Inverse Halftoning with Variance Classified Filtering

Jing-Ming Guo, Member, IEEE and Jen-Ho Chen

Department of Electrical Engineering,
National Taiwan University of Science and Technology
Taipei, Taiwan
E-mail: jmguo@seed.net.tw

ABSTRACT

Inverse halftoning is a key technology to yield a continuous tone image from a halftone image. The main application is to make some image enhancement or further compression of halftones more feasible. Many former approaches have been proposed in the literature. Among these, a recent method proposed by Chung and Wu using edge-based lookup table achieves good image quality, where the edge feature is adopted to refine the trained Look-Up Table (LUT). However, it has three deficiencies, including 1) the edge features are limited in some predefined cases, which cannot full represent every potential possibility, 2) the lookup table grows exponentially when extreme grayscales are attempted to be recorded, and 3) the trained lookup table cannot fully include all the cases, which leaves some halftone patterns in practice have no associated output grayscale. Chang et al.'s method employed one trained filter to compensate the halftone patterns that are not recorded in LUT. However, one filter cannot fully characterize the full textures in an image. In this study, the halftone patterns are classified according to its variance and then used to train the corresponding filter sets, which are then employed to provide higher prediction accuracy by inner product with the corresponding halftone patterns. As documented in the experimental results, the proposed inverse halftoning provides excellent performance in image quality and memory consumption than former approaches.

Key words: dot diffusion, error diffusion, direct binary search, ordered dithering, digital halftoning.

1. INTRODUCTION

Digital halftoning [1]-[2] is a technique for converting grayscale images into halftone images. These halftone images resemble the original grayscale images when viewed from a distance due to the low-pass filtering nature of the Human Visual System (HVS). The technique is used widely in computer print-outs, printed books, newspapers and magazines, as they are mostly constrained to the black-and-white format (with and without ink). Another major application of digital halftoning is color quantization with a restricted color palette. Well-known halftoning methods include ordered dithering (OD) [1], dot diffusion (DD) [3]-[4], error diffusion (EDF) [5] and iteration-based halftoning: direct binary search (DBS) [6].

Conversely, Inverse halftoning is a conjugate technique to halftoning, which restores the halftone images to grayscale images. This technique can be used for certain applications such as image enhancement or compression.

Mese-Vaidyanathan's inverse halftoning [7]-[10] is classified as Look-Up Table (LUT) based method, where numerous of images with both original grayscale and halftone versions are involved in LUT construction. Thereby, most halftone patterns map to their associated grayscales. However, the trained lookup table cannot fully include all the halftone patterns, which leaves some halftone patterns in practice have no associated output grayscale. For this, they adopt best linear estimator to predict those nonexistent patterns.

In Chang et al.'s inverse halftoning [11], [12] , the classification procedure for various types of halftone images takes priority over the algorithm of inverse halftoning as they attempt to improve the resulting quality of inverse halftone images. Firstly, the halftone images are transformed into Fourier spectrums in order to characterize the features of different halftone images. However, the complexity is too high. Hence, the halftone method classification is later replaced by cooperation between one-dimensional correlation and three-layer back propagation neural network. For inverse halftoning, a hybrid approach is proposed, where the LUT approach is first applied and then the Minimum Mean Square Error (MMSE) is used to solve the nonexistent pattern problem. However, a specific halftoning approach is simply mapped to a trained filter, which makes it difficult to precisely predict a natural image with diversities of properties simultaneously, such as high or low frequencies regions.

Chung-Wu's inverse halftoning [13] improves Mese-Yaidyanathan's method by adopting edge feature in the lookup table refinement. This approach achieves sharper edge than that of the previous LUT approaches. However, the edge features are limited in some predefined cases, which cannot full represent every potential possibility. Two other deficiencies include: 1) the lookup table grows exponentially when extreme grayscales are attempted to be recorded, and 2) as the problem in Mese-Yaidyanathan's method, the trained LUT cannot fully include all the cases, which leaves some halftone patterns in practice have no associated output grayscale.

In this work, the Least-Mean-Square (LMS) is employed to train filter sets, which are then used to reconstruct the associated grayscale from a halftone pattern. The variance is used to classify halftone patterns into groups, and each group independently trains the corresponding filter. The trained filter sets are then employed to provide better reconstructed quality by inner product with the corresponding halftone patterns in the same group. As documented in the experimental results, the proposed inverse halftoning provides excellent performance in image quality and memory consumption than former approaches.

2. FORMER FILTER-BASED APPROACH

As mentioned above, this study presents a filter-based approach for inverse halftoning. Previously, Chang et al.'s method [12] employs LMS to derive a filter $w_{m,n}$, which is then used to convolute with a halftone image $b_{i,j}$ to produce a corresponding grayscale image $\hat{x}_{i,j}$ as described below.

$$\hat{x}_{i,j} = \sum_{m,n \in R} \sum w_{m,n} b_{i+m,j+n}, \quad (1)$$

The derivation procedure of the $w_{m,n}$ is given as below

$$e_{i,j}^2 = (x_{i,j} - \hat{x}_{i,j})^2, \tag{2}$$

$$\frac{\partial e_{i,j}^2}{\partial w_{m,n}} = -2e_{i,j}b_{i+m,j+n}, \tag{3}$$

$$\begin{cases} if\ w_{m,n} > w_{m,n,opt},\ slope > 0,\ w_{m,n}\ be\ decreased \\ if\ w_{m,n} < w_{m,n,opt},\ slope < 0,\ w_{m,n}\ be\ increased, \end{cases} \tag{4}$$

$$w_{m,n}^{(k+1)} = w_{m,n}^{k} + \mu e_{i,j}b_{i+m,j+n}, \tag{5}$$

where $w_{i,j,opt}$ denotes the optimum LMS coefficient; $x_{i,j}$ denotes the original grayscale image; $e_{i,j}^2$ denotes the MSE between $x_{i,j}$ and $\hat{x}_{i,j}$; μ denotes the adjusting parameter used to control the convergent speed of the LMS optimum procedure, which is set to be 10^{-5} in our following experiments. In general, the more images involved in the training process, the better generality the reconstructed filter can be.

Another issue is that how big the filter should be? For this, four different configurations are tested, e.g., 3x3, 5x5, 7x7, and 9x9. In the experiment, various sizes of filters are trained using the 30 training images above and then applied to their Floyd error-diffused halftone version as indicated by Eq. 1. Figure 1 shows the relationship between average reconstructed PSNR and filter size. Apparently, as the filter size is increased, higher PSNR is available. However, the quality improvement gets much slowly after filter size reaches 7x7. Since larger filter size leads to higher complexity, this study used the filter size of 7x7 for later experiments. Figure 2(a) shows the single filter trained by the 30 training images.

A single filter has its limitation in covering all types of images. Even in one image, different regions may have dramatically different textures. For this, we proposed a strategy using classified trained filter sets which can adapt to different types of halftones as introduced in the Section 3.

3. Proposed Variance Classified Filtering Approach

In Chung-Wu's method [13], 34 edge patterns of size 3x3 are listed for LUT refinement and achieved good image quality. However, the 34 edge patterns are far smaller than the overall possible edge patterns. For this, the pattern variance is employed to classify patterns into groups and then each group produces a corresponding trained filter. The pattern size is fixed at 7x7 in this study. Before we can calculate the variance of a halftone pattern, a rough inverse halftoning, as indicated in Eq. 1, is applied to the thirty training images to obtain the temporary grayscale images using the single trained filter as shown in Fig. 3(a). For now, the thirty halftone images and their temporary associated grayscale images are available, which makes the variance evaluation of the halftone patterns feasible using their corresponding grayscale patterns.

There are two ways for pattern classification according to its variance. The training patterns are obtained by dividing 30 training images into 7x7 overlapped patterns. The first method is to separate patterns into subsets, in which each subset has the same pattern number as others, and the patterns in the same subset have similar variances. The conceptual classification diagram is illustrated in Fig. 2. Two reconstructed filter sets, five and ten, using LMS with this method are shown in Figs. 3(b)-(c), where the j-axis is squeezed for better visual comparison. As we know that different shapes of filter sets can produce totally different inverse halftoning result. If two filters are resembled in their shapes, they cannot precisely characterize the corresponding grayscale of two different halftone patterns. Apparently, with the first classification method, some filters have similar shapes, which is the main deficiency in this method. For this, the Maximum Filters Difference Guidance (MFDG) is proposed to classify the patterns to produce the maximum difference between every two adjacent filters as introduced below:

1. The training patterns are initially classified into three equal-number sets according to their variance as the first method.
2. Three filters are trained with the LMS algorithm using the three sets in the first step, and the absolute coefficients difference of every two adjacent filters are evaluated as below, where the variable T denotes the number of the filters (In the first iteration, T=3).
$$DIF^{(k)} = \sum_{m=1}^{7}\sum_{n=1}^{7}\left|w_{m,n}^{(k)} - w_{m,n}^{(k+1)}\right|, \quad k = 1, 2, ..., T-1 \tag{11}$$
3. One more filter should be inserted into the two adjacent filters with $max(DIF^{(k)})$. Hence, the patterns correspond to the original two filters with $max(DIF^{(k)})$ are split into three sets, and each set is used to train the associated filter.
4. The procedure continues until the number of filters meets the requirement as given in Eq. 12; otherwise, go back to Step 2 and continue. The variable $AVG_{PSNR}^{(m)}$ denotes the mth iteration average PSNR of the reconstructed inverse halftone results using the 30 training images. The threshold is set at 0.005 in this work. The lower value in threshold, the better quality can be obtained. However, according to experimental results, the improvement is negligible when the threshold is lower than this value.
$$AVG_{PSNR}^{(m)} - AVG_{PSNR}^{(m-1)} < Threshold \tag{12}$$

Two examples using MFDG with five and ten reconstructed filter sets are shown in Figs. 3(d)-(e). Apparently, the filter sets reconstructed using MFDG provide the highest discrepancies between filters than that of the previous results as shown in Figs. 3(b)-(c). The corresponding inverse halftoning quality comparisons between the two methods will be organized in Section 4.

4. Experimental Results

In this section, the proposed inverse halftoning is applied for quantitative performance evaluation.

Figure 4 shows Average quality performance **of** variance classified with MFDG. Two types of images are involved for objective performance comparisons; the first set involves 30 training images, and the second set involves 11 test images. As expected, the average reconstructed quality of the training images is higher than that of the test images, since the filters are obtained from the training images.

Figures 5(a)-(e) show some practical reconstructed results which are obtained by the proposed methods and two former approaches, LIH [7] and ELIH [13]. Some observations are organized below: The results involved variance MFDG have clearer edges, such as the edge of the hat, and more uniform reconstructed textures in low frequency regions, such as face, shoulder, and some of the smooth backgrounds. The results obtained by LIH and ELIH have many error-diffused-like artifacts throughout the entire image, especially those areas with low frequencies. The reason that the proposed method can provide better edges and smooth regions can be credited from the shapes of the trained filters. For example, Fig. 3(d) shows the five filters trained by variance MFDG, where the filter closest to the origin associates to the training patterns with lowest variances, and the shape is flattest. The filters to the left are increasing in their height as the associated variances of the

training patterns are increased. A flat filter can eliminate the unwanted noises, e.g., error-diffused-like artifacts, and a high filter can retain the sharpness of the original image. Notably, the two images, as shown in Figs. 5(c)-(d), are implemented following the instructions given in [7] and [13] by ourselves. Since the training images used in this work for constructing the lookup table are different from that used in LIH [7] and ELIH [13], the corresponding PSNR may have somewhat discrepancies between them.

Table I provides memory consumption, complexity, and quality comparisons with various inverse halftoning approaches for Floyd error diffusion. In memory consumption regard, the proposed method demands 10 filters; each filter is of size 7x7, and each coefficient is in floating point. The overall consumption is 1960 bytes, which is significantly lower than the most former approaches. In complexity, the proposed method uses one 7x7 filter to temporarily produce an approximate inverse halftoning result, which is then used for variance and grayscale determination for each pixel. The two parameters, variance and grayscale, along with the corresponding dot type of its halftone image are used to select the corresponding optimized filter for convolution. Thereby, the associated complexity is very low compared to those with transformation or iteration-based former approaches. The PSNRs in the forth column are retrieved from literature. According to the objective data, the proposed method is superior to these methods in this regard as well. Notably, the PSNR of the Lena image addressed in Table 2 is 31.02dB in ELIH, which is different from the PSNR=30.81 dB we implemented in Fig. 15(j). We consider that this is due to the training images used are different between this study and [13].

5. Conclusions

Digital halftoning solves the problem of displaying limited color levels results, such as printer and electronics papers. However, the halftone structure is detrimental for some image processing, such as enhancement, compression and re-sampling. For this, inverse halftoning plays an important role in reconstructing the corresponding grayscale result.

Former related research conducted by Chang et al. [12] adopts simply one trained filter to reconstruct the grayscale image, which cannot fully characterize the diversity of various regions in an image. Another recent research conducted by Chung-Wu [13] uses edge-based lookup table approach having three deficiencies: 1) the edge features are limited in some predefined cases, which cannot full represent every potential possibility, 2) the lookup table grows exponentially when extreme grayscales are attempted to be recorded, and 3) The trained lookup table cannot fully include all the cases, which leaves some halftone patterns in practice have no associated output grayscale. This work presents an inverse halftoning approach with variance classified filtering. The filter set is developed by classifying patterns according to the variance.

As we expected, the experimental results show that the overall performance improvement is available with the proposed variance classified strategy. Although this work seems achieving excellent performance in reconstructed image quality, and memory consumption by comparing to former approaches. Here below provides some possible future research directions: 1) exploring more prominent features which can better characterize the halftones and its associated grayscale patterns; 2) hybrid method by cooperating the proposed method with other approaches, such as ELUT, MAP, and iterative projection, and et al., and 3) looking for more representable training images, as good training images can better cover every texture possibility.

References

[1] R. Ulichney, Digital Halftoning. Cambridge, MA. MIT Press, 1987.

[2] D. L. Lau and G. R. Arce, Modern Digital Halftoning. New York: Marcel Dekker, 2001.

[3] D. E. Knuth, "Digital halftones by dot diffusion," ACM Trans. Graph., vol. 6, no. 4, Oct. 1987.

[4] M. Mese and P. P. Vaidyanathan, "Optimized halftoning using dot diffusion and methods for inverse halftoning," IEEE Trans. on Image Processing, vol. 9, pp. 691-709, Apr. 2000.

[5] R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial gray scale," in proc. SID 75 Digest. Society for information Display, pp. 36-37, 1975.

[6] M. Analoui and J. P. Allebach, "Model based halftoning using direct binary search," in Proc. SPIE, Human Vision, Visual Proc., Digital Display III, (San Jose, CA), vol. 1666, pp. 96-108, Feb. 1992.

[7] M. Mese and P. P. Vaidyanathan, "Look-Up Table (LUT) Method for Inverse Halftoning," IEEE Trans. on Image Processing, vol. 10, no. 10, pp. 1566-1578, Oct. 2001.

[8] M. Mese and P. P. Vaidyanathan, "Template selection for LUT inverse halftoning and application to color halftones," IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 6, pp. 2290-2293, June 2000.

[9] M. Mese and P. P. Vaidyanathan, "Look up table (LUT) inverse halftoning," IEEE International Symposium on Circuits and Systems, vol. 4, pp. 517-520, May 2000.

[10] M. Mese and P. P. Vaidyanathan, "Tree-structured method for LUT inverse halftoning and for image halftoning," IEEE Transactions on Image Processing, vol. 11, no. 6, pp. 644-655, June 2002.

[11] P. C. Chang and C. S. Yu, "Neural net classification and LMS reconstruction to halftone images," Proc. SPIE, vol. 3309, pp. 592-602, 1998.

[12] P. C. Chang, C. S. Yu and T. H Lee, "Hybrid LMS-MMSE inverse halftoning technique," IEEE Trans. on Image Processing, vol. 10, no. 1, pp. 95-103, Jan. 2001.

[13] 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.

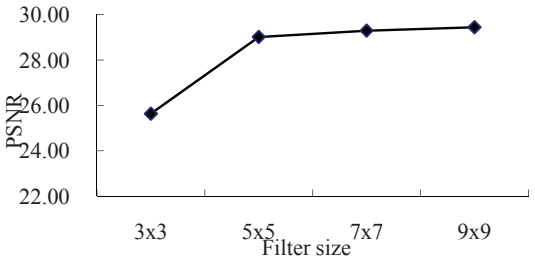

Fig. 1. Relationship between average reconstructed PSNR and filter sizes (using 30 test images)

Fig. 2. Variance classification of patterns with equal number.

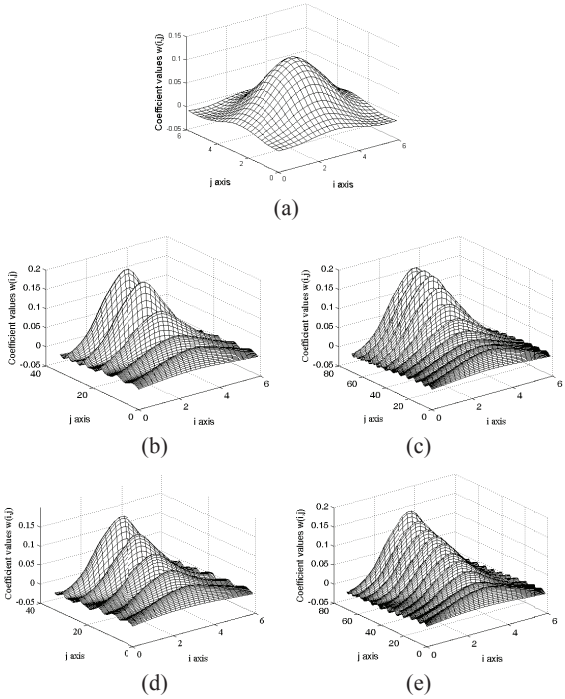

Fig. 3. Pattern variance-classified trained filter sets. (a) Without variance classification. (b) Five pattern variance-classified trained filter sets with equal training patterns in each set. (c) Ten pattern variance-classified trained filter sets with equal training patterns in each set. (d) Five pattern variance-classified trained filter sets using MFDG. (e) Ten pattern variance-classified train filter sets using MFDG.

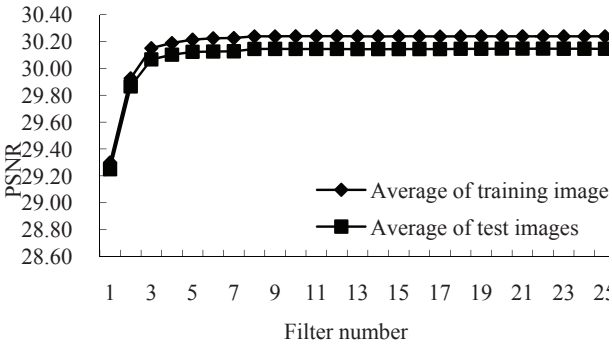

Fig. 4. Average quality performance of variance classified with MFDG.

Fig. 5. Practical inverse halftoning results using Lena image. (a) Original grayscale image. (b) Floyd error-diffused image. (c) Inverse halftoning result of LIH [7]. (d) Inverse halftoning result of ELIH [13]. (e) Inverse halftoning result with the proposed method.

TABLE I PERFORMANCE COMPARISONS AMONG VARIOUS INVERSE HALFTONING APPROACHES FOR FLOYD-STEINBERG ERROR DIFFUSION. (SUPPOSE IMAGE SIZE=NxN)

Algorithm [Reference]	Memory consumption (Bytes)	Complexity	PSNR	
			Lena	Peppers
LMS-MMSE [12]	2^{16}	Very low	30.79	30.67
LIH [7]	2^{16}	Very low	30.36	30.28
ELIH [13]	$39x2^{16}$	Very low	31.02	30.98
*Proposed method	1960	Very low	31.92	31.79

* The memory consumption of the proposed method involves 10 filters and each of size 7x7, which equivalent to 7x7x10x4=1960 bytes.