of test text characters;
7	The average of ΔE value between master text characters and white color;
8	The average of ΔE value between test text characters and white color;
9	The standard deviation of ΔE value between master text characters and white color;
10	The standard deviation of ΔE value between test text characters and white color;
11	The average of ΔE value between master text characters and test text characters;
12	The standard deviation of ΔE value between master text characters and test text characters;

As we extract the background defect feature vector and text fading defect feature vector, we should combine them. The combined feature vector is the analysis result for one symbol ROI. After processing all the symbol ROIs, we can get a symbol ROI defect feature matrix.

2.6 Raster ROI defect feature vector extraction

For the raster ROI, we only consider three types of defects: streak, bands, and the color fading defect. We have shown the streak and bands defect before; so Figure 22 only shows the color fading defect. Color fading usually happens when one or more of the printer cartridges cyan, magenta, yellow, or black is low. It is essential to know which cartridge is depleted, given a faded print for the customer. We propose a method to extract the feature vector of the color fading defect to detect this defect.

Firstly, we should crop out the corresponding raster ROI from the master image and scanned test image based on the ROI extraction result. This step is similar to the symbol ROI defect feature vector extraction.

Secondly, we need to transform the cropped raster master and test images from the RGB color space to CIE $L^*a^*b^*$

color space and calculate the ΔE value between corresponding master and test raster ROI.

After these two steps, we can extract the following 14 features, as shown in Table 3.

Table 3: Raster ROI Color Fading Defect Feature Vector

1	The average of L channel value of master;
2	The average of a channel value of master;
3	The average of b channel value of master;
4	The standard deviation of L channel value of master;
5	The standard deviation of a channel value of master;
6	The standard deviation of b channel value of master;
7	The average of L channel value of test;
8	The average of a channel value of test;
9	The average of b channel value of test;
10	The standard deviation of L channel value of test;
11	The standard deviation of a channel value of test;
12	The standard deviation of b channel value of test;
13	The average of ΔE value between master and test corresponding;
14	The standard deviation of ΔE value between master and test corresponding;

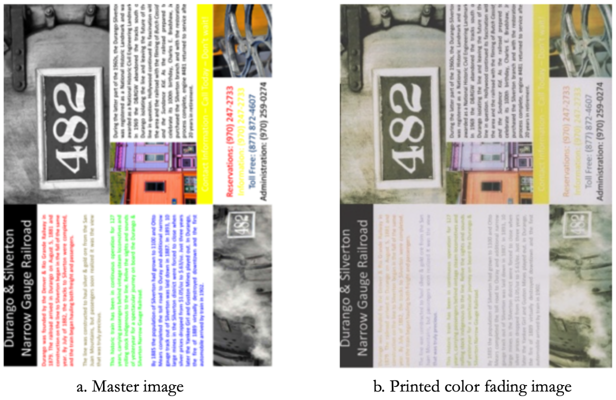
For the streak and bands defect, we extract seven features from the corresponding master and test raster ROI. In the streak and bands defect features, we usually use the L channel value because the a and b channel values are not indicative of the streak and bands defects. The description of these features is shown in Table 4.

Table 4: Raster ROI Streaks/Bands Defect Feature Vector

1	The average of L channel value of master raster ROI;
2	The average of ΔE value between the master and test corresponding ROI;
3	The average of L channel difference between the master and test corresponding ROI;
4	The standard deviation of L channel difference between the master and test corresponding ROI;
5	The average of L channel difference between the master and test corresponding ROI vertical projection result;
6	The standard deviation of L channel difference between the master and test corresponding ROI vertical projection result;
7	The average of L channel difference between the master and test corresponding ROI horizontal projection result;
8	The standard deviation of L channel difference between the master and test corresponding ROI horizontal projection result;

2.6 Bankground and Color Vector ROI defect feature vector extraction

The color fading defect for the color vector ROI is the same as the color fading defect in the raster ROI. So we can extract the same feature vector for the color fading defect in the color vector ROI; and we do not introduce this feature vector in this section. The streaks and bands defects in the background and color vector ROI are similar to those in the raster ROI. But there may be periodic streaks and bands in the extensive background and color vector ROI. Thus, there are two additional features for periodic streak and banding defect, which are listed in Table 5.


482
Durango & Silverton Narrow Gauge Railroad
Reservations: (970) 247-2733
Toll Free: (877) 872-4607
Administration: (970) 259-0274
a. Master image

b. Printed color fading image

Figure 22: The An sample showing the effect of black cartridge running low on toner.

Table 5: Background and Color Vector ROI Two Additional Features

1	The high-frequency energy of streak;
2	The high-frequency energy of bands;

Finally, we can extract nine features for the streak and banding defects in a background or color vector ROI, and 14 features for the color fading defect in a color vector ROI.

3. Conclusion

This paper proposed a comprehensive system to analyze a printed page automatically and extract the critical defect features to determine the type and to severity of defects on the scanned page. This comprehensive system incorporates many of our previous works: image alignment, color calibration, object map procedure, region of interest (ROI) extraction, text fading detection, color fading detection, streak detection, and bands detection. The input to this comprehensive system is the master image and the scanned test image. We analyze the scanned images based on different ROIs, and there are four kinds of ROIs. The symbol ROI includes 33 streak and bands features in the symbol background area, and 12 text fading features for the text character area. The raster ROI includes 14 color fading features, and 7 streak and bands features. The color vector ROI includes 14 color fading features, and 9 streak and bands features. The background ROI includes 9 streak and bands features. Finally, we combine all the feature vectors for one test image. If any type of ROI has not been identified on a given test page, that part of the composite feature vector corresponding to this type of ROI will be filled in as zero. This operation can guarantee all the composite feature vectors for different pages are the same size. This final image quality analysis result will be uploaded to the service provider server and used to analyze the printer defects.

References

[1] M. Q. Nguyen, S. Astling, R. J. Jessome, E. Maggard, T. Nelson, M. Q. Shaw, and J. P. Allebach, "Perceptual Metrics and Visualization Tools for Evaluation of Page Uniformity," *Electronic Imaging, Image Quality and System Performance XI,* Eds. San Francisco, CA, 3-5 February 2014.

[2] R. Zhang, E. Maggard, R. Jessome, Y. Bang, M. Cho, J. P. Allebach, "Block Window Method with Logistic Regression Algorithm for Streak Detection," *Electronic Imaging, Image Quality and System Performance XVI*, Burlingame, CA, January 2018.

[3] S. Hu, H. Nachlieli, D. Shaked, S. Shiffman, and J. P. Allebach, "Color-dependent Banding Characterization and Simulation on Natural Document Images," *Electronic Imaging, Color Imaging XVII: Displaying, Processing, Hardcopy, and Applications,* Eds., San Francisco, CA, 23-26 January 2012.

[4] Z. Xiao, S. Xu, E. Maggard, K. Morse, M. Shaw and J. P. Allebach, "Detection of Color Fading in Printed Customer Content," *Electronic Imaging, Color Imaging XV: Image Quality and System Performance,* Burlingame, CA, January 2018.

[5] Z. Xiao, M. Nguyen, E. Maggard, M. Shaw, J. P. Allebach, and A. Reibman, "Real-time Print Quality Diagnostics," *Electronic Imaging, Color Imaging XIV: Image Quality and System Performance,* Burlingame, CA, January 2017.

[6] Z. Xiao, M. Gao, L Wang, B Bradburn, and J. P. Allebach, "Digital Image Segmentation for Object-Oriented Halftoning," *Electronic Imaging, Color Imaging XXI: Displaying, Processing, Hardcopy, and Applications,* Burlingame, CA, January 2016.

[7] R. Zhang, E. Maggard, Y. Bang, M. Cho, J. P. Allebach, "Region of Interest Extraction for Image Quality Assessment," *Electronic Imaging, Image Quality and System Performance XVII,* Burlingame, CA, January 2018.

[8] Y. Ju, E. Maggard, R. J. Jessome, and J. P. Allebach, "Autonomous Detection of Text Fade Point with Color Laser Printers," *Electronic Imaging, Image Quality and System Performance XII,* Eds. San Francisco, CA, 8-12 February 2015.

[9] W. Gao, X. Zhang, L. Yang and H. Liu, "An Improved Sobel Edge Detection," *IEEE, 3rd International Conference on Computer Science and Information Technology (Vol. 5, pp. 67-71),* July, 2010.

[10] L. Di Stefano and A. Bulgarelli, "A Simple and Efficient Connected Components Labeling Algorithm," *IEEE, Proceedings 10th International Conference on Image Analysis and Processing (pp. 322-327),* September, 1999.

[11] M. L. Comer, and E.J. Delp, "Morphological Operations for Color Image Processing," *J. Electronic Imaging,8(3), pp. 279-289,* 1999.

[12] A. Meijster, J. B. Roerdink, and W.H. Hesselink, "A General Algorithm for Computing Distance Transforms in Linear Time," *Mathematical Morphology and its Applications to Image and Signal Processing (pp. 331-340). Springer,* Boston, MA. 2002.

[13] N. Ikizler, and P. Duygulu, "Histogram of Oriented Rectangles: A New Pose Descriptor for Human Action Recognition," *Image and Vision Computing, 27(10), pp. 1515-1526.* 2009.

[14] K. M. Lam, and H. Yan, "Fast Greedy Algorithm for Active Contours," *Electronics Letters, 30(1), pp. 21-23,* 1994.

[15] J. Ha, R.M. Haralick, and I.T. Phillips, "Recursive XY Cut Using Bounding Boxes of Connected Components," *IEEE, Proceedings of 3rd International Conference on Document Analysis and Recognition (Vol. 2, pp. 952-955),* August, 1995

[16] J. Chen, L.H. Zou, J. Zhang, and L.H. Dou, "The Comparison and Application of Corner Detection Algorithms," *J. Multimedia, 4(6),* 2009.

[17] K. Nickels and S. Hutchinson, "Estimating Uncertainty in SSD-based Feature Tracking," *Image and Vision Computing, 20(1), pp. 47-58,* 2002.

[18] K.G. Derpanis, "Overview of the RANSAC Algorithm," *Image Rochester NY, 4(1), pp. 2-3,* 2010.

[19] D. Liu, and J. Yu, "Otsu Method and K-means," *IEEE, Ninth International Conference on Hybrid Intelligent Systems (Vol. 1, pp. 344-349),* August, 2009.

[20] P. Van Dam, S. Mouton and P. Oosterhoff, Medtronic Inc, "Template Matching Method for Monitoring of ECG Morphology Changes," *U.S. Patent 7,996,070,* 2011.

[21] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color Transfer Between Images," *IEEE Computer Graphics and Applications, 21(5), pp. 34-41,* 2001.

Author Biography

***Runzhe Zhang** received his B.S. degree in Mechanical Engineering from Qingdao University of Technology (2013), Shandong, China and received the M.S. degree in Mechanical Engineering from Boston University, MA, USA. Currently, he is pursuing the Ph.D. in Electrical and Computer Engineering at Purdue University. His research areas include digital image processing, computer vision, and machine learning.*

***Yi Yang** received her M.S in Geomatics Engineering from the Chinese Academy of Sciences in 2016. She is currently working on a Ph.D. in Electrical and Computer Engineering at Purdue University. Her primary area of research has been image processing, computer vision, and machine learning.*

***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 the Imaging Lab at HP Printing Korea Co. Ltd. She received her BS and MS 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 an engineer with HP Printing Korea. He received B.S.(1997) and M.S.(1999) in electrical engineering from the 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 and calibration. Recently, he is researching the above interests using ML and DL.*

Jan P. Allebach *is Hewlett-Packard Distinguished Professor of Electrical and Computer Engineering at Purdue University. Jan P. 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 the IS&T/OSA Edwin Land Medal and is a member of the National Academy of Engineering.*

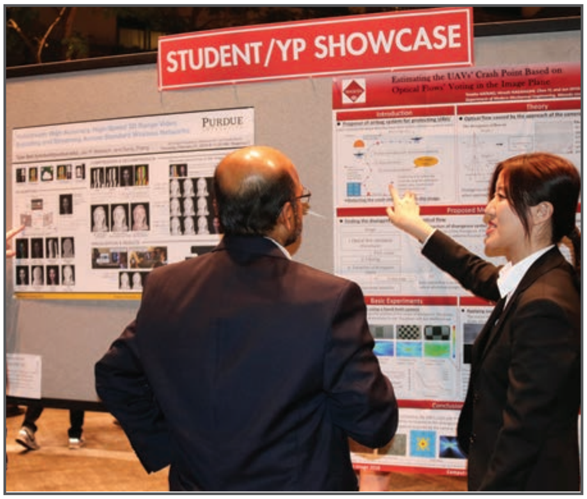
JOIN US AT THE NEXT EI!

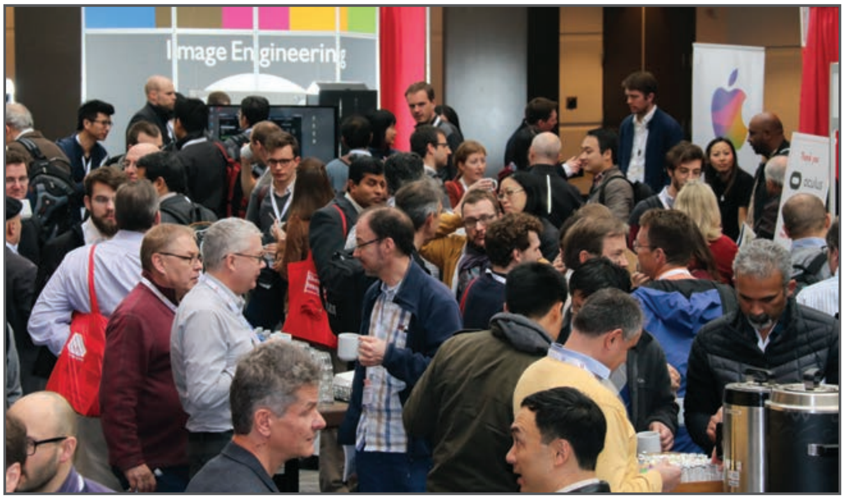
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





- AR is a fundamental and transformative interface and platform.
- 20+ years from my initial working demo to today's AR displays
- Optimistic that it won't take another 20 to reach ubiquitous consumer AR!

My first working AR prototype

Today's platforms

Optimis this is << 20

Ubiquitous consumer AR

• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org