Fast ad-hoc Inverse Halftoning using Adaptive Filtering

Oscar C. Au
Department of Electrical and Electronic Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong.
Email: eeau@ee.ust.hk
Tel.: +852 2358-7053**

ABSTRACT

We propose a novel fast inverse halftoning technique using an adaptive spatial varying filtering. The proposed algorithm is significantly simpler than most existing algorithms while achieving a PSNR close to that of the set theoretic POCS.

1. INTRODUCTION

Image halftoning is a process to convert a continue tone image into a binary or halftone image with only black and white dots, which resembles the original image when viewed from a distance. Inverse halftoning is the process to estimate the original image from the halftone image.

Existing methods for inverse halftoning include simple lowpass filtering, set-theoretic projection-on-convex-set (POCS)[1], wavelet-based method using edge information in highpass wavelet image[2], adaptive inverse halftoning using least mean square sliding window filter and wiener filter postprocessing[3], MMSE and MAP projection-based method[4], and three-level cascade algorithm[5], etc. These methods can usually give good visual quality of estimated images, but they are usually computationally expensive. In this paper, we will restrict ourselves to inverse halftoning using linear filter.

It is well known that simple lowpass filtering is a poor way to do inverse halftoning as shown in Figure 1. Using the simple lowpass filter F1=[0 1 0;1 1 1;0 1 0]/5 (in Matlab notation), a peak-signal-to-noise ratio (PSNR) of 16.32dB is obtained for the 256x256 Lena image, where PSNR is defined as $10\log_{10}(255^2/MSE)$ and *MSE* is the mean square error. The filter F2=[1 1 1;1 1 1;1 1 1]/9 gives a PSNR of 22.24dB and F3=[1 4 1;4 7 4;1 4 1]/27 gives 22.48dB. The image obtained from F3 is shown in Figure 1.

Projection-onto-convex-set (POCS) [1] is one of the ways to remedy the problem of spatial invariant filters. After initial lowpass filtering, POCS would iteratively apply "projection" followed by filtering, followed by projection and filtering, and so on. In general, if image halftoning is applied to the filtered image, it would not generate the original halftone image. Therefore, in the "projection" step, the filtered image is altered or projected in such a way that the projected image would generate the original halftone image. POCS can yield significantly better image than simple spatial invariant filter. An example is shown in Figure 2. POCS with F1 gives a PSNR of 27.83dB after 6 iterations. POCS with F2 can give a PSNR of 27.40dB after 2 iterations. POCS with F3 can give a PSNR of 28.05dB after 3 iterations. The estimated image using POCS with F3 is shown in Figure 2. While POCS can give significantly higher PSNR than simple lowpass filtering, its complexity is much higher which is undesirable.

In this paper, we propose a simple and ad-hoc class of inverse halftoning algorithm using spatially varying linear filters. Compared with POCS, the proposed method does not require the projection step which makes it significantly faster in spite of the minor added complex for the spatial varying filter. The PSNR achieved by many algorithms in the proposed class are above 27dB. The highest PSNR observed is 27.54dB, which is within 0.5dB from that achieved by POCS.

2. MOTIVATION

Consider a neighborhood around a pixel located at (i,j). The linear estimate $y(i,j)$ of the pixel at (i,j) should be in the form of

$$y(i,j) = \sum_{(k,l)\in N} a(i+k, j+l)x(i+k, j+l)$$

where $x(i,j)$ is the ij^{th} pixel of the halftone image. Some possible neighborhood, the cross, 3x3 and 5x5, are shown in Figure 3. For spatial invariant filters, the coefficients do not change with i and j, i.e. $a(i+k, j+l) = a(k,l)$.

Such spatial invariant filters tend to be effective in suppressing the artificial high frequency signal components in the halftone images. However, spatial invariant filters do not work well at object boundaries as pixel values from different objects are allowed to influence the same pixel without any discrimination. If discrimination is imposed such that the influence from pixels of the different object is minimized, and that from the same objects is maximized, good filtering results should be possible.

3. THE PROPOSED AD-HOC ADAPTIVE FILTERING

Here we propose to achieve inverse halftoning using a class of spatially varying filter with coefficients being a function of the absolute pixel difference

$$a(i+k, j+l) = f(|x(i,j) - x(i+k, j+l)|)$$

Obviously, the pixel difference is zero for $k=l=0$. And the pixel $x(i,j)$ should be more reliable than the pixels in the neighborhood. When the absolute pixel difference between $x(i+k,j+l)$ and $x(i,j)$ is large, it is not very likely that the two pixels come from the same object. In this case, the weight for $x(i+k,j+l)$ should be small. Similar, when $x(i+k,j+l)$ and $x(i,j)$ are similar, it should be quite likely that they come from the same object. The weight should then be large for $x(i+k,j+l)$. With such observations, the function f should in general be a monotonic non-increasing function.

While there are numerous possible definitions for f, we study several classes:

3.1 Polynomial map

$$f_{lin,k}(i) = (1 - i/255)^k \qquad \text{for } i=0,..., 255$$

with k ranging from 1 to 10. With k=1, this is the linear map.

3.2 Exponential map

$$f_{\exp,k}(i) = e^{-a_k i} \qquad \text{for } i=0,..., 255$$

where the set of a_k are preset constants.

3.3 Exponential map with shift

$$f_{\exp,k,j}(i) = (0.2j - 0.1) + e^{-a_k i} \text{ for } i=0,...,255$$

where j ranges from 1 to 5.

3.4 Piecewise linear map

$$f_{pl,i_1,i_2}(i) = \begin{cases} 1 & for\ i \le i_1 \\ 1-(x-i_1)/(i_2-i_1) & for\ i_1 \le i \le i_2 \\ 0 & for\ i_2 \le i \end{cases}$$

where i_1 and i_2 are parameters. In this paper, we test many pairs of i_1 and i_2. The four classes of maps are shown in Fig. 4.

In general, we can divide the pixels around (i,j) into M different neighborhood. The pixels within a neighborhood are roughly equi-distant from (i,j). We can define a mapping function f_k for each neighborhood to reflect different likelihood patterns for pixels at various distance from (i,j).

4. RESULTS AND DISCUSSIONS

In this paper, we studied three type of neighborhood: the cross, 3x3 and 5x5 shown in Figure 3. The 5x5 actually can have 2 maps with one for the inner 3x3 and the other for the rest of the pixels. The polynomial maps, the exponential map, exponential map with shift, and the piecewise linear map were simulated.

The 256x256 Lena image is used a the test image, which is shown in Figure 5. The halftone image obtained by error diffusion using the Floyd-Steinberg kernal is shown in Figure 6. The PSNR of the proposed algorithm using various mapping and neighborhood are shown in Table 1, 2 and 3. It can be observed that the piecewise linear $f_{pl,k,70}$ coupled with the 3x3 neighborhood gives a PSNR of 27.54dB, which is only 0.5dB lower than the computationally more expensive POCS with F_3. The estimated image is shown in Figure 7 and 8 with 2 iterations and 4 iterations respectively. It can be observed that the visual quality of $f_{pl,k,70}$ coupled with the 3x3 is very similar to that of POCS with F_3.

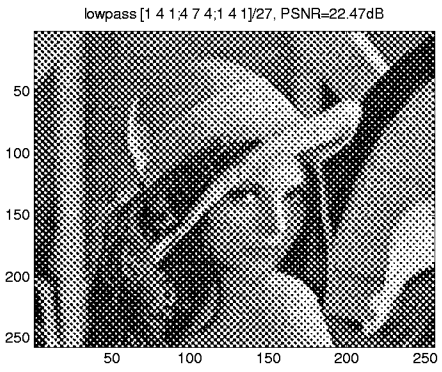


Figure 1: Inverse Halftone using lowpass filter F_3

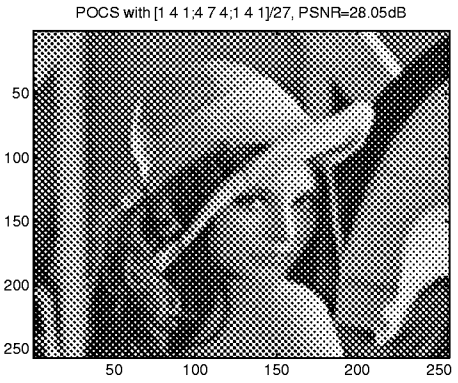
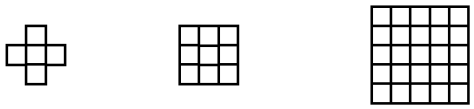


Figure 2: Inverse Halftoning with POCS and F_3



(a) (b) (c)

Figure 3: Possible neighborhood for each pixel

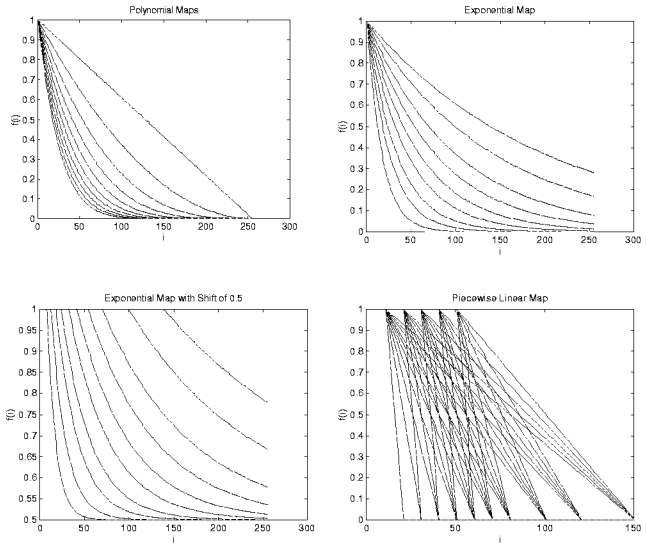


Figure 4: The Polynomial, Exponential, Exponential with shift and Piecewise Linear Mappings

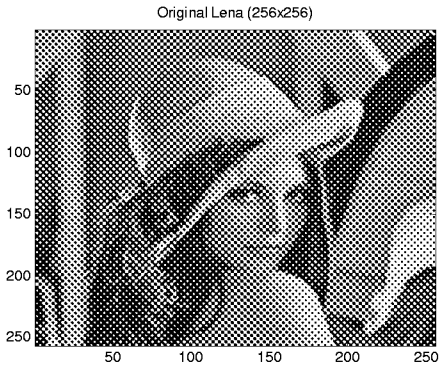


Figure 5: Original 256x256 Lena

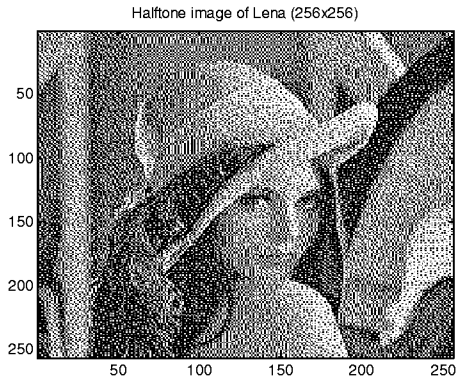


Figure 6: Halftone image of Lena by Error Diffusion

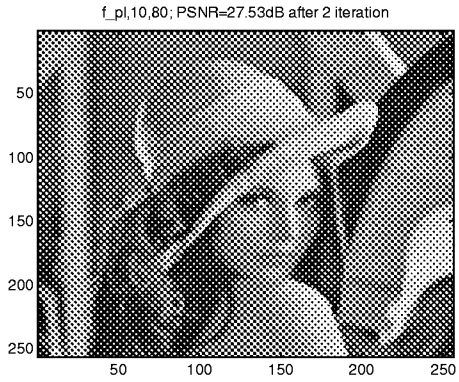


Figure 7: Estimated Image using $f_{pl,k,70}$ coupled with the 3x3 neighborhood after 2 iterations

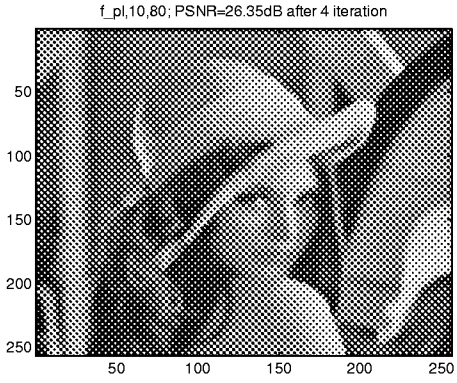


Figure 8: Estimated Image using $f_{pl,k,70}$ coupled with the 3x3 neighborhood after 4 iterations

5x5	k=1	k=2	k=3	k=4	k=5
$f_{lin,k}$	24.27	24.42	24.76	24.95	25.06
$f_{exp,k}$	24.46	24.92	25.23	25.09	25.04
$f_{exp,k,0.1}$	24.46	24.88	25.29	25.45	25.42
$f_{exp,k,0.3}$	24.40	24.73	25.10	25.36	25.56
$f_{exp,k,0.5}$	23.23	23.34	23.49	23.59	23.79
$f_{exp,k,0.7}$	23.45	23.48	23.64	23.80	23.96
$f_{exp,k,0.9}$	24.00	24.00	24.00	24.02	24.07
	k=0	k=10	k=20	k=30	k=40
$f_{pl,k,150}$		24.41	24.32	24.24	24.17
$f_{pl,k,120}$		24.37	24.27	24.18	24.11
$f_{pl,k,100}$	24.01	23.81	23.75	23.69	23.66
$f_{pl,k,80}$	23.47	23.63	23.57	23.52	23.50
$f_{pl,k,70}$	23.05	23.27	23.24	23.22	23.23
$f_{pl,k,60}$	22.44	23.04	23.04	23.05	23.07
$f_{pl,k,50}$	22.04	22.88	22.91	22.94	23.02
$f_{pl,k,40}$	21.44	22.48	22.49	22.55	
$f_{pl,k,30}$	20.58	22.36	22.34		
$f_{pl,k,20}$	19.56	22.26			

Table 1: PSNR of the proposed algorithm using various mapping for a 5x5 neighborhood

3x3	k=1	k=2	k=3	k=4	k=5
$f_{lin,k}$	25.61	25.20	24.58	23.72	22.93
$f_{exp,k}$	25.91	25.75	25.42	25.28	24.96
$f_{exp,k,0.1}$	25.96	25.87	25.70	25.53	25.34
$f_{exp,k,0.3}$	25.98	25.97	25.92	25.88	25.85
$f_{exp,k,0.5}$	25.99	25.96	25.98	25.99	25.98
$f_{exp,k,0.7}$	26.00	26.00	25.97	25.98	26.01
$f_{exp,k,0.9}$	26.00	26.00	26.00	26.00	26.00
	k=0	k=10	k=20	k=30	k=40
$f_{pl,k,150}$	26.97	26.90	26.77	26.60	26.45
$f_{pl,k,120}$	27.20	27.14	27.00	26.80	26.61
$f_{pl,k,100}$	27.39	27.34	27.21	26.99	26.78
$f_{pl,k,80}$	27.52	27.53	27.45	27.23	27.02
$f_{pl,k,70}$	27.45	27.54	27.52	27.33	27.14
$f_{pl,k,60}$	27.29	27.38	27.48	27.37	27.26
$f_{pl,k,50}$	26.98	27.16	27.27	27.23	27.32
$f_{pl,k,40}$	25.73	25.90	26.08	26.09	
$f_{pl,k,30}$	24.58	24.94	25.14		
$f_{pl,k,20}$	20.61	21.26			

Table 2: PSNR of the proposed algorithm using various mapping for a 3x3 neighborhood

cross	k=1	k=2	k=3	k=4	k=5
$f_{lin,k}$	25.83	25.69	25.44	25.04	24.51
$f_{exp,k}$	25.80	25.84	25.80	25.68	25.43
$f_{exp,k,0.1}$	25.78	25.82	25.82	25.77	25.65
$f_{exp,k,0.3}$	25.70	25.77	25.80	25.79	25.75
$f_{exp,k,0.5}$	25.55	25.63	25.73	25.76	25.73
$f_{exp,k,0.7}$	25.53	25.53	25.58	25.64	25.69
$f_{exp,k,0.9}$	25.53	25.53	25.53	25.53	25.54
	k=0	k=10	k=20	k=30	k=40
$f_{pl,k,150}$	26.43	26.25	26.07	25.91	25.78
$f_{pl,k,120}$	26.69	26.47	26.25	26.04	25.88
$f_{pl,k,100}$	26.95	26.70	26.43	26.19	26.00
$f_{pl,k,80}$	27.28	27.01	26.71	26.43	26.20
$f_{pl,k,70}$	27.40	27.18	26.89	26.55	26.32
$f_{pl,k,60}$	27.46	27.34	27.08	26.72	26.41
$f_{pl,k,50}$	27.16	27.21	27.13	26.91	26.73
$f_{pl,k,40}$	25.86	26.01	25.96	25.69	
$f_{pl,k,30}$	23.79	24.22	24.42		
$f_{pl,k,20}$	19.42	20.05			

Table 3: PSNR of the proposed algorithm using various mapping for a "cross" neighborhood

REFERENCE

[1] Thao N.T., "Set Theoretic Inverse Halftoning", *Proc. of IEEE Int. Conf. on Image Processing*, Vo. 1, pp. 783-6, Oct. 1997.

[2] Z. Xiong, M.T. Orchard, K. Ramchandran, "Inverse Halftoning using Wavelets", *Proc. of IEEE Int. Conf. on Image Processing*, Vol. 1, pp. 569-72, Sept. 1996..

[3] L.M. Chen, H.M. Hang, "An Adaptive Inverse Halftoning Algorithm", *IEEE Trans. of Image Processing*, Vol. 6, No. 8, pp. 1202-9, Aug. 1997.

[4] C.M. Miceli, K.J. Parker, "Inverse Halftoning", *J. of Electronic Imaging*, Vol. 1, No. 2, pp. 143-51, Apr. 1992.

[5] P.W. Wong, "Inverse Halftoning and Kernel Estimation for Error Diffusion", *IEEE Trans. on Image Processing*, Vol. 4, No. 4, pp. 486-98, Apr. 1995.