10.2352/EI.2022.34.15.COLOR-279

Improving an inkjet printer: removing stray-dots by constraining error diffusion in highlight regions*

***Sige Hu*[1], *George Chiu*[1], *Davi He*[2], *White He*[2], *Rain Guo*[2], *Jan Allebach*[1]**
[1] ***Purdue University, West Lafayette, IN 47906, U.S.A***
[2] ***Sunvalley Tek, Shenzhen, China***

Abstract

In our previous paper [1] presented in the last year, we mostly focused on the color saturation problem in our inkjet printer. However, our partner reported that there are some boundary noise pixels on the background, which are quite visible when the background is white. By checking the pipeline of our printing procedure, we realized that the noise stray dots are generated during the halftoning procedure. This paper is dedicated to separate the white background from the foreground, which enables us to constrain the error diffusion process inside the white background. The main idea is to apply image segmentation, which could help us to precisely extract the background.

Introduction

To evaluate the quality of a print, color is obviously one of the most important aspect to consider, which is also what we concentrated on during the past two years. In one of our previous papers, we implemented a color management method [2] in our printer, which maps the source/image gamut into our destination/printer gamut. We also implemented a color saturation method [1] to increase the saturation of our prints, which makes our printing results more vivid.

But besides color, noise pixels are also one of the important aspects that needs to be considered when we investigate our printing results. We notice that there are a lots of noise pixels on the background next to the main object, which makes the prints to be not as smooth as we expect, and the overall beauty is highly affected. Since the noise stray dots are generated during the halftoning procedure, during which the error is diffused from the main objects, we decide to constrain the error diffusion in highlight regions. Segmentation is obviously the most straightforward method we could use to extract the background/highlight regions from the image.

In this paper, we will introduce several segmentation methods in three sections. The first section will mainly focus on a K-means algorithm based error diffusion map. The second section illustrates a pixel thresholding based error diffusion map. And the last section is an edge based error diffusion map. The error diffusion map is a binary map, in which we expect to assign all the white background pixels to value 0 and the foreground pixels to value 1. Therefore, when we perform our error diffusion, the background white pixels will not participate in the process and stay as pure white.

*Research supported by Sunvalley Tek, Shenzhen, CHINA.

Method and Analysis

Overview of the boundary stray dots

The stray dots mostly occur on the boundary of the foreground objects, which is shown in Fig. 1. The left part of Fig. 1 is a digital halftone result of the bread image with a red bounding box that covers the boundary of the bread slice, and the right part of Fig. 1 is the zoom-in version of the red bounding box region. We can clearly observe that there are many noise pixels on the white background. And we want to increase the image quality by minimizing the effect of those noise pixels.

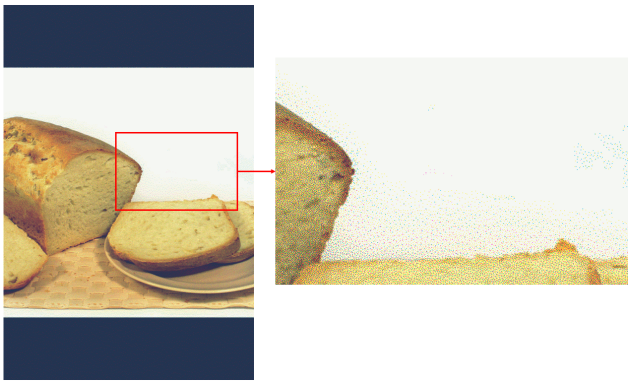


Figure 1. *Zoom in on the digital halftone image at the boundary region to show the stray dots.*

K-means algorithm based error diffusion map

The error diffusion map, as we mentioned in the introduction, is a binary map with the same size as our input digital image, which determines whether the pixel join the error diffusion procedure or not. The first method that comes to our mind is the K-means clustering algorithm [3], which is a well-known segmentation algorithm, which we hope can help us to extract the white background regions.

The procedure of this approach is straightforward. We first perform the standard naive K-means algorithm by fixing 5 clusters in the YyCxCz color space [4]. We decide to use the YyCxCz color space because it is both device-independent and more perceptually uniform. After the K-means algorithm, we have generated a 5-clusters map. For each cluster, we calculate the average lightness, the average chroma, and the standard deviation by using Eq. 1 - Eq. 3. The Cl_n represents the cluster set, where in our case n can be varied from 1 to 5 according to the fixed number of clusters, which is 5.

$$\text{Chroma} = \sqrt{C_x^2 + C_z^2} = c \qquad \text{Lightness} = Y_y \tag{1}$$

$$\begin{aligned} \text{Average lightness} &= \frac{\sum_{p \in Cl_n} \text{lightness}}{\text{Number of pixels in } Cl_n} \\ \text{Average chroma} &= \frac{\sum_{p \in Cl_n} \text{chroma}}{\text{Number of pixels in } Cl_n} \end{aligned} \tag{2}$$

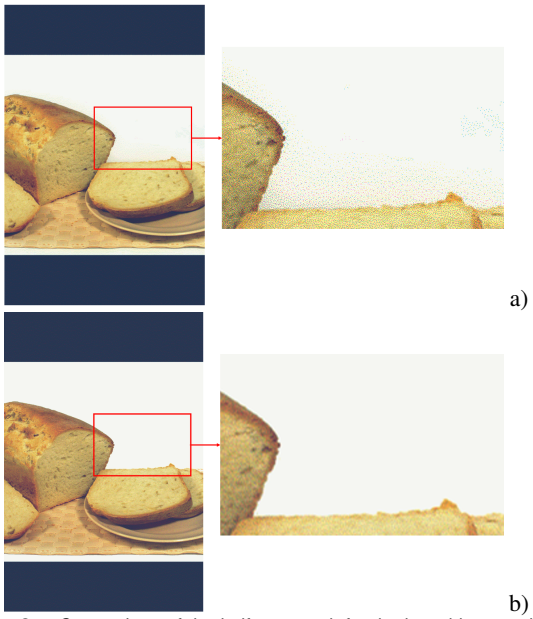
$$\text{Standard deviation} = \sqrt{\frac{\sum_{p \in Cl_n} (\text{lightness} - \text{average lightness})^2}{\text{Number of pixels in } Cl_n}} \tag{3}$$

Once we obtain all this information for each cluster, we can generate the error diffusion map by some self-defined criteria. If the cluster satisfies all of the following three conditions, all pixels in that cluster are set to the value zero, which means we consider that cluster as the white background cluster. Otherwise, we set all pixels in the cluster to the value one. The three conditions basically depend on the lightness threshold, the chroma threshold, and the standard deviation threshold. Since we want to take out the white background, those clusters should satisfy the criteria of the high lightness and the low chroma. Meanwhile, we want to pick those low standard deviation clusters because white background regions are usually smooth in texture. By experimenting with several different sets of criteria, we find that the thresholds within a region perform similarly and we pick one set that performs a little better. For this set of criteria, the cluster needs to satisfy the conditions that the average lightness is greater than the lightness threshold 85; the average chroma is smaller than the chroma threshold 15; and the standard deviation is smaller than the threshold 5 to be considered as the white background cluster. After constraining the K-means map by the set of thresholds, we acquire an error diffusion map. We still use the bread image as an example, as shown in Fig. 2 b), where the error diffusion map is represented as a black and white image where the black color represents the white background, and the foreground is represented by the white color.

a) b)

Figure 2. *The error diffusion map of the bread image based on a K-means map: a) Original input image. b) Error diffusion map.*

After implementing the error diffusion map in our halftoning pipeline, we obtain a new halftone result, which we compare to the original halftone result in Fig. 3. Fig. 3a) shows the original halftone result, where we can clearly notice the boundary noise pixels around the slice of bread. However, in Fig. 3b), with the implementation of the K-means algorithm based error diffusion map, the noise pixels are not generated during our error diffusion procedure, which manifests a much smoother digital halftone result.



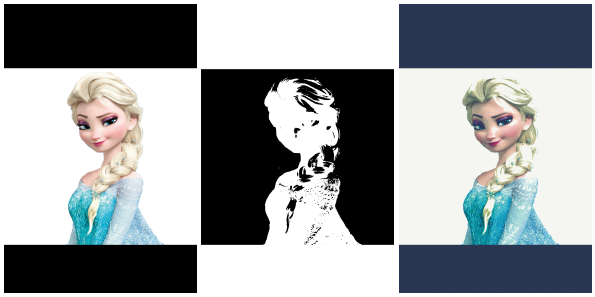
a)

b)

Figure 3. *Comparison of the halftone result for the bread image with the K-means error diffusion map: a) Original digital halftone result. b) Digital halftone result with error diffusion map.*

The K-means based method performs incredibly well in those simple texture images like the bread image while it also encounters some difficulties. Firstly, just like the standard naive K-means algorithm, this method costs hugely in time. Our nail printer is expected to print fast and precisely, so we prefer either to use a less time consuming algorithm or to pre-process the image before printing. Secondly, the number of clusters also varies the result of our error diffusion map when we process more complex texture images. For example, as shown in the Elsa image in Fig. 4, her hair color is light blond, which is close to the white background when we perform K-means clustering. This similarity results in the merger of the background and the foreground, and causes some discontinuity problems in her hair and shoulder region.

To solve the discontinuity problems, we first increase the number of clusters, but it is still hard to reasonably extract the background. Meanwhile, as an unfavorable trade-off, this also decreases the performance with images that have a simpler foreground and background. We then come up with a two pass K-means based method. After generating the first version of the error diffusion map based on K-means information only, for each row, we count the length of each run of background pixels. If the length smaller than the threshold (Image width/10), we set the background pixels back to the foreground. To compare with the



a) b) c)

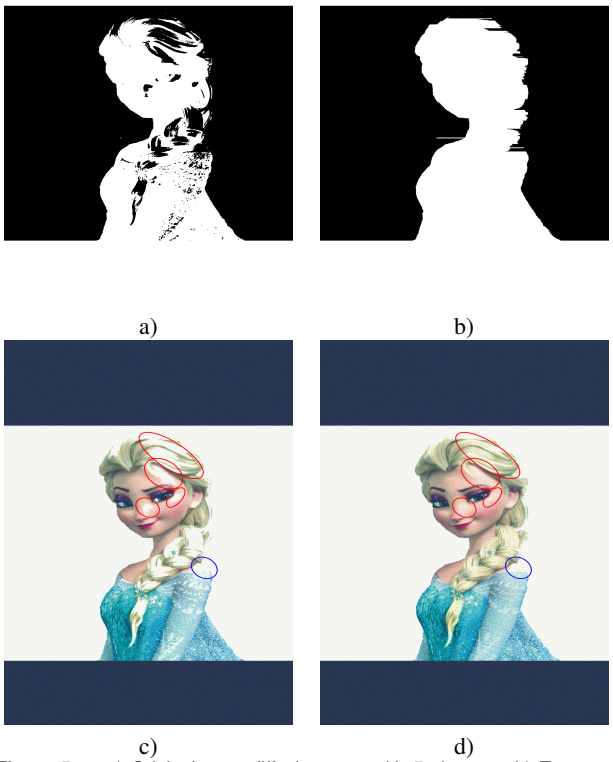
Figure 4. *a) Original Elsa image. b) K-means based error diffusion map c) Halftone result with error diffusion map.*

previous one pass method, we exhibit the error diffusion maps and the halftone results for both methods in Fig. 5. The two pass method does alleviate the problem in some regions, for example, the cheek and the hair of Elsa, where we use the red circles for better visualization. However, the two pass method cannot solve the discontinuity issue with those regions that are not bounded by at least a single foreground object pixel, as the blue circle region emphasizes. Furthermore, the two pass method can be easily trapped by noisy foreground pixel inside the background region. Therefore, we abandon the two pass method and decide to generate the error diffusion map by two other approaches, which will be illustrated in the next two sections.

Pixel thresholding based error diffusion map

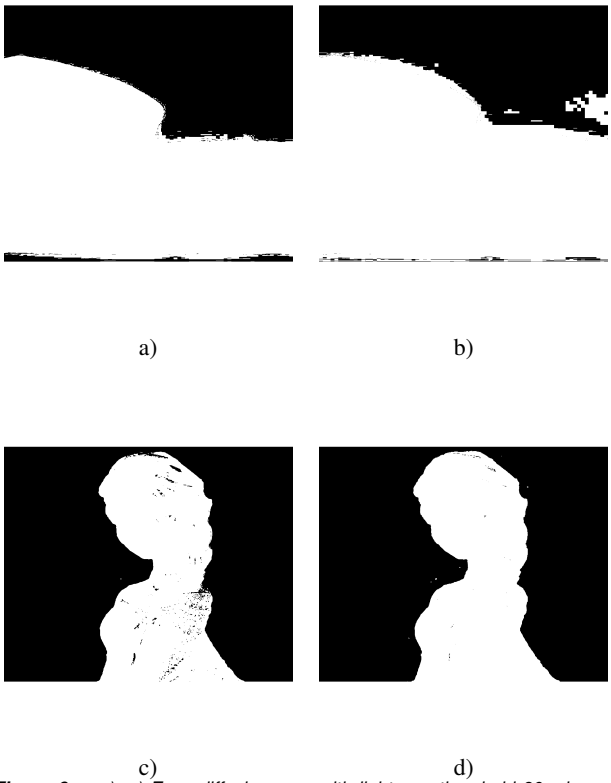
Instead of performing K-means algorithm, we attempt to perform a pixel by pixel thresholding algorithm. This method is intended to decrease the time expense while maintaining the quality of the halftone result in contrast to the K-means based algorithm. For this method, we directly compare the lightness, and the chroma to corresponding thresholds. If the pixel has larger lightness and lower chroma compared to the threshold, we consider it as a background pixel.

Even though the method does reduce the time cost, the results of the error diffusion map are much more uneven on the boundary. We experimented with several sets of thresholds, and we show two sets of them in Fig. 6. The left two error diffusion maps (Fig. 6 a) and c)) are both obtained by using the lightness threshold 80 and chroma threshold 15 while the right two error diffusion maps (Fig. 6 b) and d)) are using lightness threshold 90 and chroma threshold 15. Similar to what we observe with the K-means method, different sets of threshold values dramatically vary our results. With the lower lightness threshold 80, the simple structure bread image generates a relatively complete, but spotty error diffusion map. But the Elsa image still struggles on the discontinuity with the lower lightness threshold 80. However, on the other hand, with the higher lightness threshold 90, the Elsa image generates an incredibly smooth map while the bread image performs poorly on the edge and seems not to be able to identify noise pixels.



a) b) c) d)

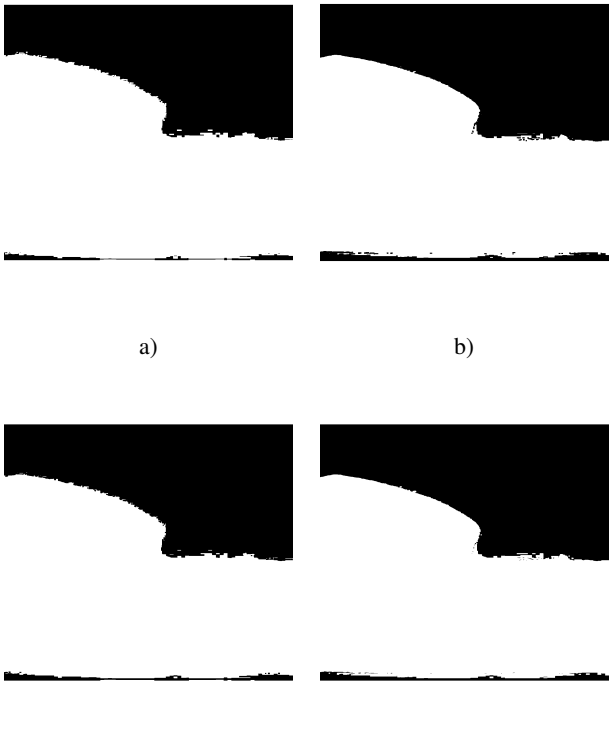
Figure 5. *a) Original error diffusion map with 5 clusters. b) Two pass error diffusion map with 5 clusters. c) Halftone result with the original error diffusion map. d) Halftone result with the two pass error diffusion map.*



a) b) c) d)

Figure 6. *a), c) Error diffusion map with lightness threshold 80, chroma threshold 15. b), d) Error diffusion map with lightness threshold 90, chroma threshold 15.*

To handle the spotty and noise boundary pixels on the error diffusion map, we apply a morphology operations [5] to the pixel thresholding based error diffusion map. Image morphology is a technique to remove the noise pixels in a binary image by accounting for the form and structure of the image. There are two basic image morphology operations, which are image dilation and image erosion; and by combining these two basic operations, there are also image closing and image opening. However, none four operations construct the smooth error diffusion map that we expect. Results of the four operations with the structuring element of a 5×5 square are shown in Fig. 7. Hence, even though the pixel thresholding based error diffusion map spends less time, the quality does not satisfy our requirement.



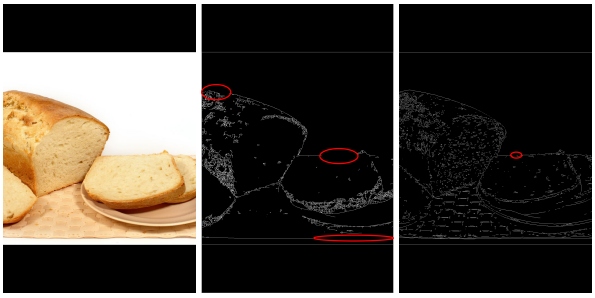
a) b) c) d)

Figure 7. *a) Error diffusion map after image dilation. b) Error diffusion map after image erosion c)Error diffusion map after image closing. d) Error diffusion map after image opening.*

Edge based error diffusion map

This method first generates a pixel thresholding based error diffusion map, which we described in the previous section. We then generate an edge map to segment the image, and we count the number of background pixels of each segment. Finally, we combine two maps according to the criteria that if there are more background pixels inside the segment, we consider the segment as the background segment.

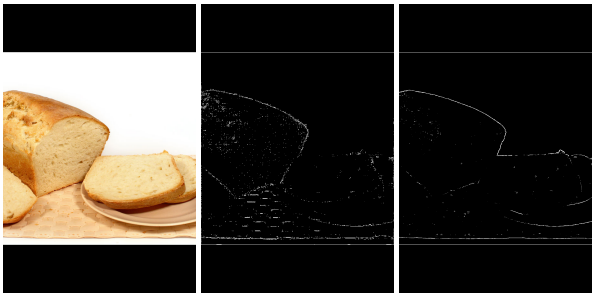
We investigate the performance of several edge detection algorithms to find the best solution. We first compare the result of the Sobel edge detector with edge thinning [6], [7] to the result of Canny edge detector [8] in Fig. 8. Both methods outline the boundary of the bread, but both methods also encounter the missing edge problem as indicated by the red circles. The missing edges are not allowed in our edge based method because they will merge the background and the foreground pixels, and consequently deceive the judgement when we combine the error diffusion map and the edge map.



a) b) c)

Figure 8. *a) Original image. b) Sobel edge detector with edge thinning. c) Canny edge detector.*

Subsequently, we experiment with two other methods, which are the Laplacian of Gaussian (LoG) [9] and the simple delta E edge method. The Laplacian is a 2-D isotropic measure of the second spatial derivative of an image, and when we apply the Laplacian to an image that has first been smoothed with a Gaussian smoothing filter in order to reduce its sensitivity to noise, we call the method the Laplacian of Gaussian (LoG). The simple delta E edge method is a method based on CIE $L^*a^*b^*$ color space. We first find the $L^*a^*b^*$ value of each pixel, and we define the white background $L^*a^*b^*$ as our target $L^*a^*b^*$. We compute the ΔE $L^*a^*b^*$ of each pixel by comparing its $L^*a^*b^*$ value with the target $L^*a^*b^*$. We consider pixels where the left and right ΔE are both larger than the threshold, or where the up and bottom ΔE are both larger than the threshold as our edge pixels. The edge map of those two methods are shown in Fig. 9. The LoG performs horribly on the bread slice boundary. While the simple delta E edge method, even though it exceeds our expectation, it still experiences the missing edge issue.



a) b) c)

Figure 9. *a) Original image. b) LoG edge detector. c) ΔE edge detector.*

Since all those edge detection methods cannot completely depict the contour of the input image, we start to consider an edge linking algorithm, and we eventually pick the Ant Colony Optimization (ACO) algorithm [10]. This algorithm first extracts endpoints from the Canny edge detector, and those end points are used as the initial starting position of virtual ants. The missing

pixels edges are iteratively retraced through the movements of the ants depositing pheromone trails. This pheromone quantity is updated locally after each ant moves and globally after all the ants' movements.

During the preparation stage, we need to record the endpoint type by where the edge comes from. These are N, S, W, E, NW, NE, SW, SE. The Heuristic matrix w, which represents probabilistic direction bias values is shown in Table 1. This is the example when the endpoint type is S or more precisely the ant is coming from south. We just need to rotate this Heuristic matrix w if the endpoint type is different. For example, if the end point type is W, where the is ant coming from west, we just need to clockwise rotate the Heuristic matrix w shown in Table 1 by 90 degrees. We also need to calculate the gradient magnitude variation map V_G based on the gradient magnitude map $g_{L^*a^*b^*}$. The gradient magnitude map $g_{L^*a^*b^*}$ is calculated by Eq. 4 where the g_{L^*}, g_{a^*}, and g_{b^*} are CIE $L^*a^*b^*$ values. The gradient magnitude variation map V_G is determined by Eq. 5.

Heuristic matrix w

1/2	1	1/2
1/4	ANT	1/4
1/12	1/20	1/12

$$g_{L^*a^*b^*}(i,j) = \sqrt{(g_{L^*}[i][j])^2 + (g_{a^*}[i][j])^2 + (g_{b^*}[i][j])^2} \quad (4)$$

$$V_G(i,j) = max \begin{bmatrix} |g_{L^*a^*b^*}[i-1][j-1] - g_{L^*a^*b^*}[i+1][j+1]|, \\ |g_{L^*a^*b^*}[i-1][j] - g_{L^*a^*b^*}[i+1][j]|, \\ |g_{L^*a^*b^*}[i-1][j+1] - g_{L^*a^*b^*}[i+1][j-1]|, \\ |g_{L^*a^*b^*}[i][j-1] - g_{L^*a^*b^*}[i][j+1]|, \end{bmatrix} \quad (5)$$

At the main stage, we first initialize the pheromone trail by Eq. 6, where $V_{G,max}$ is the maximum value of the gradient magnitude variation map V_G. We predict the Probability Transition Matrix, which specifies the probability for an ant's movements from one pixel to another. For the n^{th} movement at the k^{th} endpoint, this matrix is determined by Eq. 7 where Ω_k is the set of neighborhood nodes of the current node, and the two parameters α and β are two constants characterizing the influence of the pheromone matrix and the heuristic matrix, respectively. Among the 8 neighbors, we choose the largest probability position to move the ant. We update the pheromone matrix τ following the rule in [10], and we stop the ant once it touch as an existing edge point. The result of the ACO edge linking method applied on the bread image is shown in Fig. 10. As we can see in the zoom in version of the result, the ACO method assists us to connect the missing edges.

$$\tau_0(i,j) = \frac{V_G(i,j)}{V_{G,max}} \quad (6)$$

$$P^n_{(i,j)} = \frac{(\tau^{n-1}_{(i,j)})^\alpha w^\beta}{\sum_{(i,j)\in\Omega_k}(\tau^{n-1}_{(i,j)})^\alpha w^\beta} \quad (7)$$

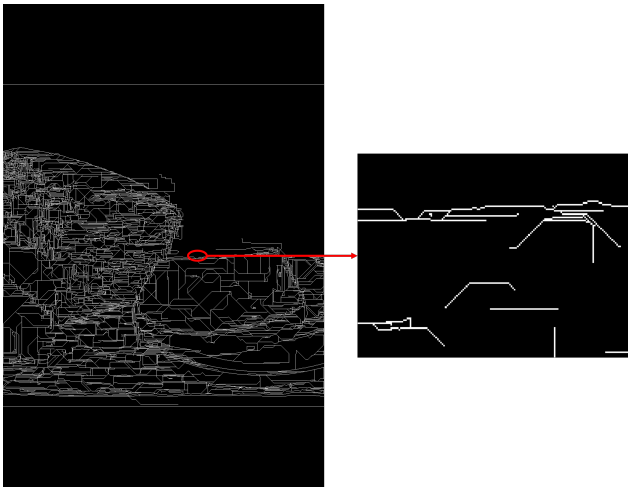


Figure 10. *Zoom in on the bread edge image after applying ACO.*

This method also works well on part of a complex image, such as the shoulder and the hair of the Elsa image. However, limited by the ACO algorithm, this method cannot deal with isolated pixels in the initial edge map since it cannot determine the endpoint type for those pixels. As shown in Fig. 11, on the right side of the chin of Elsa, those edge points are all isolated because ACO cannot link them together, which results in the merger of the right side of her chin with the background pixels on the final error diffusion map that is shown in Fig. 12.

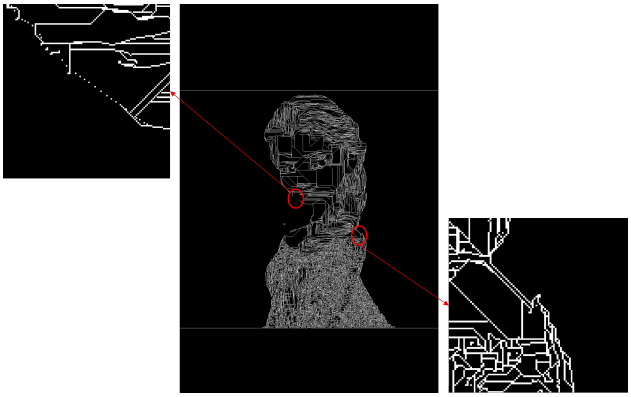


Figure 11. *Zoom in on the Elsa edge image after applying ACO.*

Conclusion

In this paper, we attempt to remove the boundary stray dots by constraining the error diffusion process to avoid the white background. We seek to solve the problem by using three different approaches. None of the three methods is able to perfectly extract the background as we would like. The K-means algorithm is computationally expensive and can be misled by complex texture or a light color region. The thresholding-based algorithm is simple and straightforward, but it contains so many noisy regions and performs unsuccessfully on sharp edges. The combination of a threshold-based algorithm and edge detection is restricted to the result of the edge map. For the future work, we might do research on an image-dependent K-means algorithm by varying the num-

Figure 12. *Error diffusion map of the Elsa edge image after applying ACO.*

ber of clusters with different images, and meanwhile reducing the computational expense of the K-means based method.

References

[1] Hu, Sige, et al. "Improving an inkjet printer: saturation enhancement based on segmentation and hue." Electronic Imaging 2021.16 (2021): 328-1.

[2] Wang, Yin, et al. "Developing an inkjet printer I: RGB image to CMY ink amounts-Image processing and color management." Electronic Imaging 2020.15 (2020): 350-1.

[3] MacQueen, James. "Some methods for classification and analysis of multivariate observations." Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics And Probability. Vol. 1. No. 14. 1967.

[4] Flohr, Thomas J., et al. "Model-based color image quantization." Human Vision, Visual Processing, and Digital Display IV. Vol. 1913. International Society for Optics and Photonics, 1993.

[5] Haralick, Robert M., Stanley R. Sternberg, and Xinhua Zhuang. "Image analysis using mathematical morphology." IEEE Transactions on Pattern Analysis And Machine Intelligence 4 (1987): 532-550.

[6] Seger, Olle. "Generalized and separable Sobel operators." Machine Vision For Three-Dimensional Scenes (2012): 347.

[7] Zhang, T. Y., and Ching Y. Suen. "A fast parallel algorithm for thinning digital patterns." Communications of the ACM 27.3 (1984): 236-239.

[8] Canny, John. "A computational approach to edge detection." IEEE Transactions on Pattern Analysis And Machine Intelligence 6 (1986): 679-698.

[9] Umbaugh, Scott E. Digital Image Processing and Analysis: Applications with MATLAB® and CVIPtools. CRC press, 2017.

[10] Benhamza, Karima, and Hamid Seridi. "Canny edge detector improvement using an intelligent ants routing." Evolving Systems 12.2 (2021): 397-406.

Author Biography

Sige Hu *received his BS in Electrical and Computer Engineering from Pennsylvania State University (2017) and is currently pursuing PhD in Electrical and Computer Engineering at Purdue University. His research is mainly focused on image processing.*

George T. Chiu *is a Professor in the School of Mechanical Engineering with courtesy appointments in the School of Electrical and Computer Engineering and the Department of Psychological Sciences at Purdue University. He also serves as the Assistant Dean for Global Engineering Programs and Partnership for the College of Engineering. Dr. Chiu received the B.S. degree in Mechanical Engineering from the National Taiwan University in 1985 and the M.S. and Ph.D. degrees in Mechanical Engineering from the University of California at Berkeley, in 1990 and 1994, respectively. From September 2011 to June 2014, he served as the Program Director for the Control Systems Program at the National Science Foundation. His current research interests are mechatronics and dynamic systems and control with applications to digital printing and imaging systems, digital fabrications and functional printing, human motor control, motion and vibration perception and control. He received the 2012 NSF Director's Collaboration Award and the 2010 IEEE Transactions on Control System Technology Outstanding Paper Award. He served as the Editor-in-Chief for the IEEE/ASME Transactions on Mechatronics from 2017-19 and as the Editor for the Journal of Imaging Science and Technology from 2012-14. Dr. Chiu served on the Executive Committee of the ASME Dynamic Systems and Control Division (DSCD) from 2007 to 2014 and as the Chair of the Division from 2012-13. He is a Fellow of ASME and a Fellow of the Society for Imaging Science and Technology (IS&T).*

Peng (Davi) He *is a software manager in Sunvalleytek International Inc, Shenzhen, Guangdong, China. He received his B.S in Communication Engineering from Hunan University of Arts and Science, Changde, Hunan, China in 2012.*

Wengui (White) He *is a Senior Director of Embedded Software Team in Sunvalleytek International Inc, Shenzhen, Guangdong, China. He received his B.S in Communication Engineering from Hunan University of Technology, Zhuzhou, Hunan, China in 2011.*

Qian (Rain) Guo *is an algorithm engineer in Sunvalleytek International Inc, Shenzhen, Guangdong, China. She received her BS in Light Chemical Engineering from Qufu Normal University, Rizhao, Shandong, China in 2013 and MS in Light Chemical Engineering from South China University of Technology, Guangzhou, Guangdong, China in 2016.*

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, the IS&T/OSA Edwin Land Medal, the IS&T Johann Gutenberg Prize, and is a member of the National Academy of Engineering*