https://doi.org/10.2352/ISSN.2470-1173.2020.9.IQSP-321

Region of interest extraction for image quality assessment*

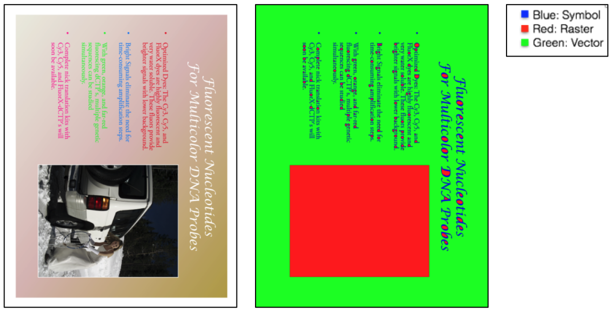
Runzhe Zhang [a], *Eric Maggard* [b], *Yousun Bang* [c], *Minki Cho* [c], *Jan Allebach* [a]
[a] *School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, U.S.A.*
[b] *HP Inc., Boise, ID 83714, U.S.A.*
[c] *HP Inc., Suwon City, KOREA.*

Abstract

Print quality (PQ) is most important in the printing industry. To detect and analyze print defects is an effective solution to improve print quality. As the different types of print defects appear in different regions of interest (ROI) in the digital image of a scanned page, extracting the different ROIs helps to detect and analyze the printer defect. This paper proposes a method to extract different ROIs based on the digital image object map [1], which includes three different labels: raster (images or pictures), vector (background and smooth gradient color areas), and symbol (symbols and texts). Our ROI extraction method will extract four kinds of ROIs based on these three labeled objects. So we need to distinguish the background area and smooth gradient color area (color vector) from other vector objects. The process of the ROI extraction method includes four parts; and each part will extract one kind of ROI. For the color vector and background ROI extraction part, we develop two approaches: one is to obtain the maximum area rectangular ROI; and the other approach is to extract the deepest rectangular ROI. With both of these two methods, we use a greedy algorithm to gather additional useful ROIs. In the final result of the ROI extraction process, we only save the left top and right bottom positions for each ROI. In the end, we design a Matlab GUI Tool and label the ROI ground truth manually. We calculate the intersection over union (IoU)) between the ROI extraction result and the ROI manually labeled ground truth to evaluate our ROI extraction algorithm, and check whether it is good enough to crop different ROIs from the image of the scanned page to detect and analyze print defects.

1. Introduction

The electrophotographic (EP) process widely used in a modern laser printer is susceptible to various print defects. As different frequency halftone screens and toners may used in different regions of the image based on the customer content, this influences only one kind of area, including the specific print defects that may occur. For example, the text fading defect only happens in the symbol ROIs, color fading defects happen in raster and color vector ROIs [3], and streak defects may happen anywhere on the printed page [4]. Based on these reasons, we should extract different ROIs from the digital image before we detect and analyze the print defects. This ROI extraction process is different from that developed for analyzing inkjet printed functional biosensors [2]. This paper proposes a ROI extraction method based on the digital image object map [5], which includes three different objects: raster (picture or photo areas), vector (background and smooth gradient color areas), and symbol (symbol and text areas), as is shown in Figure 1. In the object map, we use blue color to label symbol pixels, red color to label raster pixels, and green color to label vector pixels. The digital master image and the object map are the input to the ROI extraction algorithm. The output is the four kinds of different ROI results: symbol, raster, color vector, and background ROIs, as shown in Figure 2. We distinguish between the color vector area (uniform or gradient color) and background area (no color) region in the vector object in the object map. Each ROI is a rectangular area of the page.



a. Master image b. Object map

Figure 1: Input images: master image and object map

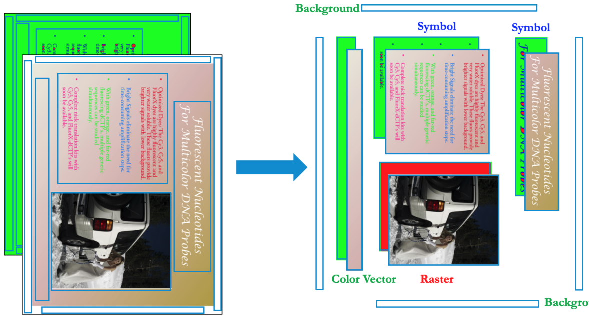


Figure 2: ROI extraction result: symbol, raster, color vector, and background ROIs

In the ROI extraction process, we prefer to make the ROI rectangular area for each type of ROI as big as possible, be-

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

cause in a large rectangular area, it is easier to detect and analyze print defects. Based on this requirement, we design two methods to crop color vector and background ROIs. One approach uses the 2-D “Chessboard” distance [6] to calculate the depth of the interest region and extract the deepest rectangular area. Another method calculates the maximum area rectangular area of interest region [7]. Because only obtaining the deepest and maximum rectangular ROI is not sufficient to detect and analyze all the printer defects on the test page, we use a greedy algorithm [8] to get more useful ROIs.

We will cover the details of the ROI extraction process, and the evaluation results in the following sections.

2. Region of interest(ROI) extraction

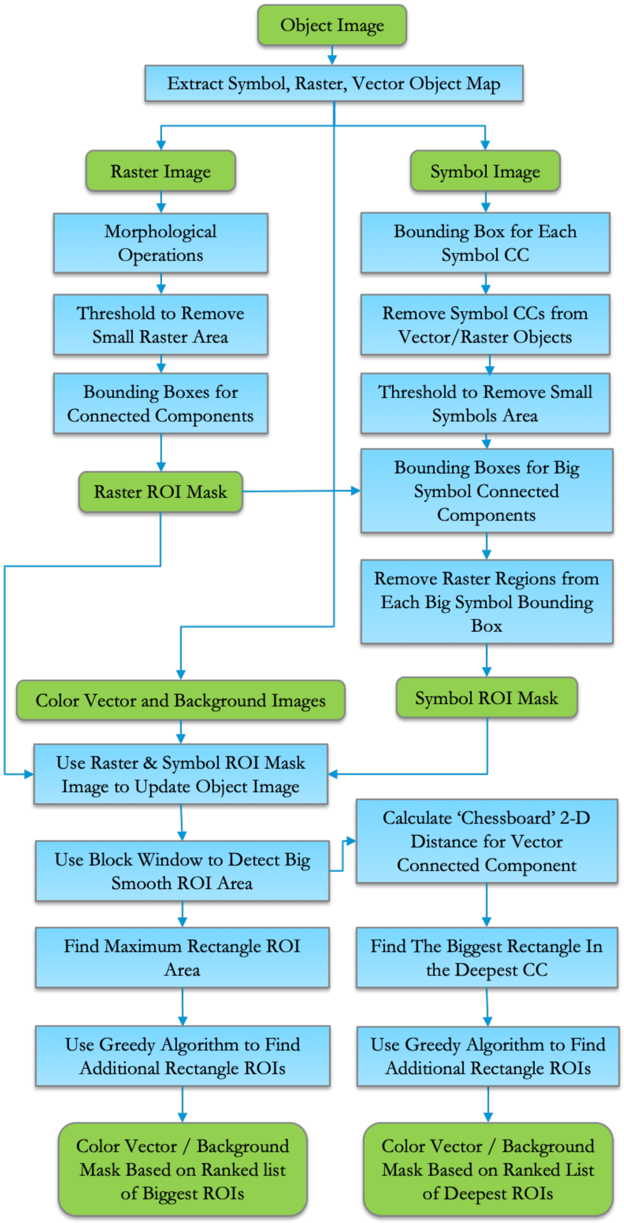

Object Image
Extract Symbol, Raster, Vector Object Map
Raster Image
Symbol Image
Morphological Operations
Bounding Box for Each Symbol CC
Threshold to Remove Small Raster Area
Remove Symbol CCs from Vector/Raster Objects
Bounding Boxes for Connected Components
Threshold to Remove Small Symbols Area
Raster ROI Mask
Bounding Boxes for Big Symbol Connected Components
Remove Raster Regions from Each Big Symbol Bounding Box
Color Vector and Background Images
Symbol ROI Mask
Use Raster & Symbol ROI Mask Image to Update Object Image
Calculate ‘Chessboard’ 2-D Distance for Vector Connected Component
Use Block Window to Detect Big Smooth ROI Area
Find Maximum Rectangle ROI Area
Find The Biggest Rectangle In the Deepest CC
Use Greedy Algorithm to Find Additional Rectangle ROIs
Use Greedy Algorithm to Find Additional Rectangle ROIs
Color Vector / Background Mask Based on Ranked list of Biggest ROIs
Color Vector / Background Mask Based on Ranked List of Deepest ROIs

Figure 3: The overall pipeline of ROI extraction

In this section, we introduce the details of the ROI extraction process. The overall pipeline of the proposed method is shown in Figure 3. It includes four parts: 1. Raster ROI extraction; 2. Symbol ROI extraction; 3. Color vector ROI extraction; 4. Background ROI extraction. Because the theory of the methods to extract the color vector ROIs and the background ROIs are the same, and background ROIs appear more frequently, we only introduce the background ROI extraction process.

First of all, we have two input images: the master image and the corresponding object map. Because the object map includes three different objects, we use it to produce three object maps first, as shown in Figure 4. They are the symbol, the raster, and the vector object maps (only the white color region is the object area).

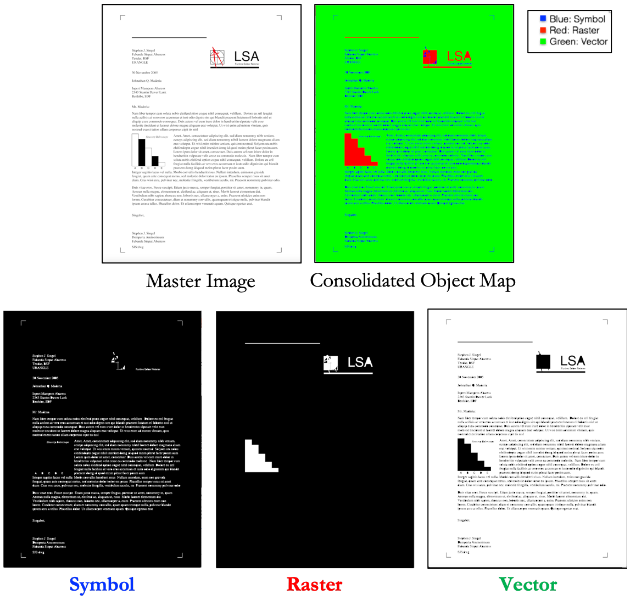


Figure 4: The input images: master image, consolidated object map, and three separated object maps.

2.1 Raster ROI extraction

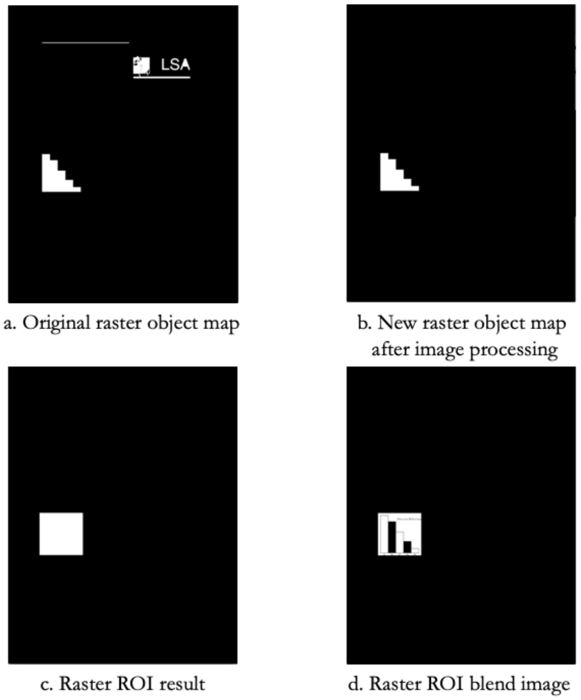
In the raster ROI extraction process, we do the morphological operations [9] on the raster object map initially, as shown in Figure 5 (a), to remove imperfections. We account for the form and structure of the image by using the average connected symbol component height as the morphological operation kernel size. We first do the dilation operation, and then continue to do the erosion operation. After the morphological operations, most of the raster areas will be connected; and we use the connected components algorithm to generate a new raster object map and use the bounding boxes [10] to label each connected raster component. For the new raster object map, we separate all the raster ROIs under three conditions to remove the symbol objects:

1. The raster object pixels account for more than 80% of the bounding box of the raster connected component. Under this condition, the bounding box of the connected component is retained as a raster ROI.
2. The raster object pixels account for more than 20% of the bounding box of the connected component; but sym-

bol object pixels occupy none or less than 20% of the remaining area. Under this condition, the bounding box of the connected components is still retained as a raster ROI.

3. The raster object pixels account for more than 20% of the bounding box of the connected component, but there is more than 20% area of the symbol object pixels in the remaining area. In this condition, we cut off the smallest rectangular area, which includes all symbol objects. The remaining area is retained as a raster ROI.

After processing the bounding box of each raster connected component [11], we get a new raster object map. We use the connected component algorithm again and remove the small raster connected components, as shown in Figure 5 (b). There are three thresholds to remove the small raster ROIs: the width of the raster ROI is more than 10% of the page width; the height of the raster ROI is more than 10% of the page height; and the area of the raster ROI is more than 5% of the page area. The bounding boxes of the connected components of the raster object are the raster ROI result, shown in Figure 5 (c), where only the white areas are the ROIs. Figure 5 (d) shows the raster ROIs and master image blending image.



a. Original raster object map

b. New raster object map after image processing

c. Raster ROI result

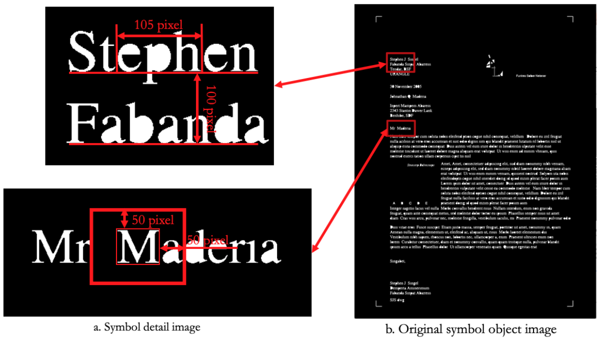
d. Raster ROI blend image

Figure 5: The raster ROI extraction process.

2.2 Symbol ROI extraction

After we extract the raster ROIs, we will extract the symbol ROI using the symbol object map and the raster ROI extraction result. In the symbol object map, most of the symbol area is comprised of text characters, and they are not connected. In the symbol ROI extraction part, we want to crop the largest rectangular areas that include the symbol objects.

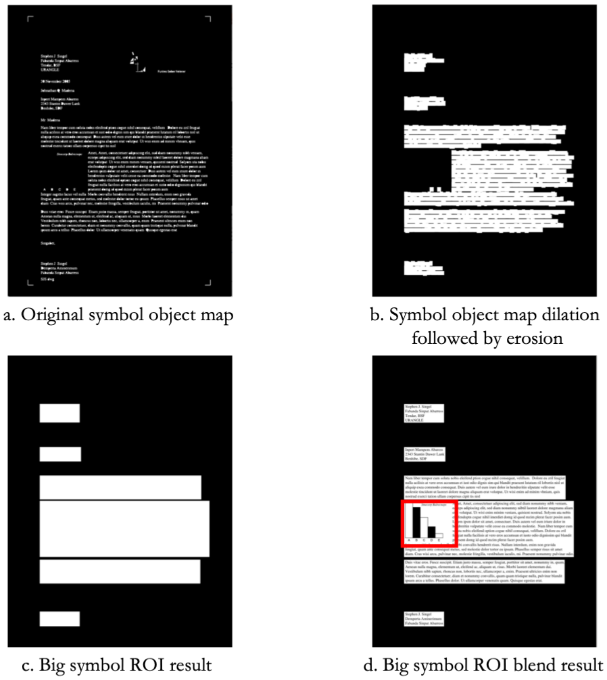
Firstly, we still use the connected component algorithm for the symbol object map to label each independent text character. We observe the distance between adjacent elements of each connected component in one paragraph, shown in Figure 6 (sample image scanned at 600 dpi). Based on the observed



a. Symbol detail image

b. Original symbol object image

Figure 6: The observation of symbol text characters.



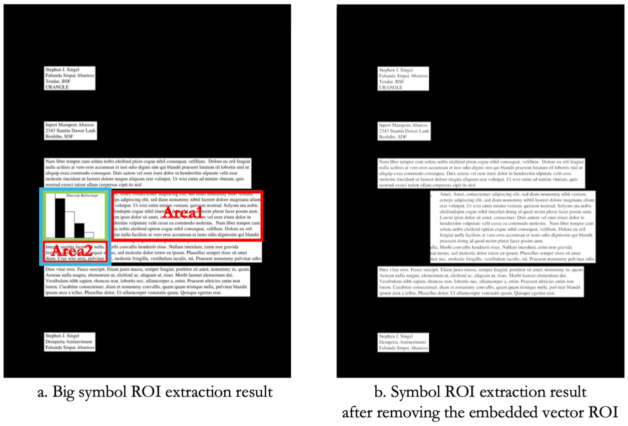
a. Original symbol object map

b. Symbol object map dilation followed by erosion

c. Big symbol ROI result

d. Big symbol ROI blend result

Figure 7: The big symbol ROI extraction result.



a. Big symbol ROI extraction result

b. Symbol ROI extraction result after removing the embedded vector ROI

Figure 8: The symbol ROI result.

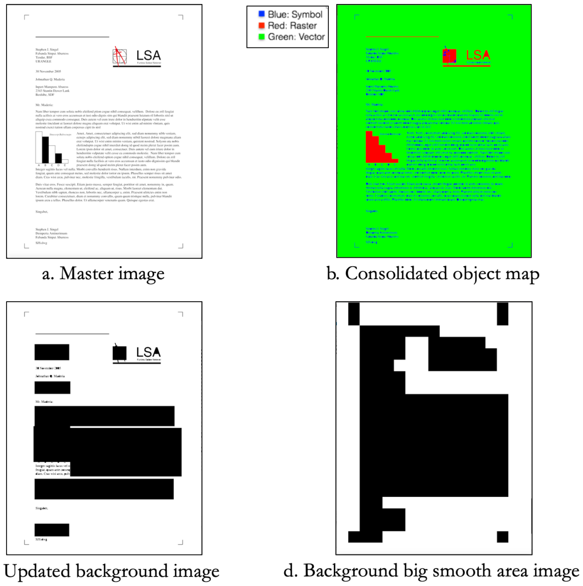
result, we use the average height of the text character connected components as the morphological operations kernel size to process the dilation followed by erosion for the symbol object connected component result.

After this step, all the text characters in one paragraph will be connected, as shown in Figure 7 (b). This connected symbol ROI result will be used to extract the significant symbol ROIs. We use the bounding box for the new symbol object map connected components to get the significant symbol ROI result, as shown in Figure 7 (d).

In this significant symbol ROI extraction result, we find that it may include a raster object. In this condition, we need to cut off the smallest rectangular area from the symbol ROI while trying to retain the symbol ROIs as much as possible. In Figure 8 (a), we find 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 red box Area 2 (vertical cut). In the general case, we want to cut off the small red area to keep as much of the symbol ROI as possible. In this condition, we will cut off the blue box, which includes the green box raster object and red box Area 2. After this step, we get the final symbol ROI result, shown in Figure 8 (b). Note that this entire process is predicated on the requirement that ROIs have a rectangular shape.

2.3 Color vector and background ROI extraction

The methods to extract color vector and background ROIs are essentially the same. The difference is that a color vector ROI is a printed uniform color area; and the background ROI does not have color. As most of the images include background areas, not color vector areas, we only show how to extract background ROIs.



a. Master image

b. Consolidated object map

c. Updated background image

d. Background big smooth area image

Figure 9: The preprocessing steps for the background object map.

First of all, we should separate the color vector objects and background objects in the original vector object map, because the color vector and background belong to the vector object in the original object map. In the following introduction, we only use the background object map. In our sample master image Figure 9 (a), the green color in the object map labels the vector object, as shown in Figure 9(b).

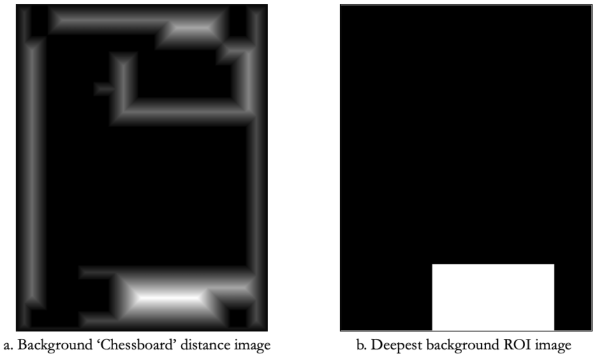
To extract the background ROIs, we need to mask the symbol and raster ROIs in the background object map, as shown in Figure 9 (c). In the mask background object map, the white area is the region of interest. We want to crop the largest rectangular area from the white region of interest. To reduce the calculation, we use a block window to separate the background object map and detect the big smooth area. We use a 300 $pixels \times 300$ $pixels$ window to identify significant background areas. For each block, if it includes a symbol ROI, or a raster ROI, it will be labeled as a block that is not of interest (black). Otherwise, it will be marked as a block that is of interest (white). The process result is shown in Figure 9 (d).

To extract the largest rectangular ROI from the background big smooth area, we design two methods: one is to calculate the 2-D “Chessboard” distance to extract the deepest rectangular ROI; another is to calculate the maximum rectangular area in the background significant smooth region.

2.3.1 Deepest ROI extraction

We use the 2-D “Chessboard” distance [6] to extract the deepest rectangular area from the background big smooth area. The “Chessboard” distance is a metric defined on a vector space where the distance between two vectors is the greatest of their differences along any coordinate dimension. If the points P and Q have Cartesian coordinates $(x_1,\ y_1)$ and $(x_2,\ y_2)$ in an image, their “Chessboard” distance is:

$$D_{Chessboard} = max\ (\ |x_1 - x_2|,\ |y_1 - y_2|\ ) \tag{1}$$



a. Background ‘Chessboard’ distance image

b. Deepest background ROI image

Figure 10: “Chessboard” distance to extract deepest background ROI.

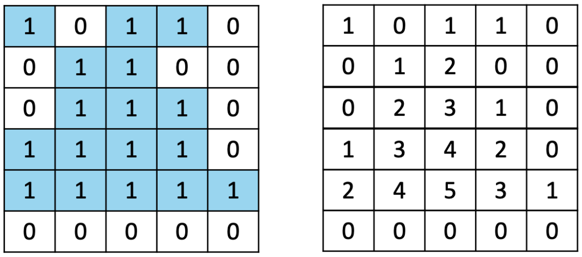
To apply this metric here, we assign each pixel in a background ROI a value that is equal to its “Chessboard” distance to the nearest boundary of the ROI. We use 2-D “Chessboard” distance to extract the ROI because the shape of the distance map looks like a set of rectangles. We calculate the 2-D “Chessboard” distance for the background big smooth area image and get the distance map. As we want to see the distance intuitively, we normalize the distance to 0 − 255, as shown in Figure 10 (a). The larger distance pixels will be more white in this normalized 2-D “Chessboard” distance map. Based on this reason, we extract the deepest rectangular area from the background big smooth area, as shown in Figure 10 (b).

2.3.2 Maximum area ROI extraction

Another way to identify the largest rectangular background ROI is to calculate the maximum rectangular area within the background big smooth region [7]. We should save the background big smooth area image as a binary image. The object pixels have value 1, and the other pixels have value 0. So we will have the binary background object matrix. In this paper, we use a small binary Matrix A, as shown in Figure 11 (a), to explain this calculation process. We then use the algorithm shown below to get the accumulation Matrix B, as shown in Figure 11 (b). The pseudocode of the algorithm is shown in Algorithm 1.

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
```



1	0	1	1	0
0	1	1	0	0
0	1	1	1	0
1	1	1	1	0
1	1	1	1	1
0	0	0	0	0

a. Binary object Matrix A

1	0	1	1	0
0	1	2	0	0
0	2	3	1	0
1	3	4	2	0
2	4	5	3	1
0	0	0	0	0

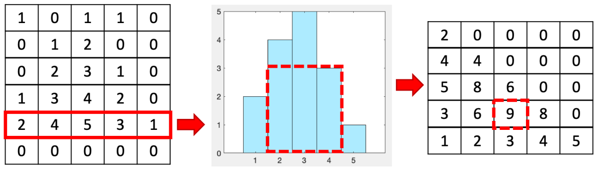
b. Object accumulation Matrix B

Figure 11: Binary object matrix and accumulation matrix.

In this algorithm, we process pixel by pixel from left to right and from top to bottom. If the pixel in the object matrix has value 1, the corresponding position of the accumulation matrix will be 1 plus the value above this pixel. If the pixel in the object matrix is 0, the corresponding position of the accumulation matrix will be 0. After processing all the pixels, we get the accumulation matrix.

With the aid of the accumulation Matrix B, the maximum rectangular ROI for each row of the Matrix B can be calculated. Here, we take the Row 5 in Figure 12 (a) as an example to show how to calculate the maximum rectangular ROI for one row of Matrix B. We first generate the bar graph of Row 5, which two dimensions are the element indices in the corresponding column and its corresponding element value, respectively, as shown in Figure 12 (b). After the bar graph is plotted, we calculate all possible rectangular areas that can be formed from each bar itself, or that bar combined with bars to the left of it, to obtain our rectangle area Matrix C, is shown in Figure 12 (c).

For example, the first row of the Matrix C corresponds to all the possible rectangular areas that the first bar can have in Figure 12 (b). The first bar, itself, has a rectangular area of 2 ($height \times width$); and there are no other bars to the left of it. So, there is only one rectangular area for the first bar; and the elements of the first row in the rectangular area Matrix C are [2, 0, 0, 0, 0]. For the second bar in the bar graph, its rectangular area is 4, and the largest rectangular area of the combined second bar and the first bar (the bar to the left of the second bar) is 4. Since there are no other bars to the left of the second bar except for the first bar, there are two rectangular areas for the second bar, each with area 2. So we have that the second row of the Matrix C is [4, 4, 0, 0, 0]. For the third bar in the bar graph, its area is 5, and the largest rectangular area of itself and the second bar (the bar to the left of the third bar) is 8. We continue to merge the third bar, second bar, and first bar to calculate the maximum rectangular area of 6. Because there are no other bars to the left of the third bar, except for the first bar and the second bar, there are three rectangular areas for the third bar; and we complete the third row of the Matrix C by recording the calculated maximum rectangular areas, which are [5, 8, 6, 0, 0]. Similarly, we calculate all possible rectangular areas of the remaining bars in the bar graph to obtain the final rectangular area Matrix C. Based on Matrix C, we can find the maximum rectangular area for Row 5 in Matrix B, which is 9. The corresponding rectangular area is labelled by the dotted red lines in Figure 12(b).



1	0	1	1	0
0	1	2	0	0
0	2	3	1	0
1	3	4	2	0
2	4	5	3	1
0	0	0	0	0

a. Object accumulation Matrix B

b. Bar graph for Row 5 in Matrix B

2	0	0	0	0
4	4	0	0	0
5	8	6	0	0
3	6	9	8	0
1	2	3	4	5

c. Rectangle area Matrix C for Row 5 in Matrix B

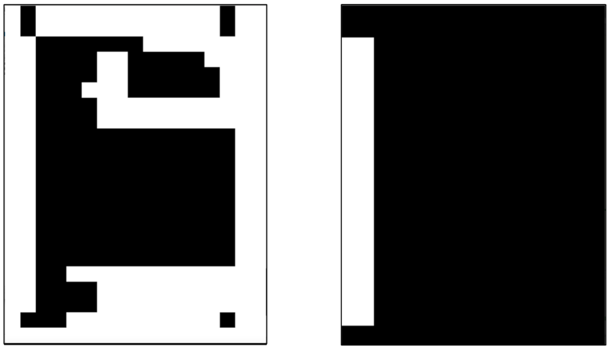
Figure 12: Calculate the maximum rectangle for one row.

Algorithm 2 MAXIMUM RECTANGLE

```
for row i do
    for col $j_1$ do
        for col $j_2 < j_1$ do
            Calculate the maximum area between $j_1$ and $j_2$
            if Current Calculation Area > Record Maximum Area then
                Update the record maximum area
            end if
        end for
    end for
end for
```

We can process all the rows of Matrix B by using the same method. Then, the maximum rectangular ROI of all the rows, as represented by the element with largest value in the Matrices C for all the rows, is the maximum rectangular ROI for the whole page. The pseudocode of this maximum area rectangle algorithm is shown in Algorithm 2, which determines the largest area rectangular region within the background big smooth region.

After executing these two algorithms using the background big smooth image, we get the maximum background area rectangle ROI, as shown in Figure 13.



a. Background big smooth area image

b. Maximum background area rectangle ROI

Figure 13: The maximum background area rectangle ROI result for the image shown in Figure 9.

2.3.3 *Use of the greedy algorithm to improve the ROI extraction result*

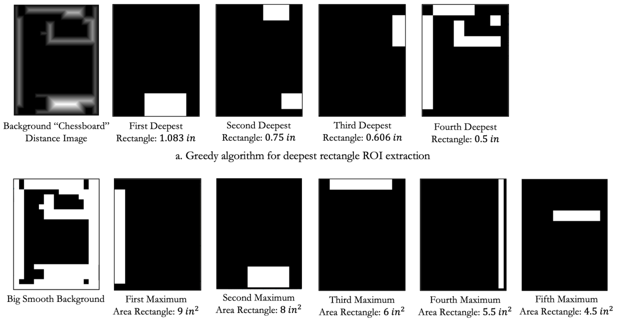
In the background ROI extraction method described so far, we only find the deepest or maximum rectangular background ROI. This is not sufficient to detect all the print defects in the background region of the page. In the background big smooth area image a lot of background area remains be extracted. Here, we use a greedy algorithm [12] to recursively extract a ranked list of ROIs according to the deepest rectangle method and the maximum area rectangle algorithm.

The greedy algorithm for extraction of a set of ROIs, according to the deepest ROI rectangle method is shown in Figure 14 (a):

1. Set a threshold for the smallest deep rectangular area that may include a defect. For this 600 dpi sample, we set it to be 300 $pixels = 0.5\ in$;
2. Use the “Chessboard” distance to find the deepest rectangle from the background big smooth area image;
3. After extracting the deepest rectangle, mask this area as not being of interest;
4. Then, find the new deepest rectangle in the new “Chessboard” distance image.
5. Repeat Steps 2-4 until the minimum dimension of the deepest rectangle that is found is below the threshold.

The results from using the greedy algorithm for extraction of a set of ROIs, according to the maximum area rectangle ROI extraction method is shown in Figure 14 (b):

1. Set a threshold for the smallest useful rectangular area. For this 600 dpi sample, we set it to be 300 $pixels \times 300\ pixels \times 10 = 0.5\ in \times 0.5\ in \times 10 = 2.5\ in^2$;
2. Use the same maximum area rectangle algorithm shown before to extract the maximum area rectangle from the background big smooth area image;
3. After extracting the maximum area rectangle, mask this area as not being of interest;
4. Then, find the new maximum area in the image.
5. Repeat Steps 2-4 until the area of the maximum area rectangle that is found is below the threshold.



Background “Chessboard” Distance Image

First Deepest Rectangle: 1.083 in

Second Deepest Rectangle: 0.75 in

Third Deepest Rectangle: 0.606 in

Fourth Deepest Rectangle: 0.5 in

a. Greedy algorithm for deepest rectangle ROI extraction

Big Smooth Background

First Maximum Area Rectangle: 9 in^2

Second Maximum Area Rectangle: 8 in^2

Third Maximum Area Rectangle: 6 in^2

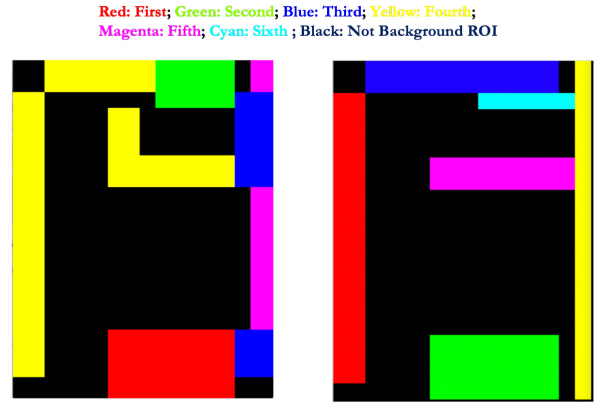
Fourth Maximum Area Rectangle: 5.5 in^2

Fifth Maximum Area Rectangle: 4.5 in^2

b. Greedy algorithm for maximum area rectangle ROI extraction

Figure 14: The greedy algorithm application.

After all, we get the background ROI result, as shown in Figure 15. The sequence of ROI extraction results for maximum area and deepest rectangle ROI based on the greedy algorithm is Red, Green, Yellow, Magenta, and Cyan. Black is the region that is not of interest.


Red: First; Green: Second; Blue: Third; Yellow: Fourth;
Magenta: Fifth; Cyan: Sixth ; Black: Not Background ROI

a. Deepest rectangle background ROI extraction result

b. Maximum area rectangle background ROI extraction result

Figure 15: The background ROI extraction result after application of the greedy algorithm.

Because the structure of the color vector object map is the same as that of the background object map, except that the color vector has color and the background is white, we use the same method to extract the color vector ROIs.

After we extract the background and color vector ROIs, we have completed the entire ROI extraction process.

3. Experimental results

Before applying this ROI extraction algorithm to a print quality analysis system, we prefer to evaluate the ROI extraction algorithm result [13]. Here, we designed a MATLAB GUI tool [15], and manually labelled 100 page images to generate ground truth. In this MATLAB GUI tool, we use a cursor to choose the left top and right bottom pixels of each ROI rectangle area, and choose the corresponding ROI label to save. After we label the image ROIs' ground truth, the MATLAB GUI tool uses rectangles of four different colors to show all four kinds of the labeled ROI ground truth areas, as illustrated in Figure 16.

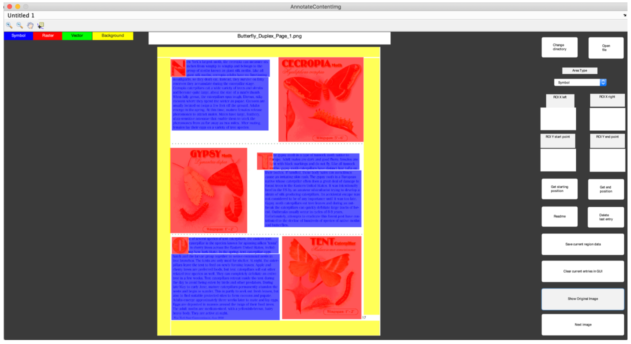


Figure 16: The MATLAB GUI tool.

To evaluate the ROI extraction algorithm result, we use the intersection over union (IoU) method [14]. The experimental results are shown in Table 1. Here, "Deepest" means that the ROIs were extracted using the Deepest rectangle method; and "Maximum" means that the ROIs were extracted using the Maximum area rectangle method. Because the Maximum Area ROI extraction method IoU experimental results are much better than the Deepest ROI extraction results, we use the Maximum Area ROI extraction method in the image quality analysis system to extract the different ROIs from master images.

Table 1: ROI extraction algorithm experimental results.

ROI Type	IoU Accuracy
Raster	86.46%
Symbol	84.43%
Color Vector (Deepest)	73.00%
Background (Deepest)	68.00%
Color Vector (Maximum)	87.00%
Background (Maximum)	81.00%

4. Conclusion

This paper proposed a method to extract the different ROIs based on the digital image object map, which includes three kinds of labels: raster (pictures or photos), vector (background and smooth gradient color areas), and symbol (symbols and text). The ROI extraction method extracts four types of ROIs based on the three kinds of labeled objects because we distinguish between the background and smooth gradient (color vector) regions, both of which are classified as vector in the object map. The process of the ROI extraction method includes four parts. Each part will extract one type of ROI. We developed two methods to obtain the color vector and background ROIs: one is to get the deepest rectangular ROIs; the other is to capture the maximum area rectangular ROIs. We also use a greedy algorithm in the color vector and background ROI extraction processes to get the most useful set of ROIs.

References

[1] Z. Xiao, M. Gao, L Wang, B Bradburn, and J. Allebach, "Digital image segmentation for object-oriented halftoning," Electronic Imaging, Color Imaging XXI: Displaying, Processing, Hardcopy, and Applications, Burlingame, CA, January 2016.

[2] M. Zhao, R. Zhang, S. Diaz-Amaya, L. Lin, A. Deering, L. Stanciu, G. Chiu and J. Allebach, "Detection, imaging, and quantification of DNA-Based pathogen based on Inkjet-Printed test strips," NIP & Digital Fabrication Conference (Vol. 2019, No. 1, pp. 177-181). Society for Imaging Science and Technology, San Francisco, CA, September, 2019.

[3] R. Zhang, E. Maggard, R. Jessome, Y. Bang, M. Cho, J. Allebach, "Block window method with logistic regression algorithm for streak detection," Electronic Imaging, Image Quality and System Performance XVI, Burlingame, CA, January 2018.

[4] Z. Xiao, M. Nguyen, E. Maggard, M. Shaw, J. Allebach, and A. Reibman, "Real-time print quality diagnostics," Electronic Imaging, Color Imaging XIV: Image Quality and System Performance, Burlingame, CA, January 2017.

[5] M. Nguyen, S. Astling, R. Jessome, E. Maggard, T. Nelson, M. Shaw, and J. Allebach, "Perceptual metrics and visualization tools for evaluation of page uniformity," Electronic Imaging, Image Quality and System Performance XI, SPIE Vol. 9016. San Francisco, CA, 3-5 February 2014.

[6] 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.

[7] 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.

[8] M. Bleyer, and M. Gelautz, "A layered stereo algorithm using image segmentation and global visibility constraints," IEEE, International Conference on Image Processing,. ICIP'04. (Vol. 5, pp. 2997-3000). October, 2004.

[9] M.L. Comer, and E.J. Delp, "Morphological operations for color image processing," J. Electronic Imaging, 8(3), pp. 279-289. 1999.

[10] D. Hirschberg, A. Chandra and D. Sarwate, "Computing connected components on parallel computers," Communications of the ACM, 22(8), pp. 461-464. 1979.

[11] J. Ha, R. Haralick, and I. 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

[12] R. Ruiz, and T. Stützle, "A simple and effective iterated greedy algorithm for the permutation flowshop schedul-

ing problem," European Journal of Operational Research, 177(3), pp. 2033-2049, 2007.

[13] J. Phillips, and P. Phillips, "Handbook of training evaluation and measurement methods. Routledge," 2016.

[14] L. Tychsen-Smith, and L. Petersson, "Improving object localization with fitness NMS and bounded IOU loss," IEEE, Computer Vision and Pattern Recognition (pp. 6877-6885), 2018.

[15] S. Smith, "MATLAB: advanced GUI development," Dog Ear Publishing, Indianapolis, IN, 2006.

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.*

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.*

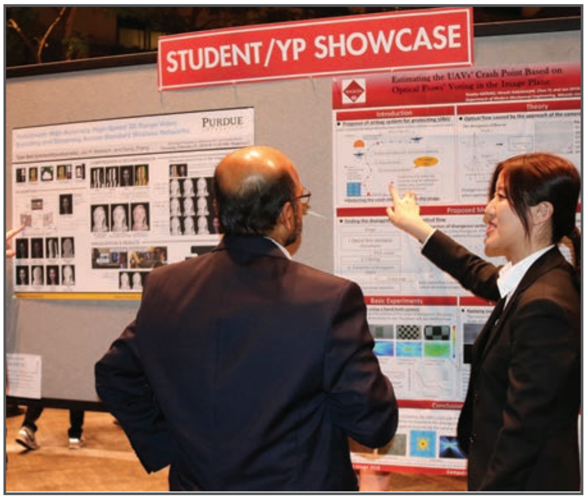
JOIN US AT THE NEXT EI!

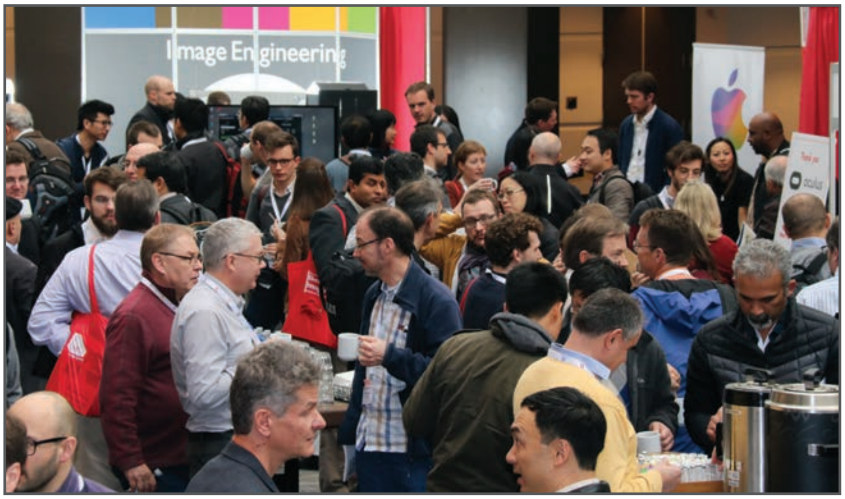
IS&T International Symposium on

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!

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

www.electronicimaging.org

IS&T
imaging.org