RESOLVING THE BIASES IN THRESHOLD DECOMPOSITION-BASED MULTITONING

Lai-Yan Wong and *Yuk-Hee Chan*

Centre of Multimedia Signal Processing
Dept. of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong

ABSTRACT

Threshold decomposition is a technique commonly used to produce multitones in multilevel halftoning. It separates an input image nonlinearly into energy planes, halftones them sequentially with a binary halftoning algorithm, and finally combines the binary halftoning results to produce a multitone. As planes are handled sequentially from the brightest layer to the darkest layer under a stacking constraint, there are biases to favor the brighter layers and the darker features of the image. This in turn makes the resultant multitone impossible to report the original image ideally. This paper proposes a solution to eliminate these biases and improve the visual quality of a produced multitone.

Index Terms — Halftoning, multilevel halftoning, multitoning, error diffusion, multiscale error diffusion

1. INTRODUCTION

In the past few decades, the printing technology has been advanced significantly such that printers nowadays can produce outputs of more than two intensity levels. Accordingly, advanced digital halftoning technique is required to convert a gray-scale image into a multilevel output for being printed. The corresponding conversion process is generally referred to as digital multitoning.

Various studies have reported that a straight forward extension of conventional binary halftoning algorithms did not work properly to produce multilevel output of good quality especially when the output levels are limited to a few. In a region over which intensity values gradually change, all intensity levels are quantized to a few output levels in the multitone. When the intensity difference of the few output levels is visually remarkable, the multitone fails to reproduce the gradual variation in gradation and there are banding artifacts.

Threshold decomposition (TD) is a technique now widely used with a binary halftoning algorithm to produce multitones[1-3]. This technique separates an input image nonlinearly into several energy planes, halftones them sequentially with a binary halftoning algorithm, and finally combines the binary halftoning results to produce a multitone.

Different energy planes carry different amount of energy. To eliminate the banding artifacts, the energy planes are halftoned from the one of the highest energy to the one of the lowest energy one by one subject to a stacking constraint [1-3]. In particular, if a location in a plane of higher energy is assigned a black pixel in its binary output, black pixels must also be assigned to the corresponding locations in the binary outputs of the planes of lower energy. This stacking constraint confines the pixel assignment in the binary outputs of the planes of lower energy. The lower the energy of a plane, the more restricted we can assign a white pixel. Since the final multitone is produced by combining the binary halftones of all planes, restricting the positions of white pixels in some binary halftones implies restricting the positions of the bright pixels in the final multitone. In other words, pixel assignments have a bias to favor darker dots. Consequently, we may not be able to put bright dots at the right positions in the multitone to preserve bright features in the original gray level image.

Another observation is that in this conventional framework energy planes are separately processed. When an energy plane is processed, all the planes of lower energy are not taken into account. It is impossible for one to have a complete view on the entire multitoning process when halftoning individual energy planes. From that point of view, the quality of the multitone cannot be globally optimized.

In this paper, we propose a new approach to do multitoning based on the idea of TD. Algorithms developed based on this approach are able to improve the output quality by taking all energy planes into account at any stages of multitoning and eliminating the aforementioned bias. A multitoning algorithm for producing 3-level multitones is presented in this paper as an example to illustrate the idea of the proposed approach.

2. THRESHOLD DECOMPOSITION

This section provides a brief review on the framework of conventional TD-based multitoning algorithms and elaborates the limitations of this conventional framework.

Consider the case that we want to convert a gray–level image $\boldsymbol{X}$ to a 3-level image $\boldsymbol{Y}$. The pixel values of $\boldsymbol{X}$ are bounded in [0,1], where 0 and 1 denote the minimum (black) and the maximum (white) intensity levels respectively, while the pixel values of $\boldsymbol{Y}$ are confined to be 0, 0.5 or 1. Without loss of generality, we assume that the size of images $\boldsymbol{X}$ and $\boldsymbol{Y}$ is $2^k \times 2^k$, where k is a positive integer. For reference, $I(m,n)$ denotes the pixel value of image $\boldsymbol{I}$ at position (m,n).

Figure 1 shows the framework of a conventional TD-based 3-level multitoning algorithm. First, the input image $\boldsymbol{X}$ is decomposed into two energy planes, each of which is denoted as $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$, such that we have

$$X(m,n) = \frac{X_1(m,n) + X_2(m,n)}{2} \quad \text{for } m,n = 0,1,\ldots,2^k\text{-}1 \quad (1)$$

As suggested by Suetake [1], $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ can be determined as

$$X_d(m,n) = X_{d-1}(m,n) - X(m,n)^{d-1}\left(\frac{2!}{(d-1)!(3-d)!}\right)(1 - X(m,n))^{3-d} \quad \text{for } d = 1,2 \quad (2)$$

where $X_0(m,n) = 1$ for all (m,n). The decomposition curves plotted in Figure 2 shows how $X_1(m,n)$ and $X_2(m,n)$ vary with $X(m,n)$. From the curves, one can derive that energy plane $\boldsymbol{X_1}$ always carries more energy than energy plane $\boldsymbol{X_2}$ after decomposition.

After layer decomposition, $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ are halftoned with a binary halftoning algorithm sequentially subject to a stacking constraint. The 3-level multitone is then obtained by averaging $\boldsymbol{Y_1}$ and $\boldsymbol{Y_2}$, the binary halftones of $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$, as

$$Y(m,n) = \frac{Y_1(m,n) + Y_2(m,n)}{2} \quad \text{for } m,n = 0,1,\ldots,2^k\text{-}1 \quad (3)$$

Note that any binary halftoning algorithm can be applied to produce $\boldsymbol{Y_1}$ and $\boldsymbol{Y_2}$, but the selection of the algorithm affects the quality of the final multitoning result.

As mentioned before, as energy planes are sequentially processed from the one of the highest energy to the one of the lowest energy subject to a stacking constraint, the locations of bright pixels in the final multitone are more restricted than those of dark pixels, and it makes the multitone difficult to preserve bright image features. Note that reversing the processing order to process $\boldsymbol{X_2}$ first does not help. There will be another stacking constraint to satisfy and it just sacrifices the dark image features to preserve the bright image features.

Another observation is that not all energy planes are taken into account when processing a particular energy plane. The binary halftoning processes for individual energy planes cannot be jointly optimized simultaneously. As a consequence, the halftones produced for individual energy planes can be at most optimized for their associated planes only. As the multitoning output is produced by combining the halftones, its quality cannot be globally optimized.

In [4], Wong et al. proposed to process the brightest and the darkest energy planes in an interleaving manner to reduce the bias to either white or black dots. However, it still cannot solve the problems from the root. Obviously, in order to solve these problems, energy planes should not be processed one by one sequentially. Instead, they should all be taken into account at the same time when determining the output intensity value of a specific pixel in the final multitone. In the following section, we will show how this can be done with our proposed approach.

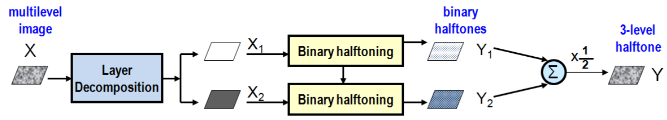

Figure 1. The framework of conventional TD-based 3-level multitoning algorithms

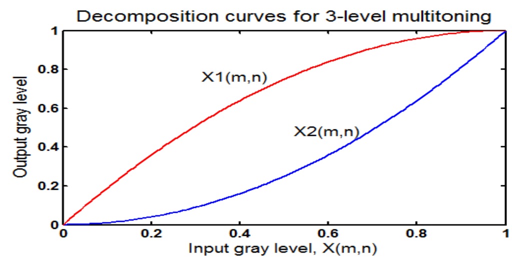

Figure 2. The decompositon curves for 3-level multitoning

3. ALGORITHM

In the proposed framework, the 3-level multitoning process is considered as a process in which one puts white dots and black dots onto a gray substrate. For each pixel in the multitone, its intensity value, say $Y(m,n)$, is determined by evaluating both $X_1(m,n)$ and $X_2(m,n)$ simultaneously at the time the pixel is selected to assign a value. In such an arrangement, there is no bias to favor either white or black dots and hence the aforementioned problems can be resolved automatically.

To start the process, one has to estimate the total budgets of black dots and white dots that should be put on the gray substrate. The budgets can be determined based on the working principle of conventional TD-based multitoning algorithms as follows. According to the rules, the white and black dot budgets, which are defined to be the expected total numbers of white and black pixels, for the binary halftones $\boldsymbol{Y_1}$ and $\boldsymbol{Y_2}$ should be given by

$$\textit{white dot budget for } Y_d = \sum_{(m,n)} X_d(m,n)$$

$$\textit{black dot budget for } Y_d = \sum_{(m,n)} \{1 - X_d(m,n)\} \quad \text{for } d = 1,2 \quad (4)$$

The stacking constraint requests that $Y_2(m,n)=0$ whenever $Y_1(m,n)=0$. Under the stacking constraint and the connection given by eqn. (3), $Y_1(m,n)=0$ is the necessary and sufficient condition for $Y(m,n)=0$. Hence, the black dot budget for the 3-level multitone $\boldsymbol{Y}$ is the same as the black dot budget for $\boldsymbol{Y_1}$ and is given by

$$\textit{black dot budget for } Y,\ B_b = \sum_{(m,n)} \{1 - X_1(m,n)\} \quad (5)$$

Similarly, the connection given by eqn. (3) and the constraint that $Y_1(m,n)=1$ if $Y_2(m,n)=1$ make $Y_2(m,n)=1$ be the necessary and sufficient condition for $Y(m,n)$=1. Accordingly, the white dot budget for the 3-level multitone $\boldsymbol{Y}$ is the same as the white dot budget for $\boldsymbol{Y_2}$ and is given by

$$\text{white dot budget for } Y,\ B_w = \sum_{(m,n)} X_2(m,n) \qquad (6)$$

As a gray substrate, $Y(m,n)$ are initialized to be 0.5 for all (m,n). With the dot budgets on hand, pixels in $\boldsymbol{Y}$ are selected and assigned white or black dots by changing values of $Y(m,n)$ to 1 or 0 one by one until all white and black dot budgets are used up. The implementation is basically a 2-step iterative process adapted from the one used in feature-preserving multiscale error diffusion (FMED) [5].

The first step of the iterative process is to select a pixel to put a dot. The selection is based on a complex energy plane $\boldsymbol{E}$ which is initialized by combining energy planes $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ as follows.

$$E(m,n) = X_2(m,n) + j(1 - X_1(m,n)) \quad \text{for all } (m,n) \qquad (7)$$

where $j = \sqrt{-1}$. It is updated at every iteration to guide us to select the most currently critical pixel in $\boldsymbol{Y}$ to put a dot.

Starting with the energy plane $\boldsymbol{E}$ as the region of interest, we repeatedly divide the region of interest into nine overlapped sub-regions of equal size and select the sub-region whose cost is maximum to be the new region of interest. The cost of a region is defined as

$$J_R = \left\| \max(\mathrm{Re}(C_R),0) + j\max(\mathrm{Im}(C_R),0) \right\|^2 \qquad (8)$$

where $\mathrm{Re}(C_R)$ and $\mathrm{Im}(C_R)$ are, respectively, the real and the imaginary parts of the complex value defined as $C_R = \sum_{(u,v)\in R} E(u,v)M(u,v)$, R denotes the set of pixels in the region and $M(u,v)$ is a mask defined as

$$M(u,v) = \begin{cases} 1 & \text{if } Y(u,v) = 0.5 \\ 0 & \text{otherwise} \end{cases} \qquad (9)$$

Note that the physical meaning of $Y(u,v)$=0.5 is that pixel (u,v) in $\boldsymbol{Y}$ has not yet been selected to put either a white or black dot. We repeat the above procedures to update the region of interest until a pixel location is reached.

In the conventional TD framework, $X_2(m,n)$ and $(1 - X_1(m,n))$, respectively, reflects the appropriateness of assigning 1 (a white dot) to $Y_2(m,n)$ and 0 (a black dot) to $Y_1(m,n)$. The closer to 0 their values, the less appropriate the assignments are. In other words, the larger the value of $(1 - X_1(m,n))^2 + X_2(m,n)^2$, the more appropriate to make either $Y_2(m,n)$=1 or $Y_1(m,n)$=0. As $Y_2(m,n)$=1 and $Y_1(m,n)$=0 lead to $Y(m,n)$=1 and $Y(m,n)$=0 respectively, it implies that $Y(m,n)$ should no longer be 0.5. Adjusting the initial value of $Y(m,n)$ is equivalent to putting a dot (either black or white) onto the gray substrate at location (m,n). The searching procedures in this step guide us to search for the most appropriate location to put a dot.

The second step is to assign an appropriate value to the pixel selected in the first step and update energy plane $\boldsymbol{E}$ accordingly. Let the location of the selected pixel be (p,q). The intensity value of the dot assigned to $Y(p,q)$ is determined as

$$Y(p,q) = \begin{cases} 1 & \text{if } X_2(p,q) > 1 - X_1(p,q) \text{ and } B_w > 0 \\ 0 & \text{otherwise} \end{cases} \qquad (10)$$

After the dot assignment, the corresponding dot budget for $\boldsymbol{Y}$ is reduced by 1.

Based on eqn. (3), we have $Y_1(p,q) = Y_2(p,q) = Y(p,q)$ if $Y(p,q)$=0 or 1. Energy planes $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ should then be updated accordingly to update the complex energy plane $\boldsymbol{E}$ for selecting another pixel in $\boldsymbol{Y}$ to put a dot in next iteration. In particular, for each energy plane $\boldsymbol{X_d}$, where $d\in\{1,2\}$, the difference between $Y_d(p,q)$ and $X_d(p,q)$ is diffused to $X_d(p,q)$'s neighbors to update energy plane $\boldsymbol{X_d}$ as follows.

$$X'_d(m,n) = \begin{cases} 0 & \text{if } (m,n) = (p,q) \\ X_d(m,n) - w_{m-p,n-q}\cdot\left(Y_d(p,q) - X_d(m,n)\right)\cdot\dfrac{M(m,n)}{S} & \text{otherwise} \end{cases} \qquad (11)$$

where $X'_d(m,n)$ and $X_d(m,n)$ are, respectively, the values of pixel (m,n) in energy plane $\boldsymbol{X_d}$ after and before the error diffusion process, $Y_d(p,q)$ is the value assigned to pixel (p,q), $w_{s,t}$ for $(s,t)\in\Omega$ is a filter weight of a non-causal diffusion filter with support Ω, and

$$S = \sum_{(m-p,n-q)\in\Omega} \left(w_{m-p,n-q}\cdot M(m,n)\right) \qquad (12)$$

The two steps are repeated until all black and white dot budgets for $\boldsymbol{Y}$ are exhausted. Since both steps are carried out based on the complex energy plane $\boldsymbol{E}$ and the whole iterative process complies with the framework of FMED, the process is referred to as complex plane MED in this paper. Figure 3 shows the flow of the proposed multitoning method.

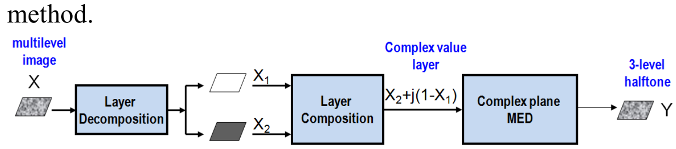

Figure 3. The flow of the proposed method

4. SIMULATION RESULT

Simulations were carried out to evaluate the performance of the proposed method with a set of nine 256-level testing images of size 512×512 each. In its realization, a 5×5 noncausal diffusion filter with filter coefficients

$$w_{s,t} = \begin{cases} 1/\sqrt{s^2+t^2} & \text{if } |s|,|t| \le d = 2 \text{ and } |s|+|t| \ne 0 \\ 0 & \text{otherwise} \end{cases} \qquad (13)$$

was exploited as the default diffusion filter in the simulation. When S=0, the value of d is increased by 1 gradually to increase the filter support until $S \neq 0$. For comparison, the performance of four TD-based multitoning algorithms ([1-4]) was also evaluated in the simulations.

Table 1 shows the performance of various algorithms in terms of Mean Structural Similarity Index (MSSIM) [6]. MSSIM is an improved version of Universal Objective Image Quality Index (UQI) [7] which can be used to measure the information loss after multitoning. The MSSIM measurement is sensitive to structural information degradation which reflects the difference between the original image and its multitoning results. The higher the value of MSSIM, the closer to the original a multitoning output is, which implies a better quality of the output.

Figure 4 shows some 3-level halftoning results obtained with different evaluated algorithms for visual comparison. One can see that the proposed algorithm can render the

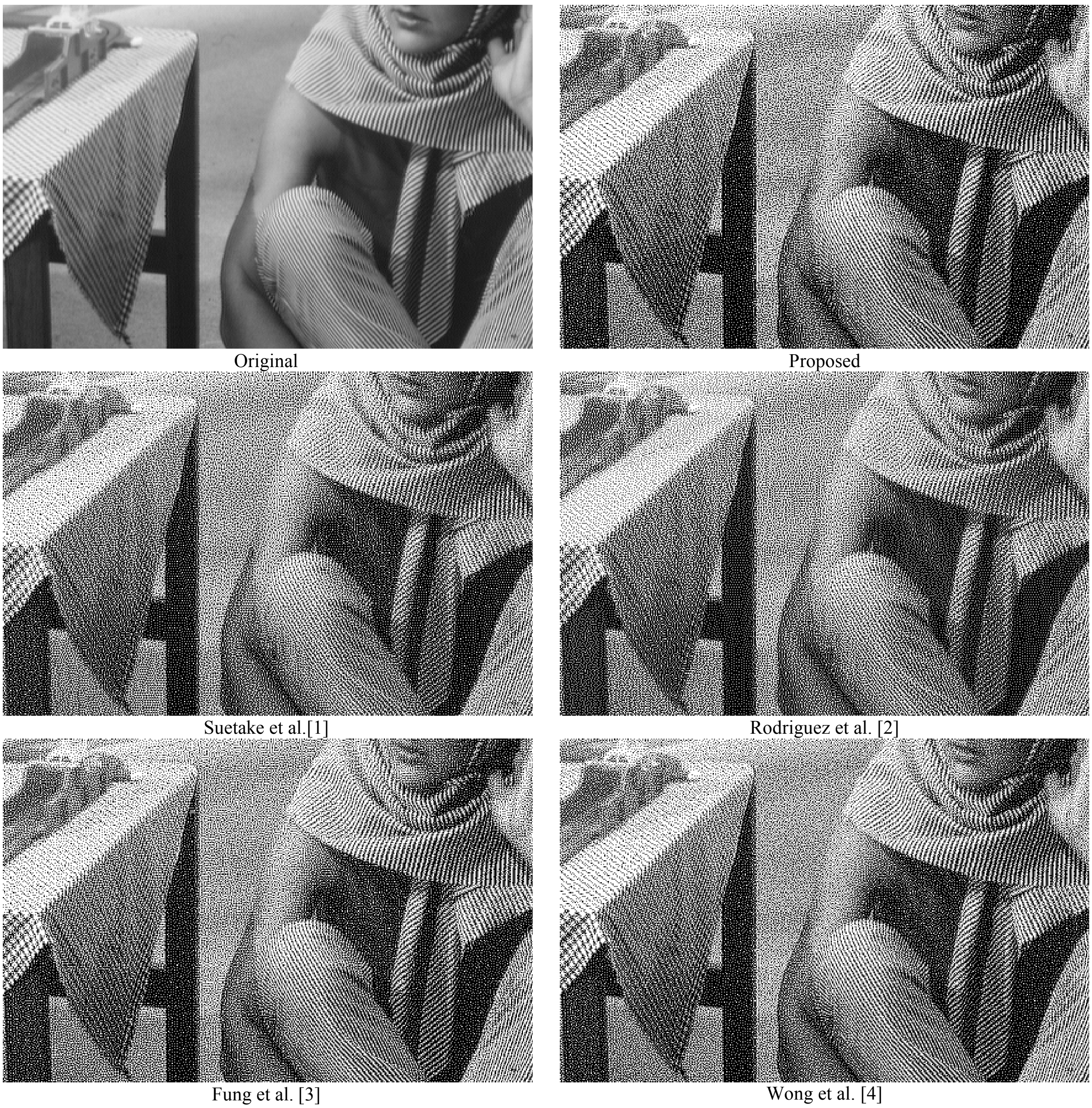

Figure 4. Simulation results of various multitoning algorithms

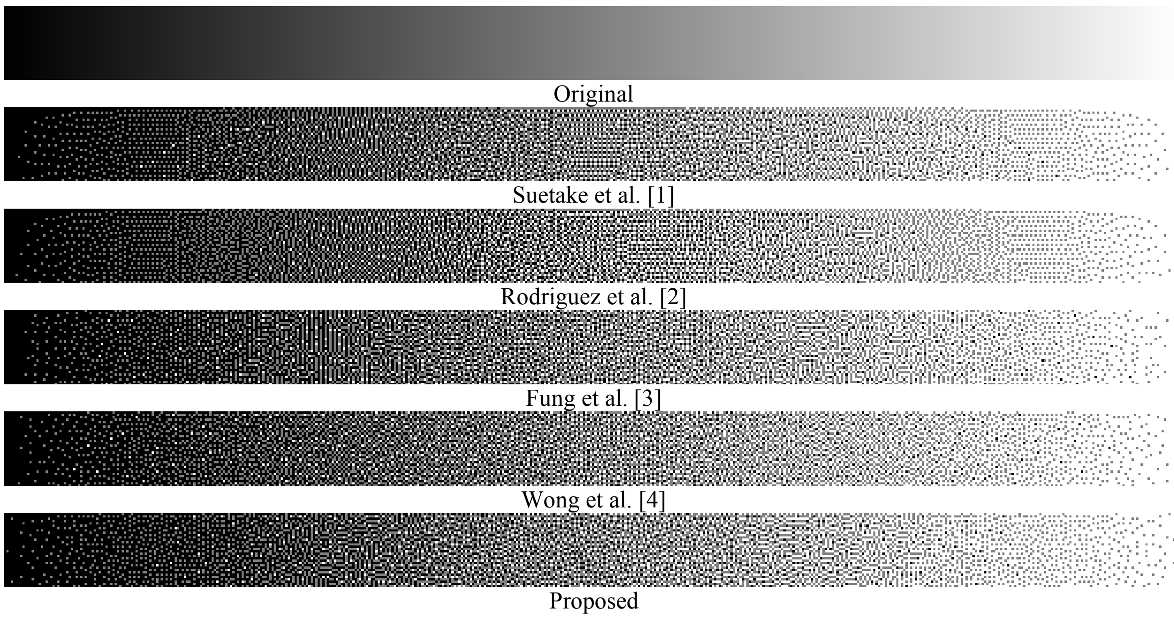

Figure 1. Simulation results of a ramp image

texture (e.g. the patterns on the tablecloth, the scarf and the trousers etc.) accurately. Both bright and dark spatial features are reported faithfully in the output of the proposed algorithm. Figure 5 shows the multitones obtained with various algorithms for a ramp input.

TABLE I. MSSIM PERFORMANCE OF VARIOUS ALGORITHMS

Image	Ours	[1]	[2]	[3]	[4]
airplane	0.1045	0.0747	0.0732	0.1023	0.1016
barbara	0.1866	0.1327	0.1229	0.1801	0.1834
boat	0.1219	0.0829	0.0770	0.1167	0.1174
girl	0.0591	0.0367	0.0348	0.0576	0.0585
goldhill	0.1104	0.0663	0.0612	0.1059	0.1080
lena	0.0916	0.0597	0.0561	0.0874	0.0886
man	0.1161	0.0777	0.0717	0.1111	0.1133
mandrill	0.2736	0.1875	0.1701	0.2686	0.2719
peppers	0.0969	0.0590	0.0560	0.0931	0.0972
Average	**0.1290**	**0.0864**	**0.0803**	**0.1248**	**0.1267**

5. CONCLUSIONS

Threshold decomposition is a technique widely used with a binary halftoning algorithm to produce multitones. It separates an input image nonlinearly into several energy planes and processes the energy planes sequentially to produce a multitone. This unavoidably introduces a bias to favor brighter energy planes and a bias to favor darker features of the input image during multitoning. In this paper, we proposed a solution which allows us to handle all energy planes simultaneously to eliminate the biases. Both dark and bright spatial features of the input images can then be faithfully preserved in their multitoning outputs. Simulation results showed that the proposed method can give a better result than conventional TD-based multilevel halftoning algorithms[1-3] in both subjective and objective measures.

ACKNOWLEDGMENTS

This work was supported by Center for Multimedia Signal Processing and a grant from The Hong Kong Polytechnic University (PolyU Grant G-YL53).

REFERENCES

[1] N. Suetake, M. Sakano, E. Nakashima and E. Uchino, “Simple multilevel halftoning excelled in gradation reproducibility,” *Optics Letters*, Vol. 33, No. 4, pp.339-341, 2008.

[2] J. B. Rodriguez, G. R. Arce and D. L. Lau, “Blue-Noise Multitone Dithering,” *IEEE Trans. Image Process.*, Vol. 17, No.8, pp. 1368–1382, Aug. 2008.

[3] Y. H. Fung and Y. H. Chan, “Multilevel Halftoning using Multiscale Error Diffusion,” *Journal of Electronic Imaging*, Vol.19, Issue 3, 2010.

[4] L.Y. Wong and Y.H. Chan, “A feature preserving multilevel halftoning algorithm,” *Journal of Electronic Imaging* 21(4), 043016 (Nov 27, 2012), doi:10.1117/1.JEI.21.4.043016.

[5] Y. H. Chan and S. M. Cheung, “Feature-preserving multiscale error diffusion for digital halftoning,” *Journal of Electronic Imaging*, Vol.13, Issue 3, 2004.

[6] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” Image Processing, *IEEE Trans. Image Process.* Vol.13, No.4, pp.600-612, 2004.

[7] Z. Wang and A. C. Bovik, “A universal image quality index,” *IEEE Signal Process. Lett.*, vol.9, no. 3,pp.81-84, Mar 2002