

Inverse Halftoning Based on Edge Detection Classification

Qi-Xuan Ong and Wen-Liang Hsue

Department of Electrical Engineering, Chung Yuan Christian University, Chung-Li, Taiwan, R.O.C.

E-mail: wlhsue@cycu.edu.tw Tel: +886-3-2654832

Abstract—Inverse halftoning is a technique to reconstruct gray-level images from halftone images. Since image halftoning process results in information loss, inverse halftoning cannot reconstruct perfectly original gray-level images from corresponding halftone images. Consequently, several inverse halftoning methods were proposed, e.g., LIH and ELIH [1]-[2], etc. In this paper, we will first review an existing inverse halftoning technique with variance classified filtering [3]. We will replace LMS (least-mean-square) algorithm by the LS (least-square) algorithm to improve the training stage for variance classified inverse halftoning in [3]. Then we will use edge detection to classify image data instead of variance used in [3]. Experiment results show that both LS filtering and edge detection classification proposed in this paper enhance quality of output gray-level images for inverse halftoning.

I. INTRODUCTION

The continuous-tone digital images are gray-level images, which have 256 tones. Since human visual system cannot distinguish the gray-level images and analog images, we call the gray-level images as the continuous tone images [4]. Since image halftoning technique is widely used in printing [5], we need to know the meaning of the dpi (dots per inch). The resolution of human visual system is about 300dpi [6], for which the size of a pixel is 1/300 inch. However, resolution of printers is usually much more than 300 dpi. The halftoning technique is to use the resolution of the space to exchange for the resolution of the brightness. Therefore, printers usually have high resolution in thousands of dpi.

Since most of printers and low-level monitors can only display binary black-and-white images, we need to change the gray-level images to the halftone images in order that they can be displayed in those printers and monitors. The halftone technique represents brightness by density of black points or white points in a small region. According to the algorithms of assigning the black and white output pixels in halftone images from gray-level images, there are various existing halftoning techniques, the ordered dither [7]-[8], error diffusion [9], and dot diffusion [10], etc.

II. INVERSE HALFTONING WITH FILTERING

A. Inverse Halftoning with the LMS Filter

The LMS (least-mean-square) adaptive filter [11] automatically adjusts its filter coefficients according to input data. In this subsection, we will briefly review how the LMS filter can be trained and its mathematical principle. Assume

the filter coefficients to be trained is $w_{m,n}$, where m, n are the row number and column number of the coefficients respectively, and the size of the filter is $M \times N$. If the binary pixel value of the halftone image is $b_{i,j}$ and its corresponding reconstructed gray-level value is $\hat{g}_{i,j}$, where i and j denote the position of the halftone image and the reconstructed gray-level image, then we can reconstruct the gray-level image by

$$\hat{g}_{i,j} = \sum_{m=1}^M \sum_{n=1}^N w_{m,n} \times b_{i+m-\lfloor \frac{M}{2} \rfloor, j+n-\lfloor \frac{N}{2} \rfloor}. \quad (1)$$

We then apply the reconstructed pixel value to update coefficients of the filter. First, error of the filter output with respect to the original gray-level image pixel $g_{i,j}$ is:

$$e_{i,j}^2 = (g_{i,j} - \hat{g}_{i,j})^2, \quad (2)$$

$$e_{i,j}^2 = \left(g_{i,j} - \sum_{m=1}^M \sum_{n=1}^N w_{m,n} \times b_{i+m-\lfloor \frac{M}{2} \rfloor, j+n-\lfloor \frac{N}{2} \rfloor} \right)^2. \quad (3)$$

Then the partial derivative of (3) with respect to $w_{m,n}$ is:

$$\frac{\partial e_{i,j}^2}{\partial w_{m,n}} = -2e_{i,j} \times b_{i+m-\lfloor \frac{M}{2} \rfloor, j+n-\lfloor \frac{N}{2} \rfloor}. \quad (4)$$

The partial derivative means the slope. When the slope equals to 0, the error between the reconstructed image and the original gray-level image is minimal. Therefore, the coefficient $w_{m,n}$ is optimal when the partial derivative of (4) is equal to 0. If the slope is larger than 0, and then $w_{m,n}$ is larger than the optimal value and we subtract it by a small value. If the slope is less than 0, then $w_{m,n}$ is less than the optimal value and we add it by a small value. As a result, we can adjust the LMS coefficients by

$$w_{m,n}(k+1) = w_{m,n}(k) + \mu \times e_{i,j} \times b_{i+m-\lfloor \frac{M}{2} \rfloor, j+n-\lfloor \frac{N}{2} \rfloor}, \quad (5)$$

where μ is the step size related to the convergent speed. In our simulations, μ is chosen as 10^{-5} .

B. Inverse Halftoning with Variance Classified Filtering

Before filtering in the inverse halftoning process, if we first classify the halftone images according to their properties, then we can produce better quality of the inverse halftoned images. To find the appropriate method to classify the halftone image data is one of our subjects in this paper. Here we review one existing method to classify the halftone image data proposed in [3].

First, decide the size of the mask, e.g., 7×7 . Then apply the mask, the training image and its halftone image to LMS algorithm to train the coefficient matrix w . After that, apply w to produce the rough inverse halftone image I by the following:

$$I_{i,j} = \sum_{m=1}^M \sum_{n=1}^N w_{m,n} \times b_{i+m-\lfloor \frac{M}{2} \rfloor, j+n-\lfloor \frac{N}{2} \rfloor}, \quad (6)$$

where $b_{i+m-\lfloor \frac{M}{2} \rfloor, j+n-\lfloor \frac{N}{2} \rfloor}$ is the pixel value of halftone image.

We classify the 7×7 blocks of $b_{i,j}$ from rough inverse halftone image $I_{i,j}$ which are covered by the mask according to their variances. The variance can be computed as following

$$V = \frac{1}{M \times N} \sum_{m=1}^M \sum_{n=1}^N \left(I_{i,j} - \frac{\sum_{m=1}^M \sum_{n=1}^N I_{i+m-\lfloor \frac{M}{2} \rfloor, j+n-\lfloor \frac{N}{2} \rfloor}}{M \times N} \right)^2. \quad (7)$$

We can sort those blocks of the halftone image and their corresponding pixel values in the training image according the values of V , and we assume this set is A . One simple way to classify the set A is to directly separate A into subsets with equal elements, and then train the filter coefficients for each subset. Another method of classification we want to review is maximum filters difference guidance (MFDG) [3]. First we equally separate three subsets from the set A , and training the 3 filters $w_{m,n}^{(k)}$ respectively, where (k) denotes the k -th filter. Next, to find the difference of the coefficients of filters between each two adjacent sets, we have

$$DIF^{(k)} = \sum_{m=1}^M \sum_{n=1}^N |w_{m,n}^{(k)} - w_{m,n}^{(k+1)}|, \quad k=1,2,\dots,T-1, \quad (8)$$

where T is the number of the current filters. Therefore, the initial value of T is three. We then find the maximum difference $\max(DIF^{(k)})$ between each two adjacent sets and combine the two adjacent set, then equally separate 3 sets from the combined two set with maximum difference and train 3 filters for this combined set. Finally, repeat the steps above until we have the number of training filters desired.

III. LS FILTER AND EDGE DETECTION CLASSIFICATION

A. Inverse Halftoning with the LS filter

When the LMS filter is convergent, its coefficients are not optimal, but oscillate around the optimal values [12]. Instead of the LMS filter, we can apply the LS (least square) filter to find the filter optimal coefficients. The LS filter is widely applied for original signal estimation [12].

If b is the $M \times N$ halftone image pixels under a mask, then we permute b column by column to a vector. We can find the optimal LS filter by the following. Let the $M \cdot N \times M \cdot N$ autocorrelation matrix be

$$\phi = E(XX^T) \quad (9)$$

where $X = [b_{1,1}, b_{2,1}, \dots, b_{M,1}, b_{1,2}, b_{2,2}, \dots, b_{M,2}, \dots, b_{M,N}]^T$ and

$$z = E(Xy) \quad (10)$$

where z is the cross-correlation vector of X and the current original gray-level pixel value y . From statistics, we have the $MN \times 1$ optimal LS coefficients

$$W = \left(\frac{\sum_{t=1}^T \phi_t}{T} \right)^{-1} \left(\frac{\sum_{t=1}^T z_t}{T} \right) \quad (11)$$

where ϕ_t and z_t are the t -th autocorrelation matrix and cross-correlation vector, T is the number of all data samples.

B. LS Filter and Variance Classified Filtering

In [3], Guo apply the LMS algorithm for the classified training filter to find the coefficients of the filter. Here we propose applying the LS algorithm for the filter to find the coefficients, and we find that the coefficients of the filter are much suitable for inverse halftoning. For this purpose, we also need the rough inverse halftoned images for variance calculations of image blocks. The rough inverse halftoned images can be produced by the LMS algorithm or the LS algorithm. In our experiments, no matter when we apply the LS algorithm for classified filter trainings or for the rough inverse halftoned images, the LS algorithm provides much more correct information than the LMS for us.

From (5), the LMS algorithm adjusts the coefficients of the filter every time a new image block is available. It means that, the coefficients of filter are aimed to the current block of the training image to adjust. If the optimal filter for the latest blocks of training image is different from most of other blocks, then the filter is not the optimal filter for all of the blocks. On the other hand, from (11), the LS filter coefficients are determined once the training images are selected. Unlike LMS, the sequence orders of training blocks do not affect LS filter coefficients.

The halftone images we want to reconstruct and the training images are two independent sets, test image set and training image set. In this paper, we use 39 images to train the filter, and use 12 test images which are different from the training images to reconstruct their gray-level images. We analyze the quality by PSNR (peak signal to noise ratio) between the reconstructed image and the original gray-level image of the halftone.

To improve quality of the reconstructed gray-level images of the halftones, we choose low-frequency images to train the filter since halftones of these training images can represent the gray-level images well. To test the performance of the filter, the 12 test images are not limited to the low-frequency images.

C. Edge Detection Classification

In [3], Guo uses one LMS filter to generate a rough inverse halftoned image, then the training image blocks are classified by their variances for training filter coefficients as reviewed in Section II. However, it is found from experiments that output reconstructed gray-level images of Guo's inverse halftoning method [3] have larger errors near edges. Consequently, we propose to classify image blocks of rough inverse halftoned images according to their edge information to enhance quality of inverse halftoned gray-level images, especially near edges.

Here we use the Sobel discrete gradient operator to compute horizontal and vertical discrete first derivatives of images as well as direction of the gradient. Then we use these properties to classify blocks of rough inverse halftoned images I . The details are explained as follows. Let

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad (12)$$

$$G_x = \sum_{m=1}^3 \sum_{n=1}^3 S_{x,(m,n)} \times I_{i+m-2, j+n-2}, \quad (13)$$

$$G_y = \sum_{m=1}^3 \sum_{n=1}^3 S_{y,(m,n)} \times I_{i+m-2, j+n-2}, \quad (14)$$

$$G' = \sqrt{G_x^2 + G_y^2}, \quad \theta = \tan^{-1} \left(\frac{G_x}{G_y} \right), \quad (15)$$

where G_x is the horizontal derivative, G_y is the vertical derivative, and G' is the magnitude of the Sobel gradient operator. In (15), θ is the direction angle of the Sobel gradient. We then classify image blocks using G' and θ . Because of the Sobel coefficients, maximum values for magnitudes of G_x and G_y are both $255 \times 4 = 1020$. From experimental experiences, we partition values of G' into 12 intervals, with thresholds being 20, 60, 100, 140, 180, 220, 260, 300, 360, 420, and 480. As an example, $G' = 80$ is in the interval between 60 and 100. For each interval of G' , we further classify image blocks according to their direction of gradient θ , with 10 degrees for each group. However, when G' is less than 20, they are classified as the same subset whatever their values of θ are. The subset for these image blocks corresponds to the lowest frequency blocks. In order to save training time, we use an upper bound for number of image blocks in each subset, which is 50000. 50000 image blocks are enough for training filter coefficients. Consequently, with our method, we compute 397 different 7×7 filter coefficients, which are saved for inverse halftoning, from 397 different subsets of image blocks. To reconstruct higher quality inverse halftoned gray-level images, we choose filter coefficients according to output of Sobel gradient operator for image blocks of rough inverse halftoned images. It should be noted that we also need to save the LMS or LS filter coefficients computed for the rough inverse halftoned images. After edge detection classification, we can choose LMS or LS algorithm to compute filter coefficients for subset of classified image blocks. In the next section, their respective performances will be compared.

IV. EXPERIMENTAL RESULTS

To compare the performances of inverse halftoning methods, Fig. 1 shows one of the test images. Fig. 2 is the halftone image of Fig. 1 produced by the Floyd-Steinberg kernel. Fig. 3 is reconstructed gray-level image of Fig. 2 generated by variance classification and the 397 LMS training

filters. Fig. 4 is the reconstructed image of Fig. 2 produced by the edge detection classification and 397 LS training filters. The PSNR of Fig. 3 is 31.57 dB, and that of Fig. 4 is 32.62 dB. Therefore, the performance of edge detection classification LS filters for the test image is better.



Fig. 1. The test image.

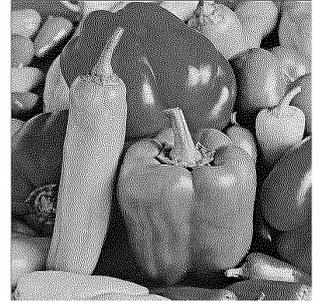


Fig. 2. The halftone image of Fig. 1 produced by error diffusion.



Fig. 3. The inverse halftone image of Fig. 2 produced by variance classified LMS filtering.

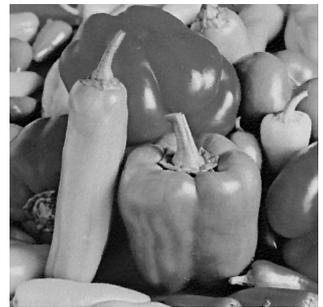


Fig. 4. The inverse halftone image of Fig. 2 produced by edge detection classified LS filtering.

Table I shows PSNRs of the reconstructed images which are produced by the LMS training filter, LS training filter, variance classified LMS training filter, variance classified LS training filter, edge detection classified LMS training filter, and edge detection classified LS training filter. The variance classified filters all apply the MFDG to classify filters. For comparison, all kinds of the classified filters use 397 filters, and halftone images are produced by the Floyd-Steinberg kernel. The 12 test images are shown in Fig. 5, which have different properties. The low-frequency images have better output PSNR quality. Because halftone images lose a lot of high-frequency information of the gray-level images, we cannot reconstruct high-frequency part of the gray-level images with inverse halftoning. From Table I and Fig. 5, we can verify this observation.

From the last row of Table I, the edge detection classified LMS training filter improves 0.05 (dB) in PSNR over the variance classified LMS training filter. Their difference is very small. But the edge detection classified LS training filter improves 0.5 (dB) in PSNR over the variance classified LS training filter. This verifies that the LMS is not the optimal training filter again, and the LS provides obvious improvement for classified filtering. The edge detection classified LS training filter improves 1.3 (dB) in PSNR over

the single LS training filter, which is a significant improvement.

Although the LS algorithm is more time-consuming than the LMS in training stage [12], this does not increase computation complexity for LS inverse halftoning. Because when we produce inverse halftoned images, we only need to know the filter coefficients that have been trained. The time that is required for training filters is not related to the time for inverse halftoning.

In fact, the PSNR value of an image only shows the overall performance. But the reconstructed gray-level image, as compared to the original gray-level image, has larger errors in regions of the edge. If we compute difference of the original gray-level image and the inverse halftoned images, we can observe errors of each pixel. We find that for the low-frequency signal, the edge detection classified filter is better than variance classified filter because the edge detection classified filter can successfully classify the low-frequency blocks. Besides, edge detection classified filter also performs better for high-frequency blocks. For edge detection classified LS filter, we find that the filter coefficients are larger at the edge. In addition, if pixel differences between two sides of the edge are larger, the filter coefficients are also larger.

TABLE I

PSNRs OF THE INVERSE HALFTONED IMAGES FOR DIFFERENT FILTERS

PSNR (dB)	LMS	LS	Variance-LMS filter	Variance-LS filter	Edge detect-LMS filter	Edge detect-LS filter
airplane	30.46	30.69	31.76	31.93	31.71	32.03
barbara	24.45	24.82	24.87	24.93	24.68	24.99
boat	29.02	29.40	30.13	30.22	29.99	30.70
bridge	25.44	25.90	26.20	26.20	26.13	26.76
elaine	30.90	30.49	31.22	31.40	31.76	31.88
goldhill	29.56	29.66	30.05	30.11	30.51	30.82
kiel	23.57	24.13	24.92	24.91	23.76	24.94
lenna	31.08	31.02	31.88	32.05	32.38	32.58
mandrill	22.99	23.53	24.01	23.96	23.64	24.47
milkdrop	32.96	32.08	33.91	34.27	34.41	35.33
pepper	30.93	30.62	31.57	31.81	31.93	32.62
sailboat	28.51	28.79	29.47	29.57	29.70	29.88
average	28.32	28.43	29.17	29.28	29.22	29.75



Fig. 5. The 12 test images of size 512 × 512.

V. CONCLUSION

In this paper, we propose an inverse halftoning method based on edge detection classification to improve existing variance classified inverse halftoning method. With edge

detection classification, the reconstructed gray-level images have sharper edges and larger PSNRs. We also propose to use the optimal LS algorithm to replace the existing LMS algorithm to train the filter coefficients for inverse halftoning. Effectiveness of our proposed edge detection classification inverse halftoning with LS filters is shown from computer experiments.

Classification could reconstruct the inverse halftoned image closer to the original image. However, improper classification could result in even worse reconstructed output image. From our experiments, we found that with single filter there exists much noise in low-frequency components of output inverse halftoned image, which can be reduced using classified multi-filters. However, in practice the filters that we choose for inverse halftoning are usually in the same smaller set. For increasing the quality of inverse halftoned images, we use much more computations and memory for many filters that few halftone image blocks to be reconstructed will use. Reducing filter number is a problem that will be necessary to be solved.

ACKNOWLEDGMENT

This work was supported by the National Science Council, R. O. C., under Contract NSC 100-2221-E-033-068-MY3.

REFERENCES

- [1] M. Mese and P. P. Vaidyanathan, "Look-up table (LUT) method for inverse halftoning," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1566-1578, Oct. 2001.
- [2] K. L. Chung and S. T. Wu, "Inverse halftoning algorithm using edge-based lookup table approach," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1583-1589, Oct. 2005.
- [3] Jing-Ming Guo and Jen-Ho Chen, "Inverse halftoning with variance classified filtering," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 1293-1296, April. 2009.
- [4] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd Edition, 2007.
- [5] Je-Ho Lee and J. P. Allebach, "Inkjet printer model-based halftoning," *IEEE Transactions on Image Processing*, vol. 14, pp. 674-689, 2005.
- [6] Mengtsung Tai, Shin Liao, and Rochelle Yi-Hsuan Yang, "Scanning and printing resolution," *Information Sciences and Interaction Sciences (ICIS), 2010 3rd International Conference on*, pp 708-710, 2010.
- [7] R. Ulichney, *Digital Halftoning*, Cambridge, MA. MIT Press, 1987.
- [8] Henry R. Kang, *Digital Color Halftoning*, Peerless Systems Corporation, 1999.
- [9] J. F. Jarvis, C. N. Judice, and W. H. Ninke, "A survey of techniques for the display of continuous-tone pictures on bilevel displays," *Comput. Graph. Image Process.*, vol. 5, pp. 13-40, 1976.
- [10] D. E. Knuth, "Digital halftones by dot diffusion," *ACM Trans. Graph.*, vol. 6, pp. 245-273, Oct. 1987.
- [11] P. C. Chang and C. S. Yu, "Neural net classification and LMS reconstruction to halftone images," in *Proc. SPIE Conf. Visual Communications Image Processing*, vol. 3309, pp. 592-602, Jan. 1998.
- [12] Simon Haykin, *Adaptive Filter Theory*, 4th Edition, 2001.