Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 34 (2008) 2491–2501
Expert Systems with Applications
www.elsevier.com/locate/eswa

Neural network based method for image halftoning and inverse halftoning

Win-Bin Huang *, Alvin W.Y. Su, Yau-Hwang Kuo

Department of Computer Science and Information Engineering, National Cheng Kung University, No. 1, Ta-Hsueh Road, Tainan, Taiwan, ROC

Abstract

A hybrid neural network based method for halftoning and inverse halftoning of digital images is presented. The halftone image is performed by single-layer perceptron neural network (SLPNN), and its corresponding continuous-tone image is reconstructed by radial-basis function neural network (RBFNN). The combined training procedure produces halftone images and the corresponding continuous tone images at the same time. The PSNR performance and visual image quality of these contone images achieved is comparable to the well-known inverse halftoning methods. The resultant halftone images compared with the error diffusion halftone are visually good, too. Furthermore, we apply different kinds of halftone images to a bi-level image compression method, called Block Arithmetic Coding for Image Compression (BACIC), which is better than the current facsimile methods.

Keywords: Halftone; Inverse halftoning; RBF neural network; SLP neural network

1. Introduction

Halftoning technology is mainly applied in the fields of printing devices and image displays. It is a process of transforming continuous tone (also called contone) images into halftone images for the devices that can only process bi-level images. The challenge of halftoning still lies in how to produce a high quality image that is visually close to its original through bi-level devices. According to computational style, the halftoning algorithms can be classified into three categories (Kim & Allebach, 2002): neighborhood based methods, point based methods, and iterative optimization methods. Neighborhood based methods, such as error diffusion (Floyd & Steinberg, 1976) and tone-dependent error diffusion (Li & Allebach, 2004), determines the halftone value by comparing the sum of the current pixel and weighted neighborhood errors with a threshold. Point based method, such as screening (Bayer, 1973) and Look-Up-Table (Sullivan, Ray, & Miller, 1991; Li & Allebach, 2000; Mese & Vaidyanathan, 2002), produces the halftone images according to a screen defined by a matrix of threshold values, which is periodically tiled over the image. In general, the halftone images generated by the screening method are visually worse than those generated by the error diffusion method. Iterative optimization methods (Analoui & Allebach, 1992; Kim & Allebach, 2002; Mulligan & Ahumada, 1992; Pappas & Neuhoff, 1992) use a human visual system model to minimize the perceived error between contone images and halftone images. This kind of methods can produce high quality halftone images, but a great deal of computation is required. On the other hand, error diffusion method has good visual quality with low computational complexity, but it may generate worms and other undesirable artifacts. In this paper, we try to develop a neural network model, which can automatically adjust the weights and quantization threshold, to improve the efficiency of image halftoning by error diffusion method.

Besides image halftoning, inverse halftoning that converts halftone images back to the corresponding contone images also has a wide range of applications such as compression, printed image processing, scaling, and

* Corresponding author.
E-mail address: huangwb@cad.csie.ncku.edu.tw (W.-B. Huang).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.04.013

enhancement. In these applications, operations cannot be done on the halftone image directly so that inverse halftoning is mandatory. Many good studies in inverse halftoning have been available in the literature, such as kernel estimation (Wong, 1995), wavelet (Xiong, Orchard, & Ramchandran, 1996), POCS (Bozkurt & Cetin, 1999; Hein & Zakhor, 1995), MAP estimation (Stevenson, 1997), filtering (Fan, 1992; Kite, Venkata, Evans, & Bovik, 2000; Thao, 1997; Ting & Riskin, 1994), and set theoretic approaches (Chang, Yu, & Lee, 2001). A Look-Up-Table (LUT) based algorithm proposed by Mese and Vaidyanathan (2001) produces very good results with low computational complexity. However, the LUT method uses the mean of corresponding contone values of existent binary input patterns to predict the contone value of non-existent binary input patterns. In other words, the LUT method creates the reconstructive function according to the training images. Three methods, Low-pass filtering, Hamming distance and Best linear estimator, are presented in Mese and Vaidyanathan (2001), and Best linear estimator is recommended in LUT. The problem can be regarded as a function approximation problem. It is natural to consider radial basis function neural network (RBFNN) because RBFNN is good for universal nonlinear approximation (Schilling, Carroll, & Al-Ajlouni, 2001). Thus, we propose a RBFNN model to improve the LUT inverse halftoning in this paper.

Since halftoning and inverse halftoning perform reverse operation, it is obvious that quality of inverse halftoning depends on the starting halftone method, and it is possible that the reconstruction quality can also be improved if the adaptive halftones are achieved. Besides, a reasonable conjecture that the efficiency of adopted halftoning and inverse-halftoning models would be improved in case their parameter values can be determined from a whole. In this paper, we make this conjecture realizable on the two adopted neural network models by performing an integrated learning procedure. As shown in Fig. 1, a single-layer perceptron neural network (SLPNN) is adopted for halftoning, and a radial-basis function neural network (RBFNN) is adopted for inverse-halftoning. The halftone images can be obtained by SLPNN halftoning and its corresponding contone images are reconstructed by RBFNN inverse halftoning. The integrated training procedure produces halftone image and the corresponding contone image at the same time, and all parameters of two neural networks are adjusted according to the error between original image and reconstructed image. More details will be presented in the later part of this paper.

This paper is organized as follows. The generic error diffusion method and equivalent SLPNN halftoning model are discussed in Section 2. The RBFNN inverse halftoning is proposed in Section 3. Then, the integrated learning algorithm of two adopted neural networks is presented in Section 4. Finally, experimental results and concluding remarks are presented in Sections 5 and 6, respectively.

2. Halftoning by single-layer perceptron neural network

The basic diagram of a generic error diffusion system is shown in Fig. 2, where $c[m,n]$ represents the input contone image, $u[m,n]$ is the updated pixel image, $a[m,n]$ is the adjustment matrix of pixel image, and $h[m,n]$ is the output halftone image determined by a threshold operation

$$h[m,n] = \begin{cases} 1, & \text{if } u[m,n] \geqslant t[m,n] \\ 0, & \text{otherwise} \end{cases}. \tag{1}$$

Here $t[m,n]$ is the threshold matrix and is specifically is equal to 0.5 for all $[m,n]$ in the generic error diffusion algorithm (Kang, 1999). The quantization error $e[m,n]$ is computed according to

$$e[m,n] = h[m,n] - u[m,n]. \tag{2}$$

This quantization error incurred from pixel $[m,n]$ is diffused forward onto unprocessed pixels on the right and below according to the weights in Fig. 3 with different modes in the error diffusion algorithm. The adjusted value of the pixel $[m,n]$ in back error compensation mode is collected by the error filter as follows:

$$a[m,n] = \sum_{ij} \left(\frac{w[i,j]}{\sum w[i,j]} \right) \cdot e[m-i, n-j], \tag{3}$$

where $w[i,j]$ is the generic error weighting matrix, satisfying $\sum_{ij} \frac{w[i,j]}{\sum w[i,j]} = 1$ in order to preserve the local tone. Therefore, the input to the quantizer can be represented as

$$u[m,n] = c[m,n] - a[m,n]. \tag{4}$$

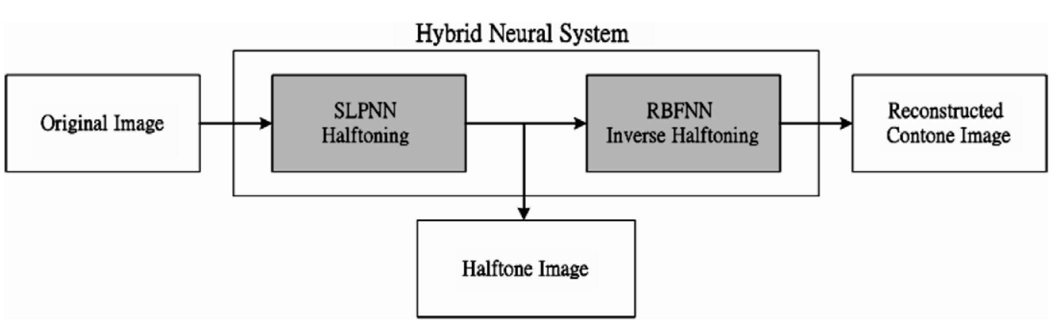

Fig. 1. Block diagram of the proposed system.