Two image denoising approaches based on wavelet neural network and particle swarm optimization

Yunyi Yan (闫允一) and Baolong Guo (郭宝龙)

ICIE Institute, School of Electro Mechanical Engineering, Xidian University, Xi'an 710071

Received September 7, 2006

Two image denoising approaches based on wavelet neural network (WNN) optimized by particle swarm optimization (PSO) are proposed. The noisy image is filtered by the modified median filtering (MMF). Feature values are extracted based on the MMF and then normalized in order to avoid data scattering. In approach 1, WNN is used to tell those uncorrupted but filtered by MMF and then the pixels are restored to their original values while other pixels will retain. In approach 2, WNN distinguishes the corrupted pixels and then these pixels are replaced by MMF results while other pixels retain. WNN can be seen as a classifier to distinguish the corrupted or uncorrupted pixels from others in both approaches. PSO is adopted to optimize and train the WNN for its low requirements and easy employment. Experiments have shown that in terms of peak signal-to-noise ratio (PSNR) and subjective image quality, both proposed approaches are superior to traditional median filtering.

OCIS codes: 100.0100, 110.4280, 100.7410.

Image denoising is a traditional and classic problem in image processing. The goal of image denoising is to remove noise from image while preserving as much as possible important features. Impulse noise is a familiar kind of noise caused by artificial reasons or errors in decoding. Traditional median filtering (TMF) is an often used and efficient method to remove this kind of noise. But TMF filters all pixels and many uncorrupted pixels are filtered and as a result the image will lose some fine details and may be over-smoothing. Many efforts have been done to reduce noise while overcoming the drawback of blurring image, such as nonlinear filtering[1], center weighted median filtering[2], switching median filtering[3], and besides median filtering. Wavelet has been introduced into image denoising in recent years for its ability of localization and approximation[4−7]. But most denoising based on wavelet cannot remove impulse noise as efficiently as Gaussian noise. Neural network has been in wide use for classification for its strong ability of learning[8,9]. As a combination of wavelet and neural network, wavelet neural network (WNN) has been in wide use for its approximation ability[10] and WNN has been applied in the fields of chaotic systems identification (CSI)[11], nonparametric estimation[12], speech signal processing[13] and contrast enhancement[14].

The reason that TMF blurs image is that TMF does not pick out what needs filtering but filter all pixels. In this paper, WNN and modified median filtering (MMF) derived from TMF are combined to remove impulse noise. Feature values will be extracted on the base of MMF results. Normalized feature values are input into WNN to determine whether a pixel is corrupted or not. The WNN is optimized by particle swarm optimization (PSO).

TMF often has a filter window with odd size, for example 3×3 or 5×5, and the result of TMF is the median value of all pixels in the filter window. But when the noise density is very high, saying over 20%, there are too many noisy pixels left in TMF results which reduce subjective quality and peak signal-to-noise ratio (PSNR) obviously. Experiments showed that most left noisy pixels were maximum or minimum pixels in the image. According to that, TMF are modified to MMF as

$$f^{\mathrm{M}}(i,j) = \begin{cases} E[f(i,j)]_{\Omega}, & f^{\mathrm{T}}(i,j) = f_{\max} \\ E[f(i,j)]_{\Omega}, & f^{\mathrm{T}}(i,j) = f_{\min} \\ f^{\mathrm{T}}(i,j), & \text{otherwise} \end{cases}, \tag{1}$$

where f represents the noisy image to be denoised; f^{M} represents MMF result image; f^{T} represents TMF result image; $E[f(i,j)]_{\Omega}$ represents the mean value of all pixels in the filter window Ω which centers on the pixel (i,j); $f_{\max}$ and $f_{\min}$ represent the maximum and the minimum values of the noisy image, respectively. MMF can be seen as the combination of median filtering and mean filtering.

General speaking, neural network seldom adopts raw data as its input for it often solves a problem according to a certain feature extracted. So the input of neural network is not the result of MMF but the feature value extracted by

$$e(i,j) = \frac{\left| f^{\mathrm{M}}(i,j) - f(i,j) \right|}{f^{\mathrm{M}}(i,j)}. \tag{2}$$

For a gray image with gray levels from 0 to 255, feature values extracted by Eq. (2) range from 0 to $+\infty$ which are too scattered and it is difficult for WNN to converge in such a wide range. To avoid data scattering and to converge rapidly, the feature values must be normalized, that is to say feature values should be mapped from $[0, +\infty)$ onto $(0, 1]$. In this paper, nonlinear normalization, in fact an exponent normalization, is adopted

$$\sigma = \exp(-\alpha e), \quad \alpha > 0. \tag{3}$$

More details on exponent normalization can be found in Ref. [15].

WNN takes wavelet functions as its activation functions. WNN can achieve fine approximation performance

because WNN is inspired by both neural network and wavelet decomposition[10]. The WNN, $N_{\rm Q}$ has the form

$$N_{\rm Q}(\sigma)=\sum_{k=1}^{K} w_k\varphi_k(\sigma)+\lambda_k=\sum_{k=1}^{K} w_k\varphi(\frac{\sigma-b_k}{a_k})+\lambda_k, \quad (4)$$

where $N_{\rm Q}(\sigma)$ represents the output with the input σ; $\varphi(\cdot)$ represents the mother wavelet activation function adopted; Q represents the parameter set and it can be shown as

$$\mathrm{Q}=\{(w_k,a_k,b_k,\lambda_k)|k=1,\cdots,K\}, \quad (5)$$

w_k represents the weight connecting the kth unit of the hidden layer to the output layer unit; a_k and b_k represent the dilation and the translation parameters, respectively; λ_k represents the output bias for the kth output unit which makes it easier to approximate nonzero average functions; $K\in\mathrm{N}$ represents the number of kernel neurons.

Total $4K$ parameters of wavelet neural network presented by Eq. (4) need to be optimized. After neural network established, its parameters must be optimized according to desired outputs. In our approaches, the desired outputs are defined as

$$S(i,j)=\begin{cases}1, & |g(i,j)-f(i,j)|>0\\ 0, & |g(i,j)-f(i,j)|=0\end{cases}, \quad (6)$$

where g represents the original image and f represents the image corrupted by noise. If the pixel (i,j) is corrupted, WNN is desired to output 1; otherwise output 0. Accordingly, if WNN outputs near 1, the corresponding pixels are likely to be corrupted; otherwise if near 0, they may be uncorrupted in all probability. It is not difficult to see that WNN works as a classifier.

The goal of optimization is to adjust the parameters (i.e. the element in Q) in order to make the error between obtained outputs and desired outputs as little as possible. Let $A[\cdot]$ represent optimization operator, the optimization or training of WNN can be represented as

$$N_{\rm Q}=A[N_{\rm Q}], \quad s.t. \quad \min\{\|S-N_{\rm Q}\|\}. \quad (7)$$

The optimization will be accomplished by PSO algorithm.

PSO as a member of swarm intelligence with low hardware requirements and easy employ has been proved to be efficient for many global optimization problems and overcomes some shortages of other evolution computation methods[16,17]. The main idea behind PSO is that social sharing of information among the individuals of a population may provide an evolutional advantage.

Suppose that the search space is D-dimension, and $D=4K$ according to Eq. (5). Each parameter is thought as one dimension. The ith particle of the population (i.e. the swarm) can be presented by a D-dimensional vector as

$$X_i=(x_{i1},x_{i2},\cdots,x_{iD})^{\rm T}. \quad (8)$$

Each particle presents a potential solution, and the solution has D-dimensions. Each particle (solution) means a point in D-dimensional space. The population is thought as the set of potential solutions. The goal of PSO algorithm is to find the best point satisfying Eq. (7).

The velocity (i.e. the position change) of the ith particle can be represented by another D-dimensional vector as

$$V_i=(v_{i1},v_{i2},\cdots,v_{iD})^{\rm T}. \quad (9)$$

Two important variants play an important role in PSO. One is the best previously visited position of the ith particle which can be denoted as

$$P_i=(p_{i1},p_{i2},\cdots,p_{iD})^{\rm T}. \quad (10)$$

The other is the best particle of the swarm which is indexed by g and can be denoted as

$$P_g=(p_{g1},p_{g2},\cdots,p_{gD})^{\rm T}. \quad (11)$$

In PSO, "best particle" means the most wanted result, say, the result satisfying minimum mean square error (MSE).

Let the superscripts $n\in\mathrm{N}$ denote the iteration number and the swarm can be manipulated according to

$$v_{id}^{n+1}=\omega v_{id}^{n}+c_1r_1^{n}(p_{id}^{n}-x_{id}^{n})+c_2r_2^{n}(p_{gd}^{n}-x_{id}^{n}), \quad (12)$$

$$x_{id}^{n+1}=x_{id}^{n}+\lambda v_{id}^{n+1}, \quad (13)$$

where $d=1,2,\cdots,D\in\mathrm{N}$; $i=1,2,\cdots,N\in\mathrm{N}$, and N is the size of swarm (population); $n=1,2,\cdots,G\in\mathrm{N}$ determines the iteration number, and G is the maximum iteration given before $c_1\in\mathrm{R}$ and $c_2\in\mathrm{R}$ are positive constants called cognitive and social parameters respectively which can be used to adjust the impact intensity of P_i or P_g; $r_1\in\mathrm{R}$ and $r_2\in\mathrm{R}$ are random numbers distributed uniformly in [0, 1]; $\omega\in\mathrm{R}$ is called inertia weight and $\lambda\in\mathrm{R}$ is called constriction factor which is used alternatively to $\omega\in\mathrm{R}$ to limit velocity. For PSO, the iterative process will continue until a satisfied result obtained or the maximum iteration reached.

It is worth noticing that velocity should be limited by a threshold $V_{\max}$ which is very crucial because too large velocity can cause particle moving past good solutions. And on the other hand, too small velocity also can cause insufficient exploration of the solution space. In fact, it is better to locate optimum area in large velocity and then continue to search in the area finely in small velocity. Velocity adjustment can be achieved by adjusting the parameters: ω, λ, and $V_{\max}$.

As a classifier, the task of WNN is to map corrupted pixels to a number near 1 and to map uncorrupted pixels to a number near 0. There are two kinds of pixels in a noisy image: corrupted and uncorrupted. Accordingly two denoising approaches can be established.

Approach 1 (App1): WNN is used to distinguish the uncorrupted pixels from others in MMF result image. The distinguished pixels by WNN according to a threshold $T_1\in[0,1]$ are replaced by their original values in the noisy image. Let R_1 represent the result image obtained by App1, then App1 can be summarized as

$$R_1(i,j)=\begin{cases}f(i,j), & \text{if } N_{\rm Q}(\sigma)\le T_1,\\ f^{\rm M}(i,j), & \text{otherwise,}\end{cases} \quad T_1\in[0,1]. \quad (14)$$

Approach 2 (App2): WNN is used to distinguish the corrupted pixels from others in noisy image. The distinguished pixels should be replaced by the corresponding MMF results, other pixels should keep unchanged. Like App1, App2 can be summarized as

$$R_2(i,j) = \begin{cases} f^{\mathrm{M}}(i,j), & \text{if } N_{\mathrm{Q}}(\sigma) \geq T_2, \\ f(i,j), & \text{otherwise,} \end{cases} \quad T_2 \in [0,1]. \quad (15)$$

It is noticeable that the two approaches can remove noise in noisy image independently. Experiments show that it is better to set T_1 in the interval [0.22, 0.29] and T_2 in [0.30, 0.40]. In fact, the two approaches App1 and App2 perform two different strategies: App1 picks those pixels which are uncorrupted but are filtered as noisy points by mistake; App2 only picks those corrupted pixels and then others will be filtered.

Mexican hat function was selected as the activation function or the mother wavelet function which is defined as

$$\varphi(t) = (1-t^2)\mathrm{e}^{-t^2/2}. \quad (16)$$

The number of kernel neurons was set to 3 and the parameters of WNN used in the two approaches were listed in Table 1. The parameters in PSO were listed in Table 2. More details on parameter selection and convergence in PSO can be founded in Ref. [18].

To overview the total image denoising process, Fig. 1 shows the diagram of the optimization process of WNN and relative symbols appeared in this paper are shown in brackets; Figure 2 shows the diagram of the process of optimized WNN denoising image. And to evaluate the performance of App1 and App2 based on WNN, experiments were conducted on several 256-leveled gray images like Lena, Elaine, Cameraman, Peppers, Tree, Bridge and House (512×512) at noise densities from 0.05 to 1.00. The image Lena with 20% noise was used to train and optimize the WNN and other images were used to evaluate the performance. A comparison between App1, App2, two-dimensional (2D) TMF (with 3×3 filtering window) was made.

Table 1. Parameters Used in WNN

w_1	a_1	b_1	λ_1
0.26122	0.29464	−0.46307	0.20293
w_2	a_2	b_2	λ_2
0.67433	−0.60032	−0.013365	−0.38569
w_3	a_3	b_3	λ_3
0.28135	−0.6167	−0.21892	0.52764

Table 2. Parameter Selection in PSO

Parameter	Value
Swarm Size (N)	20
Search Space Dimension (D)	12
Maximum Iteration (G)	200
Cognitive Parameter (c_1)	2.0
Social Parameter (c_2)	2.0
Inertia Weight (ω)	0.8
Constriction Factor (λ)	1.0
Maximum Velocity ($V_{\max}$)	3.0

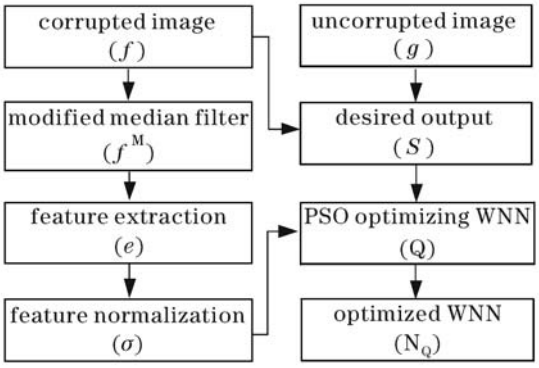

Fig. 1. Optimizing WNN by PSO.

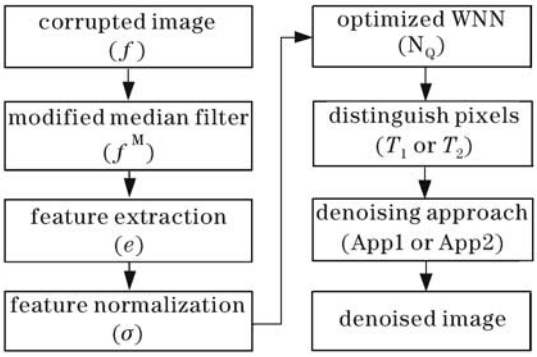

Fig. 2. Image denoising by optimized WNN.

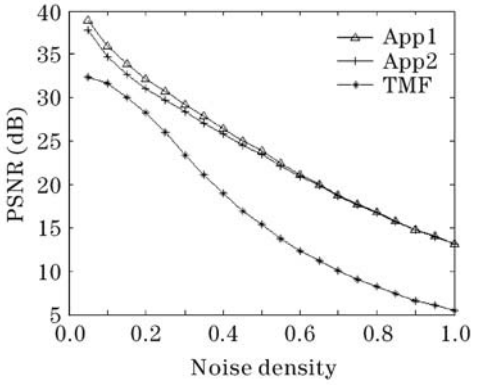

Fig. 3. Denoising results (PSNR) on Elaine.

PSNR was selected to compare the performance quantitatively. Part of the experiment results, some PSNRs obtained by three methods are listed in Table 3. And to observe the PNSRs visually, the PSNR results of the image Elaine are shown in Fig. 3. And Fig. 4 shows the denoising results by different methods on the image Elaine.

In Fig. 3 and Table 3, it is not difficult to see that both App1 and App2 are superior to TMF in terms of PSNR, especially when the image is highly corrupted. TMF has left quite a few noticeable corrupted pixels in Fig. 4(c). App1 and App2 are very close in denoising performance, which can be seen according to the subjective image qualities of Figs. 4(d) and (e). More other experiments

Table 3. Experimental Results (PSNR: dB)

	Noise Density (%)					
	5	10	20	30	50	70
			Elaine			
TMF	32.40	31.64	28.27	23.38	15.35	10.05
App2	37.76	34.78	31.08	28.36	23.43	18.68
App1	38.90	35.96	32.20	29.25	23.84	18.81
			Cameraman			
TMF	30.37	29.63	26.72	22.71	14.98	9.91
App2	34.26	32.68	29.64	27.23	22.98	18.11
App1	34.71	32.78	29.60	27.12	22.61	17.98
			Peppers			
TMF	34.30	33.05	28.27	23.27	15.12	10.04
App2	37.52	34.56	30.70	27.84	22.70	18.42
App1	38.67	36.00	31.94	28.80	22.82	18.48

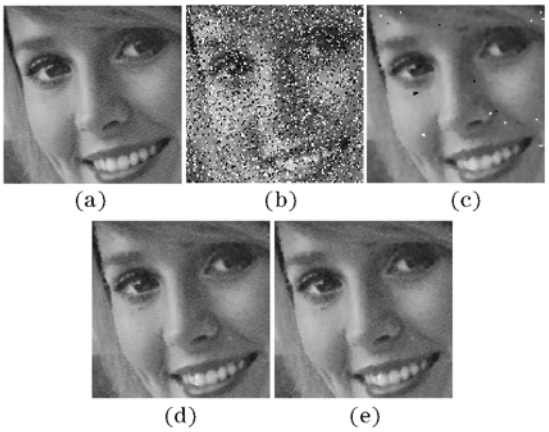

Fig. 4. Denoising results on Elaine (local 128×128 with 20% noise). (a) Original image; (b) noisy image; (c) TMF denoising, PSNR = 28.27 dB; (d) App2 denoising, PSNR = 31.08 dB; (e) App1 denoising, PSNR = 32.20 dB.

besides the listed in paper, like Tree and Bridge, have also proved that.

Wavelet neural network is introduced into image denoising in this paper. WNN is used to distinguish the corrupted pixels or uncorrupted pixels from others. Two approaches based on WNN are proposed. In App1, corrupted pixels distinguished by WNN are replaced by MMF results. And uncorrupted pixels distinguished by WNN are restored to their original values in App2. The optimization of WNN is achieved by PSO. Experimental results show that both the two approaches proposed in this paper achieve much better performance in terms of PSNR and better subjective image quality. And further effort will be conducted on multiwavelet neural network (MWNN).

This work was supported by the National Natural Science Foundation of China under Grant No. 60572152. Y. Yan's e-mail address is yyyan@xidian.edu.cn.

References

1. J. Dong and J. Zhang, J. Optoelectron. Laser (in Chinese) **14,** 1336 (2003).
2. S.-J. Ko and Y. H. Lee, IEEE Trans. Circuit and System **38,** 984 (1991).
3. H.-L. Eng and K.-K. Ma, IEEE Trans. Image Processing **10,** 242 (2001).
4. L. B. Montefusco and S. Papi, BIT Numerical Mathematics **43,** 611 (2003).
5. S. G. Chang, B. Yu, and M. Vetterli, IEEE Trans. Image Processing **9,** 1532 (2000).
6. Y. Tan, J. Tian, and J. Liu, Chin. Opt. Lett. **4,** 80 (2006).
7. P. Mrázek and J. Weickert, Lect. Not. Comp. Sci. **2781,** 156 (2003).
8. L. M. Silva, J. M. de Sa, and L. A. Alexandre, in *Proceedings of ESANN 2005* (2005) p.217.
9. S. Sohn and C. H. Dagli, Neural Processing Lett. **19,** 191 (2004).
10. Q. Zhang and A. Benveniste, IEEE Trans. Neural Networks **3,** 889 (1992).
11. L. D. S. Coelho and R. Calixto, Advances in Soft Computing **1,** 205 (2005).
12. Q. Zhang, IEEE Trans. Neural Networks **8,** 227 (1997).
13. F. Chen, D. Shi, and G. S. Ng, Lecture Notes in Artificial Intelligence **3060,** 406 (2004).
14. C. Zhang, X. Wang, and H. Zhang, Chin. Opt. Lett. **3,** 636 (2005).
15. Y. Yan and B. Guo, J. Comp. Inf. Syst. **2,** 625 (2006).
16. K. E. Parsopoulos and M. N. Vrahatis, Natural Computing **1,** 235 (2002).
17. J. F. Schutte and A. A. Groenwold, J. Global Optimization **31,** 93 (2005).
18. M. Clerc and J. Kennedy, IEEE Trans. Evolutionary Computation **6,** 58 (2002).