Journal of Digital Imaging (2019) 32:148–161
https://doi.org/10.1007/s10278-018-0110-y

An Automatic Parameter Decision System of Bilateral Filtering with GPU-Based Acceleration for Brain MR Images

Herng-Hua Chang[1] · Yu-Ju Lin[1] · Audrey Haihong Zhuang[2]

Published online: 7 August 2018

Abstract
Bilateral filters have been extensively utilized in a number of image denoising applications such as segmentation, registration, and tissue classification. However, it requires burdensome adjustments of the filter parameters to achieve the best performance for each individual image. To address this problem, this paper proposes a computer-aided parameter decision system based on image texture features associated with neural networks. In our approach, parallel computing with the GPU architecture is first developed to accelerate the computation of the conventional bilateral filter. Subsequently, a back propagation network (BPN) scheme using significant image texture features as the input is established to estimate the GPU-based bilateral filter parameters and its denoising process. The k-fold cross validation method is exploited to evaluate the performance of the proposed automatic restoration framework. A wide variety of T1-weighted brain MR images were employed to train and evaluate this parameter-free decision system with GPU-based bilateral filtering, which resulted in a speed-up factor of 208 comparing to the CPU-based computation. The proposed filter parameter prediction system achieved a mean absolute percentage error (MAPE) of 6% and was classified as "high accuracy". Our automatic denoising framework dramatically removed noise in numerous brain MR images and outperformed several state-of-the-art methods based on the peak signal-to-noise ratio (PSNR). The usage of image texture features associated with the BPN to estimate the GPU-based bilateral filter parameters and to automate the denoising process is feasible and validated. It is suggested that this automatic restoration system is advantageous to various brain MR image-processing applications.

Keywords Image denoising · Image texture · CUDA · Neural networks · Bilateral filter · Automation

Introduction

During the acquisition process, images acquired from magnetic resonance imaging (MRI), functional MRI (fMRI), diffusion-weighted MRI (DWI), and diffusion tensor MRI (DTI) are inevitably corrupted by various sources of random noise, which includes physiological motion and patient movement as well as eddy-current distortions and instabilities of the MRI scanning hardware. Such corruption introduces different levels of ambiguities in the formation and measurement of quantitative attributes that hinders the understanding and estimation of essential characteristics in the inspected tissues. In addition, the noise not only disturbs the visual inspection in medical diagnostic procedures but also deteriorates a number of computerized processes such as structure visualization, image segmentation, image registration, and tissue classification [1].

One typical solution to this dilemma is to directly average multiple acquisitions in the scanner and consequently reinforcing the signal-to-noise ratio (SNR). However, it requires a longer acquisition time to achieve a desired SNR at high spatial resolution. Practically, this time is restricted due to consideration of system throughput, patient comfort, and physical constraints arising in dynamic applications such as fMRI. As such, there is a pragmatic limitation on the SNR of acquired MR images in most applications [2]. In the postprocessing stages for brain MR images, noise reduction approaches have been conventionally applied in a wide variety of subsequent processing applications.

✉ Herng-Hua Chang
herbertchang@ntu.edu.tw

[1] Computational Biomedical Engineering Laboratory (CBEL), Department of Engineering Science and Ocean Engineering, National Taiwan University, 1 Sec. 4 Roosevelt Road, Daan, Taipei 10617, Taiwan

[2] Department of Radiation Oncology, Keck Medical School, University of Southern California, Los Angeles, CA, USA

In spite of notable advances in imaging techniques, noise reduction in brain MR images remains challenging. This is mainly because the nonlinear operation on the real and imaginary components in the *k*-space makes the noise in magnitude MR images Rician distributed, which is signal-dependent [3]. Consequently, in single-coil systems, the noise in magnitude MR images is modeled as a Rician distribution [4]. A number of image denoising algorithms based on the Rician noise model have been proposed [5–9]. In particular, Aja-Fernandez et al. [5] proposed a close-form solution of the linear minimum mean square error (LMMSE) estimator for the Rician distribution and demonstrated its effectiveness in noise removal and feature conservation. Pizurica et al. [10] introduced a versatile wavelet domain (VWD) method for noise filtration. By empirically estimating image features and noise types, the authors demonstrated its usefulness in noise suppression. Nonetheless, for its simplicity, the Gaussian filter has been extensively adopted in many MR image processing applications [7, 11, 12]. Intrinsically, this filter calculates a weighted average of pixel intensities in the neighborhood so that the weight diminishes with the distance from the kernel center. Although Gaussian filters smooth noise satisfactorily, edges are blurred indubitably.

With intrinsically nonlinear characteristics, the anisotropic diffusion filter [13] has been proposed. This approach averages pixel intensities from the neighborhood, whose shape and dimension relies upon local image fluctuations that are estimated at every point [14]. On the other hand, the bilateral filter [15] is a promising technique that adopts a non-iterative strategy for edge preserving and overcomes the shortages of Gaussian filters for brain MR image restoration. Extending the concept of the Gaussian filter, this framework aims to associate gray value similarity in the range domain with geometric closeness in the spatial domain as a nonlinear filter for image restoration. The bilateral filter has been shown performing adequately in various image noise removal applications, which made it the subject of many further studies [16–20].

For example, Walker et al. [21] compared the usage of Gaussian filters and bilateral filters in smoothing fMRI data and concluded that the bilateral filter more precisely located brain activation and increased the understanding of activation bordering sharp transitions. Rydell et al. [22] presented bilateral filtering of fMRI with improved detection of activated regions by manually tuning all parameters to optimize the performance. Hamarneh and Hradsky [23] extended the bilateral filter to perform edge-preserving smoothing of DTI data. McPhee et al. [24] demonstrated the treatment of the bilateral filter in improving image characters at the cortical areas without demolishing contrast in the brain. Recently, Jaramillo et al. [25] proposed a wavelet domain bilateral filter to reduce the noise in MR images. Wells and Dobbins [26] implemented the bilateral filter to study the amplitude modulation and waveform distortion properties of nonlinear systems. Kala and Deepa [27] introduced a fuzzy membership function into the bilateral filtering scheme to improve the denoising efficiency.

As objects consist of various kinds of texture appearances, image texture has been playing an essential role in image processing and pattern recognition [28]. In essence, those texture features can reveal typical regularities of biological structures in brain MR images [29]. Depending on the parameter values being used, the restoration process may blur edges of important anatomy structures while removing noise as illustrated in Fig. 1. Unfortunately, empirical settings and manual adjustments have been adopted in many existing bilateral filtering methods for MR image restoration [26, 30]. To facilitate the burdensome adjustments of the filter parameters, some approaches adopted a fixed parameter value in the spatial domain while relating the parameter in the range domain to the noise variance [21, 24, 25]. Optimization of the bilateral filter parameters remains an open question. Strategies through the aid of artificial intelligence techniques seems an achievable solution. Among the diverse machine learning approaches, artificial neural networks (ANNs) [31] are exceptionally appropriate for this ambition.

An ANN is a computational prototype or mathematical model that is motivated by the structure and/or functional perspective of biological neural networks. Consisting of an interconnected aggregate of artificial neurons, a neural

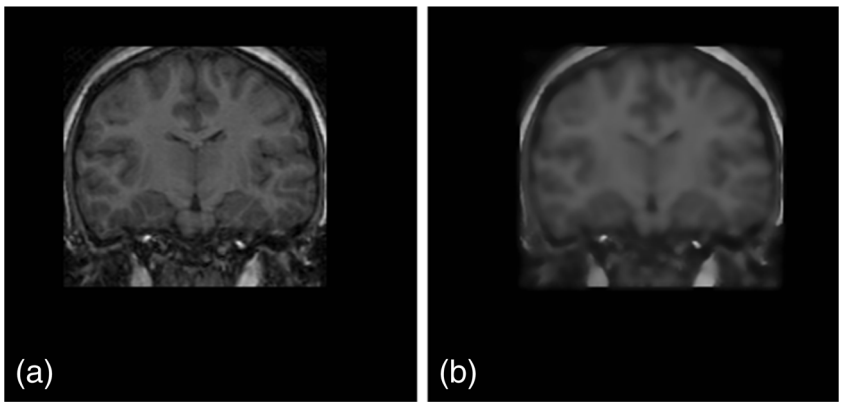

Fig. 1 Illustration of the problem of manual restoration process. **a** Noisy brain MR image. **b** Denoised image with heavily blurred structures due to improper parameter settings

network encompasses many links that connect neurons to both inputs and outputs. Being regarded as a distributed, parallel, and adaptive information processing scheme, the neural networks perform the estimation of optimal outputs. ANNs have been extensively exploited in many medical applications both academically and clinically. For example, Younis et al. [1] presented an artificial immune-activated neural network that is controlled through an energy measure to ensure accurate recognition. This model was applied to segment volumetric brain MRI data and achieved high accurate results, especially at low levels of noise. Virmani et al. [32] proposed a neural network ensemble-based computer-aided diagnostic system to assist radiologists to differentiate focal liver lesions. The principal component analysis (PCA) was adopted to reduce the dimensionality of the feature space prior to classifier design. Wang et al. [33] introduced ANN modeling to estimate volumetric breast density from full-field digital mammography.

To the best of our knowledge, the application of ANNs to estimate filter parameters and automate filtering processes is rare in the literature. Indeed, an essential key to a successful bilateral filtering system based on ANNs is the quality as well as the quantity of input arguments. This brings on a fundamental problem of exploring relevant characteristics among a wide variety of image texture features for the automation process, which has been investigated in our previous study [34]. Another critical issue is the considerable amount of repeated computations of the bilateral filter in the training phase. One way to accelerate the filter computation is through the employment of parallel computing based on the graphics processing unit (GPU), which is known as general purpose computation on GPU (GPGPU) [35]. The adoption of the GPU as an alternative computation platform rather than the central processing unit (CPU) is to achieve acceleration for computationally excessive tasks that are intrinsically parallel and have neighboring data accesses for each data element. All in all, image processing has been one of the major applications that involve high computing complexity, especially for image filtering algorithms [36].

The objective of this article is in an attempt to investigate the automation as well as the acceleration of the conventional bilateral filter. A GPU-based bilateral filtering framework with parallel computing is uniquely designed. We address optimization techniques to accelerate computation in either memory resources or thread usages. Based on the investigation of significant texture features in our earlier study, a fully automatic and parameter-free restoration framework for the accelerated bilateral filter associated with ANNs is developed and established. Finally, a wide variety of brain MR images are utilized to validate the proposed scheme regarding acceleration performance and restoration efficacy.

The major contributions of the current work are summarized as follows:

1. A wide variety of image texture features including wavelet transform and gray level variation methods have been investigated regarding the discrimination ability between brain MR slices.
2. A GPU-based acceleration technique based on the usage of shared memory for bilateral filtering has been developed.
3. An efficient ANN model for training and predicting the bilateral filter parameters has been established.
4. The connection between significant image texture features and the ANN system for filter parameter prediction has been created.
5. Extensive experiments with massive MR images in comparison with several state-of-the-art methods have been conducted to evaluate the proposed framework.

Analysis of Image Texture Features

We shall start by describing the exploitation of image texture features for the automation of bilateral filtering. Three possible categories with different aspects of texture feature extraction are considered, which has been investigated in our earlier study [34]. The first category belongs to fundamental statistic features including mean, standard deviation, variance, and entropy [37], where the original brain MR image is directly adopted for the computation. The second category is composed of wavelets [38] as described in the subsequent section.

2D Wavelet Transform

Wavelet transforms present a marvelous mechanism for feature extraction with the representation of an image at different levels of scales [39]. Traditionally, wavelet coefficients have been widely adopted to estimate noise variance, which are favorable feature candidates for the proposed filter parameter decision task. Among various families of wavelets, the Haar wavelet transform is utilized for its simplicity and effectiveness [40]. Introducing a family of wavelet functions and its associated scaling functions, the hierarchical wavelet transforms decompose an image in the spatial domain into various subbands in the frequency domain.

To overcome intensity varieties across various brain MR images, the input of the wavelet transform in this study is a normalized image defined as

$$\hat{I}(i,j) = \frac{I(i,j)}{\sqrt{\sum_{m=1}^{M}\sum_{n=1}^{N} I(m,n)^2 / MN}} \quad (1)$$

where $I(i,j)$ is the original $M \times N$ image and $\hat{I}(i,j)$ is the normalized image. The resulting 2D array of the wavelet coefficients comprises four subbands of the transformed data. These four subbands are labeled as LL_1 (low-low), HL_1 (high-low), LH_1 (low-high), and HH_1 (high-high), where the subscript 1 indicates that they are sequentially obtained by the first-order vertical and horizontal transformations. Each subband contains different levels of resolution with LL representing the approximate image, HL the horizontal details, LH the vertical details, and HH the diagonal details. The decomposition procedure is repeatedly exploited to the LL_1 subband to produce the next level of the hierarchy, namely, LL_2, LH_2, HL_2, and HH_2, where the subscript 2 represents the second-order transformations.

Three different kinds of energy measures are subsequently computed for both order subbands using [39]

$$\text{Norm}-1 \text{ energy} : e_1 = \frac{1}{M_s N_s} \sum_{m=1}^{M_s} \sum_{n=1}^{N_s} |x(m,n)| \tag{2}$$

$$\text{Norm}-2 \text{ energy} : e_2 = \frac{1}{M_s N_s} \sum_{m=1}^{M_s} \sum_{n=1}^{N_s} |x(m,n)|^2 \tag{3}$$

$$\text{Standard deviation} : e_3 = \frac{1}{M_s N_s} \sqrt{\sum_{m=1}^{M_s} \sum_{n=1}^{N_s} |x(m,n) - \bar{x}|^2} \tag{4}$$

where M_s and N_s are the dimension of the subband, $x(m,n)$ is the subband under consideration with $1 \le m \le M_s$ and $1 \le n \le N_s$, and $\bar{x}$ is the arithmetic mean of $x(m,n)$. With 8 subbands and 3 energy signatures, 24 different features are computed in this category.

Gray-Level Co-Occurrence Matrix

The gray level co-occurrence matrix (GLCM) [41] constitutes the final category, where second-order statistics are extracted based on the replicated occurrence of some gray-level composition in an image. The essential characteristic is that with respect to distance the GLCM varies quickly in fine texture images and slowly in coarse texture images. GLCM-based texture features have been exploited for detection, classification, and segmentation missions such as discriminating benign areas from malignant lesions [42].

From the perspective of mathematics, the GLCM is a matrix of repetitions at which two pixels are separated by a distance vector in an image. For a brain MR image of L gray levels, the allocation in the $L \times L$ matrix is built upon the gray tone spatial relationship between two pixels with a specified distance and angle using

$$G(i,j) = \sum_{x=1}^{W_x - d_x} \sum_{y=1}^{W_y - d_y} \begin{cases} 1, & \text{if } F(x,y) = i \text{ and } F(x+d_x, y+d_y) = j \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $G(i,j)$ is the quantized gray tone at location (i,j) with $i = 0, 1, \ldots, L-1$ and $j = 0, 1, \ldots, L-1$, $F(x,y)$ is the gray level intensity in the kernel at (x,y), W_x and W_y are the lengths of the resolution kernel in the image ordered by their row-column assignations, and d_x and d_y are the spatial separation between two computed pixels defined by a distance d and an angle θ from the kernel origin.

Varying the distance vector in Eq. (5) enables the extraction of different image texture characteristics. There are eight adjoining pixel-pairs in four isolated directions with $\theta = 0°$, 45°, 90°, and 135° for the distance, d. The matrix of relative frequencies $P(i,j)$ is defined as the probability of the pixel-pair occurrence of two correlated gray levels i and j normalized by the total counts as

$$P(i,j) = \frac{G(i,j)}{\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} G(i,j)} \tag{6}$$

To achieve best discriminating abilities and maximal distinguishing strength, the difference image I_D between the input image I and its Gaussian-filtered image I_G is adopted as the input image for the computation. This difference image roughly represents the gray level edges of anatomical structures in the brain. Once $P(i,j)$ is created, four groups of features are computed: statistics, information theory, information measures of correlation, and visual texture characteristics [42–45]. In short, there are 8 basic features multiplied by 4 directions that result in 32 different texture features in this category.

Significant Texture Features

According to the three image texture feature categories, a large number of 60 different features are computed in each individual image. To reduce the complexity of the problem and to realize the most important texture features, a paired-samples t test [46] is applied to every feature for evaluating the distinguishing capacity in both noise levels and slice positions. The noise level reflects the degree of Rician noise in the image, and the slice position indicates the variation of brain tissues in the image. Based on our experience, the noise level alone is not adequate to produce the best filter parameter values for slices that have exactly the same noise variance. This is because that the brain has quite various anatomical structures and sizes among the slice positions. For example, the best filter parameters for the middle slice are usually not the best filter parameters for other slices with the same noise variance.

In order to obtain the significant texture features, each of the 60 texture feature candidates is computed in every image. The p value of all features in each two-image set is calculated using the paired-samples t test procedure. The

two compared images are sequentially selected based on either different noise levels or distinct slice positions. Subsequently, the average p value for each feature in either group is computed and ranked. To understand the optimal combination and the corresponding features, the decision tree of the classification and regression tree (CART) algorithm [47] is exploited to 34 out of 60 features, whose average p values are less than 0.05. Our earlier study [34] suggests that the seven features of $e1(LL_1)$, $e1(LL_2)$, $e2(HL_1)$, $e3(LH_1)$, $e3(HL_2)$, CON(90°), and DIS(90°) accomplish the most accurate results among all combinations. The first five features belong to the wavelet category and the last two features are in the GLCM category, where CON represents contrast with

$$\mathrm{CON} = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} P(i,j) \times (i-j)^2 \tag{7}$$

and DIS represents dissimilarity with

$$\mathrm{DIS} = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} |i-j| P(i,j) \tag{8}$$

Acceleration and Automation of Bilateral Filtering

Rician Noise and Bilateral Filter

As described previously, Rician noise exists in single-coil scanned MR images. When the SNR approaches zero, the Rician distribution simplifies to a Rayleigh distribution. On the contrary, when the SNR is high, the Rician distribution becomes a Gaussian distribution. Thereupon, in the low SNR regions such as the background in an MR image, the noise is apt to be a Rayleigh distribution; while in the high SNR regions such as the brain structures, the noise can be modeled as a Gaussian distribution. Since the denoising purpose is mainly to improve the SNR in the brain structures, filters based on the Gaussian noise model have been exploited in the MR image denoising framework as is the bilateral filter [15].

The bilateral filter integrates gray levels in favor of near values to distant values in both range and domain. Let (θ_x, θ_y) θ_x, θ_y be the position of the pixel centered in a $(2N_k + 1) \times (2N_k + 1)$ neighborhood with N_k a positive integer and

$$\Psi_{\theta_x,\theta_y} = \left\{ \left(\mu_x, \mu_y\right) : \left(\mu_x, \mu_y\right) \in [\theta_x - N_k, \theta_x + N_k] \times [\theta_y - N_k, \theta_y + N_k] \right\} \tag{9}$$

be the pixels in the neighborhood of (θ_x, θ_y). A Gaussian function with respect to the Euclidean distance between the arguments is utilized for the spatial component $W^S_{\theta_x,\theta_y}$ and the radiometric component $W^R_{\theta_x,\theta_y}$, which are defined as

$$W^S_{\theta_x,\theta_y}\left(\mu_x, \mu_y\right) = \exp\left[-\frac{\left|\left(\mu_x, \mu_y\right) - \left(\theta_x, \theta_y\right)\right|^2}{2\sigma_S^2} \right] \tag{10}$$

and

$$W^R_{\theta_x,\theta_y}\left(\mu_x, \mu_y\right) = \exp\left[-\frac{\left|I\left(\mu_x, \mu_y\right) - I\left(\theta_x, \theta_y\right)\right|^2}{2\sigma_R^2} \right] \tag{11}$$

where $I(\cdot, \cdot)$ is the intensity of the input image at the given position $(\cdot, \cdot)$.

In Eqs. (10) and (11), the parameters σ_S and σ_R are adopted to adjust the influence of $W^S_{\theta_x,\theta_y}$ and $W^R_{\theta_x,\theta_y}$, respectively. They can be treated as approximate thresholds for classifying neighboring pixels that are adequately close or similar to the center pixel. The overall weight $W^{SR}_{\theta_x,\theta_y}$ is defined as

$$W^{SR}_{\theta_x,\theta_y}\left(\mu_x, \mu_y\right) = W^S_{\theta_x,\theta_y}\left(\mu_x, \mu_y\right) W^R_{\theta_x,\theta_y}\left(\mu_x, \mu_y\right) \tag{12}$$

Essentially, the image is filtered and normalized by the sum of the ensemble weight $W^{SR}_{\theta_x,\theta_y}$ using

$$\tilde{I}\left(\theta_x, \theta_y\right) = \frac{\sum_{\left(\mu_x,\mu_y\right)\in\Psi} W^{SR}_{\theta_x,\theta_y}\left(\mu_x, \mu_y\right) I\left(\mu_x, \mu_y\right)}{\sum_{\left(\mu_x,\mu_y\right)\in\Psi} W^{SR}_{\theta_x,\theta_y}\left(\mu_x, \mu_y\right)} \tag{13}$$

where $\tilde{I}\left(\theta_x, \theta_y\right)$ is the restored image at location (θ_x, θ_y). The computation complexity with the CPU implementation is $O\left(N_k^2 MN\right)$. If $M = N$, the complexity is quadratic with respect to the product of the kernel length and the image width, i.e., $O\left(N_k^2 M^2\right)$.

GPU-Based Bilateral Filtering with CUDA

To address the computational burden in the ANN training phase with the bilateral filter, we propose the employment of the compute unified device architecture (CUDA) technology, which is a special programming architecture for GPGPU on NVIDIA graphics cards [48]. A CUDA compliant device is a set of multiprocessor cores that is capable of concurrently executing a huge number of threads. Every CUDA enabled GPU provides several different types of memory. Registers are the fastest but have a limited amount of space around 32–64 KB. Global memory supports a much larger space but is sluggish so that the memory latency affects the performance. A compromise between these two is a parallel cache of shared memory, which is shared by all cores in each multiprocessor and is limited to 16–64 KB [49]. In CUDA design, the challenge is how to optimally manipulate shared memory to achieve the maximum performance.

Before dealing with the employment of CUDA, recall the spatial component in Eq. (10), which indicates numerous duplicated computations of the distance in the numerator. To reduce the computational load, Eq. (10) is reformulated as

$$DS(d_x, d_y) = \exp\left[-\frac{{d_x}^2 + {d_y}^2}{2\sigma_S^2}\right] \tag{14}$$

where $0 \le d_x = |\mu_x - \theta_x| \le N_k$ and $0 \le d_y = |\mu_y - \theta_y| \le N_k$ are the spatial distances in the x- and y-axes, respectively. To make use of shared memory, all possible spatial component values are computed in advance and stored in a memory buffer defined as

$$WSB(d_x, d_y) = \{DS(d_x, d_y) | d_x, d_y \in [0, \ldots, N_k]\} \tag{15}$$

where WSB is the spatial weight buffer with respect to N_k for looking up. Henceforward, the spatial component is rapidly obtained by looking up the weight buffer based on the distances, which is stored in shared memory.

To further accelerate the computation, GPU-based parallel computing with CUDA architecture is developed. Rather than executing the filter one pixel after another, a large number of filtering pixels are executed simultaneously. In our approach, the image pixels are split into several separate blocks indexing $block(i,j)$, which has its own shared memory that can be accessed by the 16×16 threads in each block. As the bandwidth of shared memory is wider than regular GPU memory, and the access time is faster and closer to cache, it is advantageous to manipulate shared memory to avoid repeated accesses to the same pixel element from different threads and to raise read/write speed. Accordingly, the WSB in Eq. (15) is stored in shared memory of each block for rapid computation. As illustrated in Fig. 2 for a 5×5 filter with $N_k = 2$, the spatial component $DS(1, 1)$ is retrieved from WSB by computing the distance between $(\theta_x, \theta_y) = (2, 2)$ and $(\mu_x, \mu_y) = (3, 3)$. The obtained spatial component weight is then utilized for computation in a parallel fashion with CUDA architecture using

$$\tilde{I}(\theta_x, \theta_y) = \frac{\sum_{(\mu_x,\mu_y)\in\Psi} WSB(d_x, d_y) W^R_{\theta_x,\theta_y}(\mu_x, \mu_y) I(\mu_x, \mu_y)}{\sum_{(\mu_x,\mu_y)\in\Psi} WSB(d_x, d_y) W^R_{\theta_x,\theta_y}(\mu_x, \mu_y)} \tag{16}$$

where Eq. (16) is the shared memory-based accelerated bilateral filter of Eq. (13).

The pseudo code of the proposed GPU-based algorithm is presented in Table 1. In this CUDA implementation of the bilateral filter, the computation complexity of a single thread is $O(N_k^2)$. Practically, the computation time is inversely proportional to the number of threads running in parallel in the GPU, which has a complexity of $O(N_k^2 MN/T)$, where T is the maximum number of concurrent threads executing in the GPU. In a perfect GPU that is able to spawn MN threads executing in parallel, the complexity of the GPU-based bilateral filter reduces to $O(N_k^2)$. Nevertheless, the ensemble computation time includes the running time of threads, memory transfer latencies, and thread block and grid block management latencies.

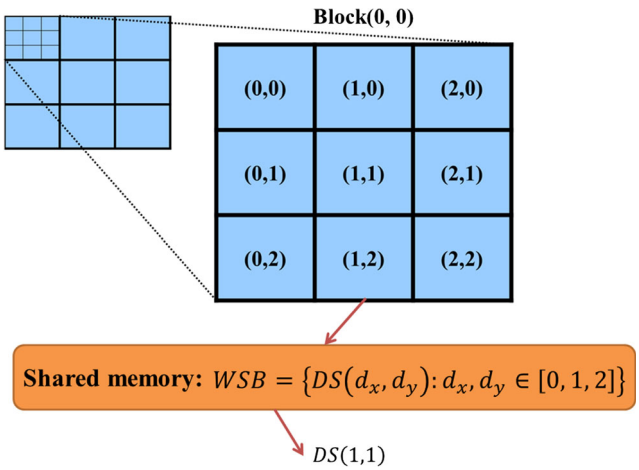

Fig. 2 Illustration of using shared memory for CUDA-based bilateral filtering

Quasi-Optimal Filter Parameter Computation

As shown in Eqs. (9) to (11), there are three parameters, N_k, σ_S, and σ_R, in the bilateral filter. To realize the quasi-optimal values of these parameters on each individual image for the neural network training phase, a brute-force approach is conducted by restoring the image based on the peak signal-to-noise ratio (PSNR):

Table 1 Pseudo code of the proposed GPU-based bilateral filter

int i = blockIdx.x,
j = blockIdx.y,
tx = threadIdx.x,
ty = threadIdx.y,
x = tx + blockDim.x * i,
y = ty + blockDim.y * j;
Create an array in shared memory for $WSB_{i,j}$ in each $block(i,j)$; **if** tx ≤ N_k and ty ≤ N_k **do**:
Compute the spatial distance DS(tx, ty) using Eq. (14),
Store DS(tx, ty) in shared memory buffer $WSB_{i,j}$ based on Eq. (15);**end;**
All threads wait until completion of saving $WSB_{i,j}$ in the array in $block(i,j)$; **for** (nx, ny) is in $\Psi_{x,y}$ of (x, y) **do**:
Retrieve the spatial component $WSB_{i,j}$(nx − x, ny − y);
Calculate the radiometric component W^R(nx, ny) using Eq. (11);
Compute the product W'(nx, ny) of the two components;
Sum up W'(nx, ny) and stored in $sum1$;
Sum up W'(nx, ny) ∗ I(nx, ny) and stored in $sum2$; **end**;
Bilateral filtering at location (x, y) is $sum2$/$sum1$ based on Eq. (16);

$$\mathrm{PSNR} = 10\cdot\log\left[\frac{MNI_{\max}{}^2}{\sum_{i=1}^{M}\sum_{j=1}^{N}\left|\tilde{I}(i,j)-I(i,j)\right|^2}\right] \tag{17}$$

where $\tilde{I}(i,j)$ is the filtered image and $I_{\max}$ is the maximum possible intensity. The higher the PSNR values, the better the restoration results. A large number of combinations of the different parameter values are incorporated into one single parameter set to restore the same image, from which the combination with the highest PSNR score is considered as the quasi-optimal parameter set for that image. The values of N_k are arranged from 1 to 3, σ_S 1 to 6, and σ_R 1 to 100 with an incremental interval equal to 1 for all computation. The quasi-optimal parameter sets for all tested images are then retrieved in the learning stage to train the neural network system and for the evaluation of the proposed bilateral filter.

Back-Propagation Network

Among the diverse ANN systems, the back propagation network (BPN) [50–52] with multilayer feedforward and error back propagation is utilized in our automatic decision framework. A three-layer BPN model is exploited that consists of input layer, hidden layer, and output layer. The seven texture features computed earlier constitutes the input layer. There are 30 neurons in the hidden layer. The transfer function between the input layer and the hidden layer is the hyperbolic tangent function, while the linear transfer function is employed between the hidden layer and the output layer. The output nodes correspond to the three filter parameters and the desired output values of these three parameters are obtained using the brute-force approach as described previously. The famous Levenberg-Marquardt (LM) learning algorithm [53, 54] is adopted to train this parameter prediction system [55]. The training of the proposed BPN model is accomplished when the ensemble error of the three output values is less than a prescribed tolerance.

Overall Procedure

As shown in Fig. 3, the proposed automatic parameter decision framework associated with the BPN model based on image texture features has two major phases: *training* and *testing*. The purpose of the training phase is to establish the architecture of the proposed BPN model while the function of the testing phase is to estimate the filter parameters, which are subsequently adopted by GPU-based bilateral filtering for automatic denoising. In both phases, the same image texture features are computed for the BPN training and predictive models.

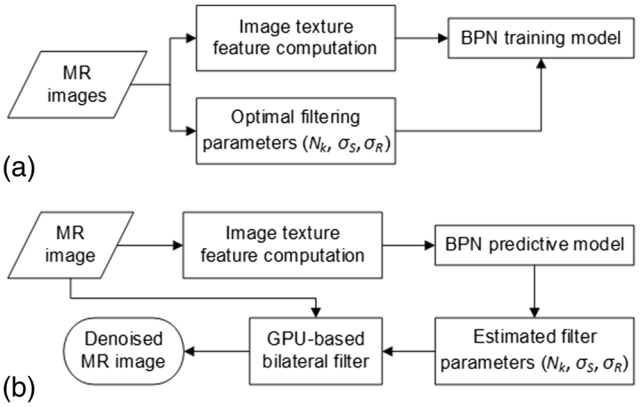

Fig. 3 Flowchart of the computer-aided filter parameter decision system based on the BPN model associated with image texture feature analysis for automatic denoising. **a** Training scheme. **b** Testing scheme

Performance Evaluation

For the evaluation of prediction accuracy of the GPU-based bilateral filter parameters, the mean absolute deviation (MAD), the mean squared error (MSE), and the mean absolute percentage error (MAPE) metrics are employed with

$$\mathrm{MAD} = \frac{1}{K}\sum_{k=1}^{K}\left|v(k)-v'(k)\right| \tag{18}$$

$$\mathrm{MSE} = \frac{1}{K}\sum_{k=1}^{K}\left[v(k)-v'(k)\right]^2 \tag{19}$$

$$\mathrm{MAPE} = \frac{1}{K}\sum_{k=1}^{K}\left|\frac{v(k)-v'(k)}{v(k)}\right|\times 100\% \tag{20}$$

where $v(k)$ represents the expected filter parameter value, $v'(k)$ represents the predicted output value, and K is the number of data being evaluated. The smaller the three scores, the smaller the error, and the higher the prediction accuracy. In general, the grade of MAPE is divided into four predictive ability levels: *high accuracy* when MAPE < 10%, *good* when 10% ≤ MAPE < 20%, *reasonable* when 20% ≤ MAPE < 50%, and *incorrect* when MAPE ≥ 50% [56].

An estimation ratio r is adopted to measure the variation level of the predicted values using

$$r = \frac{v'(k)}{v(k)} \tag{21}$$

The closer the ratio to unity is, the higher the estimation accuracy. To further understand the performance of the proposed BPN framework associated with automatic parameter decision, the k-fold cross validation method [57] is exploited. The benefit of this approach is that all samples are processed for both testing and training, and the k results are averaged to provide a single evaluation. Herein, tenfold cross validation is adopted.

Table 2 Speed-up performance analyses

Dimension	Pixel number	Time CPU(s)	Time GPU(s)	Speed-up
3360 × 2100	7,056,000	24.559846	0.121328	202
2880 × 1800	5,184,000	20.794152	0.137199	152
2560 × 1600	4,096,000	13.664997	0.128550	106
2304 × 1440	3,317,760	13.143704	0.125897	104
2048 × 1280	2,621,440	9.449557	0.120690	78
1920 × 1200	2,304,000	8.735963	0.120157	73
1680 × 1050	1,764,000	8.141209	0.116730	70
1440 × 900	1,296,000	4.486673	0.112241	40
1280 × 800	1,024,000	3.785856	0.111283	34
1152 × 720	829,440	2.975752	0.108985	27
1024 × 640	655,360	2.288589	0.108521	21
512 × 360	184,320	0.669917	0.104038	6
256 × 160	40,960	0.218749	0.103786	2

Table 3 Comparison of parameter prediction accuracy through the tenfold cross validation

Fold	PCA		T-test	
	MSE	MAPE (%)	MSE	MAPE (%)
1	623.2695	6.5036	432.2291	5.7483
2	575.4106	7.2860	408.4445	6.4652
3	609.2034	7.8746	447.0797	7.0630
4	586.0387	7.2246	441.7850	6.5599
5	486.3807	7.2785	356.7186	6.1740
6	561.2248	8.0697	391.0463	7.1365
7	671.6028	7.5229	400.1218	5.3932
8	776.2598	6.9716	470.4942	5.1927
9	673.8009	7.0192	389.0304	5.5014
10	678.8799	6.7686	471.0705	5.7315
Avg.	624.2071	7.2519	420.8020	6.0966
Std.	76.0538	0.4537	35.8323	0.6552

Results

Experimental Data

Two open-access datasets, the Internet Brain Segmentation Repository (IBSR) [58] and the BrainWeb: Simulated Brain Database (SBD) [59], were adopted to validate the proposed automatic filter parameter decision system associated with the GPU-based bilateral filter for their availability. The SBD data were extensively utilized because corresponding noiseless images are provided that allows for systematic development and quantitative analysis in both training and testing phases. As the IBSR dataset consists of clinical images, there is no noiseless image for quantitative evaluation but they are suitable for qualitative analysis. The SBD contains a set of realistic MRI data, which is made up of ten volumetric datasets that define the spatial distribution for different tissues where voxel intensity is proportional to the fraction of tissue within the voxel. More specifically, the images are generated using an MRI simulator, developed at the McConnell Brain Imaging Centre, McGill University, Canada, which enables users to obtain realistic MR images of the brain.

In the SBD, there are two anatomical models: normal and multiple sclerosis (MS). In each model, there are six noise levels: 0, 1, 3, 5, 7, and 9% and three levels of intensity non-uniformity: 0, 20, 40%, which result in 18 different combinations. For the experiments, 15 combinations derived from five noise levels (except 0%) multiplied by three intensity non-uniformities of normal scans with the same 1 mm slice

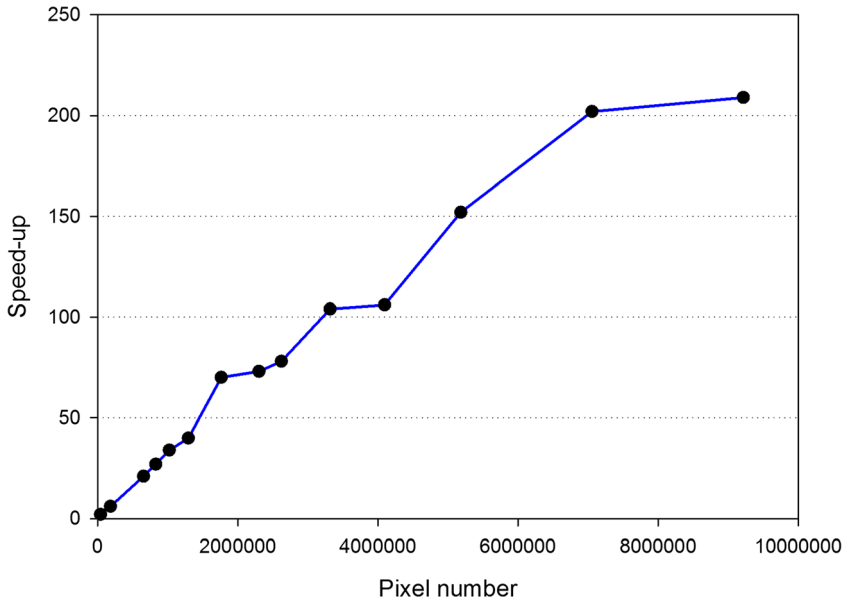

Fig. 4 Speed-up of the GPU-based bilateral filtering compared to the CPU version

Table 4 Comparison of parameter prediction accuracy on the testing datasets

Testing dataset	PCA			*T* test		
	MAPE (%)	MSE	MAD	MAPE (%)	MSE	MAD
1 mm MS	4.9467	372.5095	7.6496	5.9101	341.2912	7.8905
5 mm normal	50.7767	9332.5345	45.7303	29.9516	5036.9879	29.4145
5 mm MS	49.8130	10,077.2403	46.0156	24.1097	5631.1893	32.2132
Avg.	35.1788	6594.0948	33.1318	19.9905	3669.8228	23.1727

thickness were employed as the training dataset, which had 2250 images. All images with 1 mm MS and 5 mm normal and MS scenarios were employed for the testing purpose, which resulted in 3330 images in total. The entire system has been implemented and programmed in MATLAB 2017a (The MathWorks Inc. Natick, MA, USA) associated with C for GPU-based acceleration.

GPU-Based Bilateral Filter

For the efficiency evaluation of the proposed GPU-based bilateral filter, the experiments were performed on an Intel® Xeon(R) CPU E5–2620 v3 2.40GHz equipped with a Tesla K40c GPU card. The Tesla K40c is based on the Kepler architecture, which contains 15 multiprocessors with 192 cores for each multiprocessor that results in 2880 cores altogether. Its memory size is 12 GB with a maximum bandwidth 288 GB/s and 48 KB of shared memory. Each block has 65,536 registers and the maximum number of threads is 1024. Assigning 16×16 threads in a block and a 3×3 filter size with the CUDA driver version 7.5, a wide variety of images with various dimensions were utilized for evaluation. Table 2 presents the performance of the bilateral filter based on the CPU architecture and the accelerated version with the GPU architecture on the same images with different scenarios. It is obvious that as the pixel number increases, the speed-up is becoming apparent, which can also be realized in Fig. 4.

Table 5 Parameter prediction analyses in terms of the ratio r in Eq. (21)

Dataset parameter		Noise level (%)					Avg.
		1	3	5	7	9	
1 mm MS	N_k	1.0000	1.1000	1.0000	1.0000	1.0000	1.0200
	σ_S	0.9931	1.0164	0.9948	0.9854	0.9755	0.9930
	σ_R	1.1197	0.9769	0.9786	0.966	1.0057	1.0094
5 mm normal	N_k	1.0000	1.4000	1.0667	1.0000	1.0000	1.0933
	σ_S	1.2274	1.0514	0.984	0.9944	0.8884	1.0291
	σ_R	1.4567	0.8525	0.843	0.8873	0.9736	1.0026
5 mm MS	N_k	1.0000	1.4000	1.0667	1.0000	1.0000	1.0933
	σ_S	1.2412	1.0347	0.9918	0.9977	0.8877	1.0306
	σ_R	1.4559	0.8151	0.8251	0.8636	0.9588	0.9837

Tenfold Cross Validation

As described previously, the seven most significant features were used as the BPN input arguments. For comparison, the PCA [60] was implemented to convert the texture features into seven principal components, which had the same number of attributes as the proposed t test. The tenfold cross validation method was then applied to understand the effectiveness of the proposed BPN system associated with the corresponding characteristics obtained from both approaches. The training image data were arbitrarily divided into 10 sets, each of which had the same number of 225 images. Each set was utilized as the testing images exactly once and as the training images nine times. Table 3 demonstrates the prediction accuracy analyses for the three parameters in terms of MSE and MAPE. The average MAPE scores for both methods were less than 10%, which were classified as the best predictive ability level of "high accuracy." Nevertheless, the MAPE scores of all folds and the average MAPE score of the proposed t test scheme were smaller than the PCA method.

Filter Parameter Prediction

To further understand the abilities of both methods in predicting the filter parameters, 150 MS brain MR images were randomly selected from the 1-mm thickness dataset with five different noise levels of equal probability. Moreover, 75 images in either anatomical model with the same 5-mm thickness were also arbitrarily chosen for performance evaluation. Table 4 presents the parameter prediction results of the PCA and t test methods. While the average MAPE score of the PCA method exceeded 35%, the average MAPE score of the proposed t test was still below 20%, which was classified as "good" level. The corresponding parameter prediction results in terms of the ratio defined in Eq. (21) using the proposed automation framework were summarized in Table 5. It is evident that most ratios of the three parameters were considerably close to unity, which indicated that the predicted parameters reasonably matched the expected values.

Table 6 Performance comparison of restoration results in PSNR on the 1 mm MS dataset

PSNR	3% Noise					5% Noise				
Slice	VWD	LMMSE	BIL_{NE}	BIL_{AP}	BIL_{BP}	VWD	LMMSE	BIL_{NE}	BIL_{AP}	BIL_{BP}
1	31.90	35.40	34.51	36.71	36.71	28.15	31.37	31.17	32.96	32.96
21	32.19	36.89	36.56	37.10	37.12	29.30	33.19	32.06	32.98	32.98
32	32.23	34.62	33.52	36.08	36.08	27.67	30.63	29.34	32.94	32.94
55	32.66	36.10	35.87	36. 69	36.70	28.95	32.23	32.49	33.37	33.38
66	32.61	35.19	35.40	36.43	36.43	29.54	32.51	32.51	33.32	33.33
96	32.39	34.08	33.33	36.78	36.78	29.37	32.06	32.22	33.42	33.43
105	33.83	36.51	36.12	37.13	37.13	29.10	32.42	32.07	33.93	33.93
118	32.43	35.05	35.32	37.02	37.03	29.10	31.91	31.96	33.52	33.52
126	33.52	36.80	36.32	37.29	37.30	28.93	32.99	32.31	33.61	33.61
140	32.72	36.33	35.90	37.35	37.36	27.14	32.48	31.79	33.81	33.82

Automatic Bilateral Filtering with GPU

Various combinations of noise levels, slice thicknesses, intensity non-uniformities, and anatomical models were adopted to assess the restoration abilities of the proposed bilateral filter with automatic parameter decision and GPU-based acceleration, which was denoted as BIL_{AP}. The bilateral filter with noise variance estimation strategies (denoted as BIL_{NE}) [21, 24, 25, 27], LMMSE [5], and VWD [10] methods were also performed for comparison. Twenty randomly selected restoration results based on the PSNR scores in the 1 mm MS dataset with 3 and 5% noise were demonstrated in Table 6. All computed values of BIL_{AP} were approximately equal to the corresponding quasi-optimal values of BIL_{BP} with negligible errors, where BIL_{BP} represents the bilateral filter with the brute-force manner. The BIL_{AP} outperformed the other three methods with higher PSNR scores in all scenarios. Table 7 presents the quantitative restoration results on another 5-mm normal dataset with 7 and 9% noise levels. Not only did BIL_{AP} closely match BIL_{BP}, but it also produced the highest PSNR score in each scenario comparing to VWD, LMMSE, and BIL_{NE}.

Restoration results with preserved sharp anatomical edges in 3D visualization were demonstrated in Figs. 5 and 6 for the 5-mm normal image volume with 5% noise and the 1-mm MS image volume with 7% noise, respectively. Magnified views of the selected areas are shown on the right-hand side of the corresponding images. Obviously, apparent cortical structures with appropriate noise reduction were acquired after applying the proposed framework. Finally, Fig. 7 illustrates the visual restoration results on clinical adult MR images with Alzheimer's disease. Zoom in views of the selected areas are displayed below the corresponding images. It is evident that the grainy noise pattern in originally scanned MR images was effectively removed while maintaining clearer anatomical structures in all illustrations.

Discussion

To overcome the laboriously manual tuning of the bilateral filter parameters, an automatic decision system based on the BPN associated with image texture features for brain MR images was uniquely developed and introduced. Five of the seven features were the wavelet coefficients with two e1 energies, one e2 energy, and two e3 deviations. The two e1 energies provided the approximate intensity magnitudes of images in two different scales. The e2 energy further supplied discrimination with horizontal edge strength information. The e3 deviations additionally produced comparative edge deviations in different scales and directions. The remaining two GLCM features were contrast and dissimilarity both in the

Table 7 Performance comparison of restoration results in PSNR on the 5-mm normal dataset

PSNR	7% Noise					9% Noise				
Slice	VWD	LMMSE	BIL_{NE}	BIL_{AP}	BIL_{BP}	VWD	LMMSE	BIL_{NE}	BIL_{AP}	BIL_{BP}
2	26.32	29.21	29.06	31.31	31.31	25.23	29.02	27.40	29.29	29.29
7	26.05	30.04	29.45	31.10	31.12	24.10	28.36	27.82	29.44	29.44
12	26.32	30.66	30.62	31.33	31.35	24.11	28.76	29.01	30.17	30.18
18	26.51	29.74	29.35	31.04	31.05	25.22	28.28	27.94	29.01	29.01
22	24.69	27.02	26.93	31.46	31.47	23.96	27.24	27.76	30.36	30.36

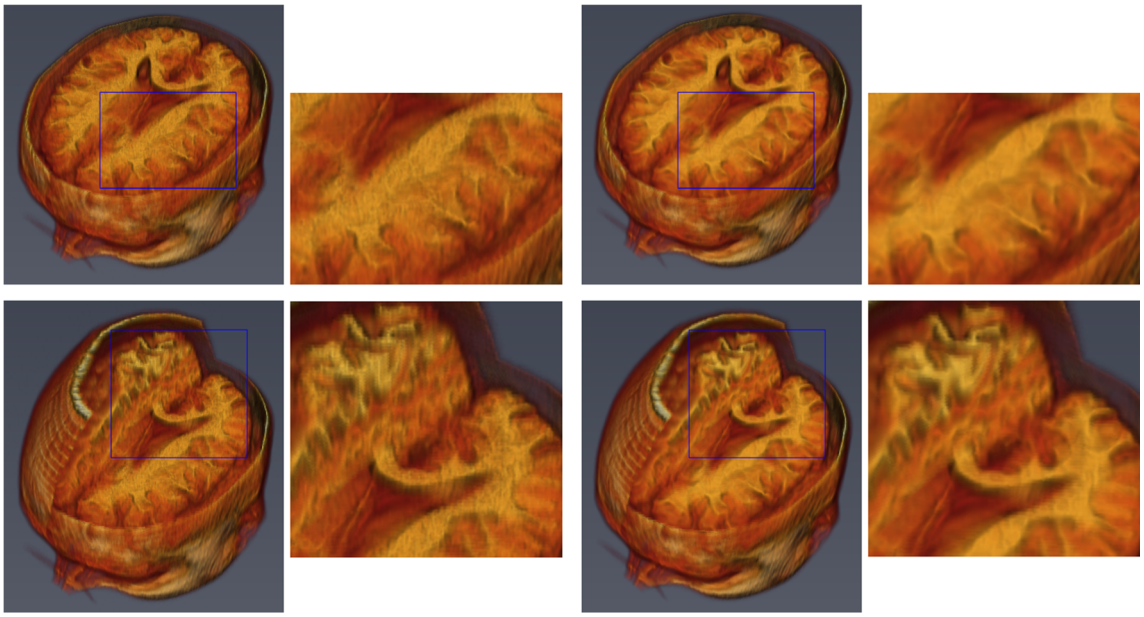

Fig. 5 Automatically GPU-based bilateral filtering the 5-mm normal image volume with 5% noise in 3D visualization. Left column: noisy image volume. Right column: restored image volume

anterior/posterior direction. While the contrast offered variance between adjacent gray tones, the dissimilarity presented divergence between contiguous gray tones. Based on these seven significant texture features, we investigated the acceleration and automation of the bilateral filter, which was the major argument of the current study.

To accelerate the computation of the bilateral filter, a GPU-based parallel computing architecture with CUDA was proposed and established. We addressed optimization techniques to accelerate computation in both thread usages and memory allocations. By exploiting shared memory, the computation speed was further expedited. As presented in Table 2 under the same kernel length, the computation time of the CPU was linearly proportional to the pixel number, while the computation time of the GPU was roughly identical between 0.1 and 0.15 s as the number of pixels grew. The speed-up rose dramatically to hundreds when the pixel number was increasing over millions. For the pixel number of 9,175,040 that was

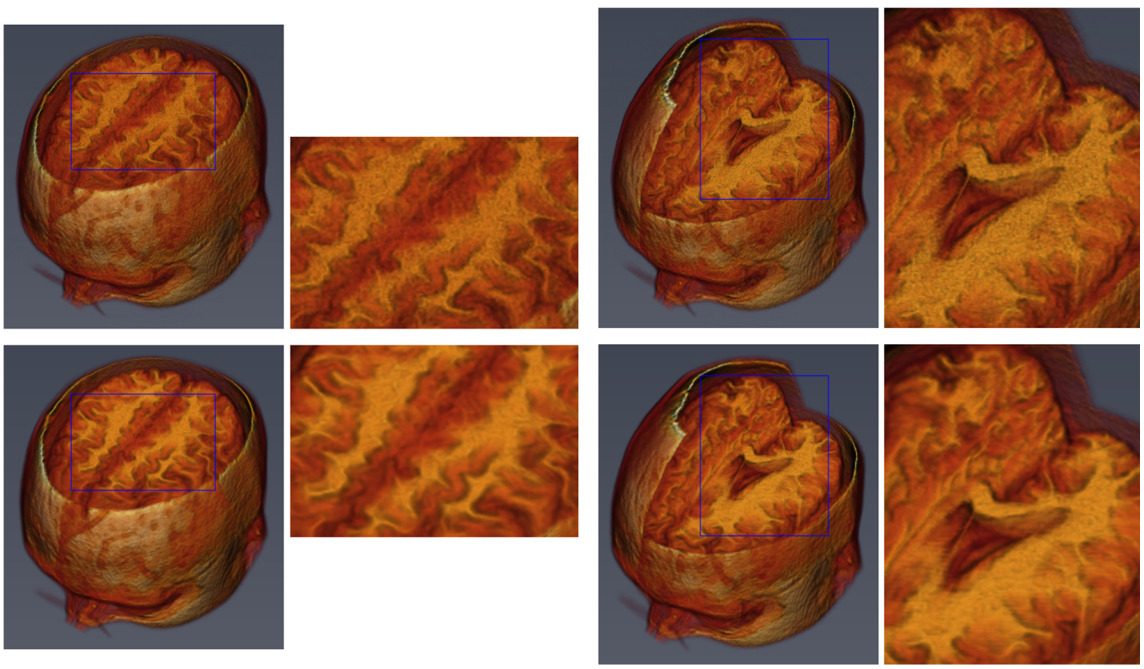

Fig. 6 Automatically GPU-based bilateral filtering the 1 mm MS image volume with 7% noise in 3D visualization. Top row: noisy image volume. Bottom row: restored image volume

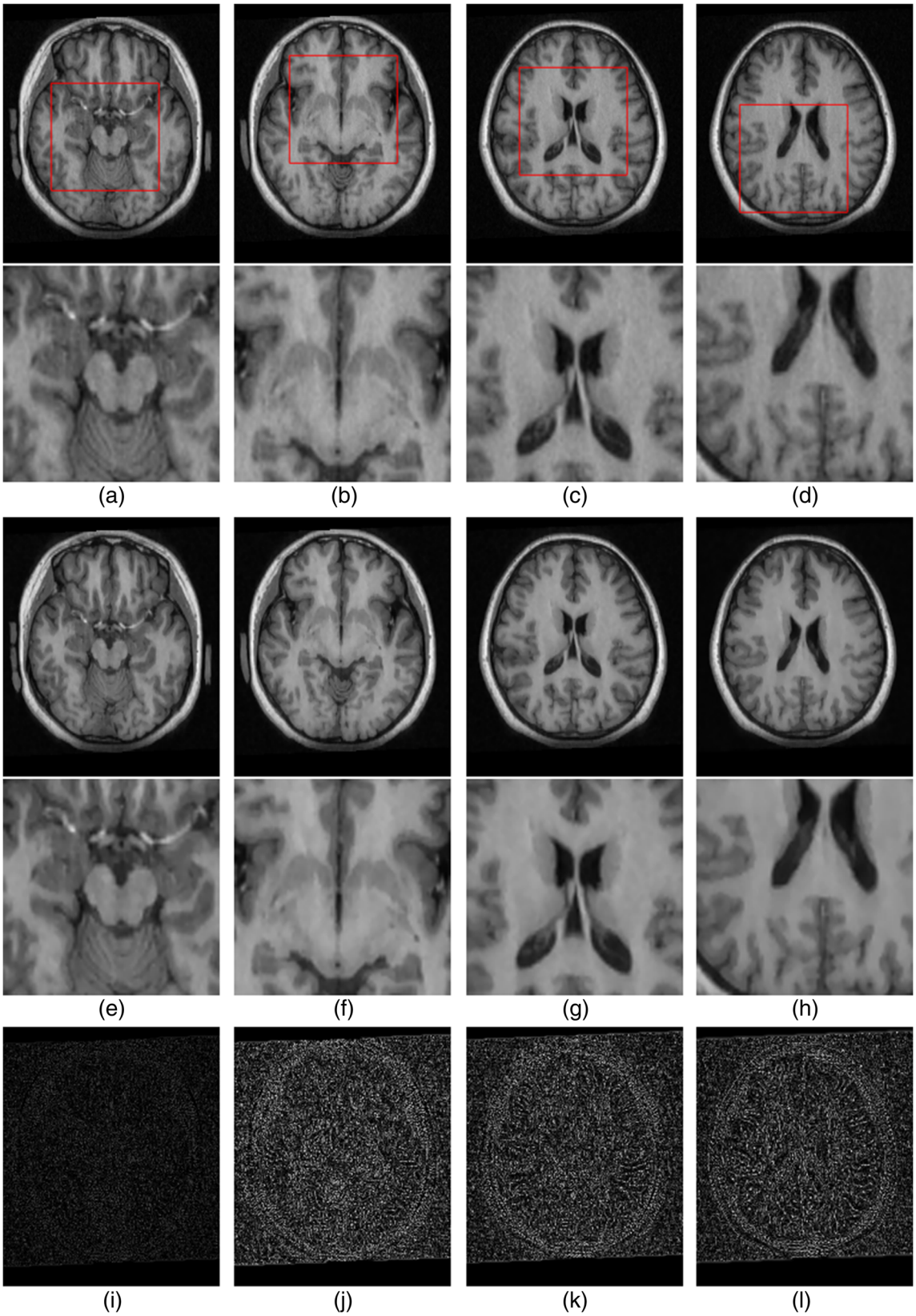

Fig. 7 Automatically bilateral filtering clinical adult MR images with Alzheimer's disease. **a** Scanned slice 55. **b** Scanned slice 63. **c** Scanned slice 85. **d** Scanned slice 88. **e** Denoised image of **a**. **f** Denoised image of **b**. **g** Denoised image of **c**. **h** Denoised image of **d**. **i** Difference image between **a** and **e**. **j** Difference image between **b** and **f**. **k** Difference image between **c** and **g**. **l** Difference image between **d** and **h**

equivalent to a common 256 × 256 brain MR image volume of 140 slices, the speed-up was approximately 208.

As described previously, the reason for utilizing the bilateral filter by numerous researchers and physicians was its efficient performance regardless of inherited Gaussian noise assumption. Nevertheless, the optimal performance of this filter has been an open question since its invention. One popular manner (BIL_{NE}) to semi-automate the filtering process was to estimate the noise variance level, which pertained to some specific parameter in the filter, say σ_R [7, 29]. By automatic setting for each of the three parameters, this paper demonstrated its excellent superiority over the BIL_{NE}, VWD, and LMMSE approaches as presented in Tables 6 and 7. To understand the best filter parameters in the training phase, a brute-force approach was employed with the kernel size being 3 × 3, 5 × 5, and 7 × 7, σ_S being 1 to 6, and σ_R being 1 to 100. The obtained filter parameter values were the best among these 1800 combinations for each image. In our experience, these quasi-optimal filter parameters based on the described setting were adequate for the automation task as tiny adjustments resulting in negligible difference based on the PSNR.

To address the fully automatic parameter setting issue, we exclusively proposed the usage of image texture features associated with the BPN. While the theoretical foundation between the texture features and filter parameters has been lacking, we have established their connection through the BPN. The outcome is a brilliant parameter-free filter, whose restoration results have been demonstrated in both simulated and clinical scenarios. Not only has the system eliminated the burden of tuning the parameter values, but the denoised images have also provided clear vision of anatomical structures for inspection. This can considerably facilitate subsequent image processing procedures for further analysis and diagnosis. For example, to more accurately delineate the cortical structures for neurology study and to more correctly identify the brain tumor lesions for treatment planning.

Conclusions

In conclusion, a new GPU-based bilateral filter based on the BPN model associated with image texture features has been developed to automatically restore brain MR images. The CUDA architecture was exploited to accelerate the bilateral filter computation with optimal thread arrangements and memory allocations. A speed-up factor of 208 was achieved for processing an entire brain MR image volume. Experimental results indicated that our parameter prediction model achieved a MAPE score of 6%, which was classified as "high accuracy." This fully automatic bilateral filter effectively removed noise in a wide variety of brain MR images and outperformed the state-of-the-art LMMSE, VWD, and BIL_{NE} methods. The usage of image texture features associated with the BPN to estimate the GPU-based bilateral filter parameters and to automate the denoising process was feasible and validated. We believe that this automatic filter is advantageous to image interpretation and quantitative assessment where clean images are required. Further research in the acceleration and automation of direct 3D filtering is worth investigating.

Funding This work was supported in part by the Ministry of Science and Technology of Taiwan under Grant No. MOST 104-2221-E-002-095 and Grant No. MOST 106-2221-E-002-082.

References

1. Younis A, Ibrahim M, Kabuka M, John N: An artificial immune-activated neural network applied to brain 3D MRI segmentation. J Digit Imaging 21:69–88, 2008
2. Jang U, Nam Y, Kim D-H, Hwang D: Improvement of the SNR and resolution of susceptibility-weighted venography by model-based multi-echo denoising. Neuroimage 70:308–316, 2013
3. Macovski A: Noise in MRI. Magn Reson Med 36:494–497, 1996
4. Aja-Fernández S, Tristán-Vega A, Alberola-López C: Noise estimation in single- and multiple-coil magnetic resonance data based on statistical models. Magnetic Resonance Imaging 27:1397–1409, 2009
5. Aja-Fernandez S, Alberola-Lopez C, Westin CF: Noise and signal estimation in magnitude MRI and Rician distributed images: a LMMSE approach. Image processing. IEEE Transactions on 17: 1383–1398, 2008
6. Manjón JV, Carbonell-Caballero J, Lull JJ, García-Martí G, Martí-Bonmatí L, Robles M: MRI denoising using non-local means. Med Image Anal 12:514–523, 2008
7. He L, Greenshields IR: A nonlocal maximum likelihood estimation method for Rician noise reduction in MR images. IEEE Trans Med Imag 28:165–172, 2009
8. Liu H, Yang C, Pan N, Song E, Green R: Denoising 3D MR images by the enhanced non-local means filter for Rician noise. Magn Reson Imaging 28:1485–1496, 2010
9. Manjón JV, Coupé P, Martí-Bonmatí L, Collins DL, Robles M: Adaptive non-local means denoising of MR images with spatially varying noise levels. J Magn Reson Imaging 31:192–203, 2010
10. Pizurica A, Philips W, Lemahieu I, Acheroy M: A versatile wavelet domain noise filtration technique for medical imaging. IEEE Trans Med Imaging 22:323–331, 2003
11. Kim J, Leira EC, Callison RC, Ludwig B, Moritani T, Magnotta VA, Madsen MT: Toward fully automated processing of dynamic susceptibility contrast perfusion MRI for acute ischemic cerebral stroke. Comput Methods Programs Biomed 98:204–213, 2010
12. Malinsky M, Peter R, Hodneland E, Lundervold AJ, Lundervold A, Jan J: Registration of FA and T1-weighted MRI data of healthy human brain based on template matching and normalized cross-correlation. J Digit Imaging 26:774–785, 2013
13. Perona P, Malik J: Scale-space and edge detection using anisotropic diffusion. IEEE Trans Pattern Anal Machine Intell 12:629–639, 1990
14. Ferrari R: Off-line determination of the optimal number of iterations of the robust anisotropic diffusion filter applied to denoising of brain MR images. Med Biol Eng Comput 51:71–88, 2013
15. Tomasi C, Manduchi R: Bilateral filtering for gray and color images. Proc. Computer Vision, 1998 Sixth International Conference on: City
16. Anand CS, Sahambi JS: MRI denoising using bilateral filter in redundant wavelet domain. IEEE Proc TENCON:1–6, 2008

17. Chang H-H, Chu W-C: Restoration algorithm for image noise removal using double bilateral filtering. Journal of Electronic Imaging 21:023028–023021, 2012
18. Dong G, Acton ST: On the convergence of bilateral filter for edge-preserving image smoothing. Signal processing letters. IEEE 14: 617–620, 2007
19. Zhang B, Allebach JP: Adaptive bilateral filter for sharpness enhancement and noise removal. Image processing. IEEE Transactions on 17:664–678, 2008
20. Farzana E, Tanzid M, Mohsin KM, Bhuiyan MIH: Bilateral filtering with adaptation to phase coherence and noise. SIViP 7:367–376, 2013
21. Walker SA, Miller D, Tanabe J: Bilateral spatial filtering: refining methods for localizing brain activation in the presence of parenchymal abnormalities. Neuroimage 33:564–569, 2006
22. Rydell J, Knutsson H, Borga M: Bilateral filtering of fMRI data. Selected topics in signal processing. IEEE Journal of 2:891–896, 2008
23. Hamarneh G, Hradsky J: Bilateral filtering of diffusion tensor magnetic resonance images. Image processing. IEEE Transactions on 16:2463–2475, 2007
24. McPhee KC, Denk C, Al-Rekabi Z, Rauscher A: Bilateral filtering of magnetic resonance phase images. Magn Reson Imaging 29: 1023–1029, 2011
25. Jaramillo R, Lentini M, Paluszny M: Improving the performance of the Prony method using a wavelet domain filter for MRI Denoising. Comput Math Methods Med 2014:10, 2014
26. Wells JR, Dobbins JT: Frequency response and distortion properties of nonlinear image processing algorithms and the importance of imaging context. Med Phys 40:091906, 2013
27. Kala R, Deepa P: Removal of rician noise in MRI images using bilateral filter by fuzzy trapezoidal membership function. Proc. 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS): City, 6–7 Jan. 2017 Year
28. Szczypiński PM, Strzelecki M, Materka A, Klepaczko A: MaZda—a software package for image texture analysis. Comput Methods Programs Biomed 94:66–76, 2009
29. López-Rubio E, Florentín-Núñez MN: Kernel regression based feature extraction for 3D MR image denoising. Med Image Anal 15: 498–513, 2011
30. Yang X, Fei B: Multiscale segmentation of the skull in MR images for MRI-based attenuation correction of combined MR/PET. J Am Med Inform Assoc 20:1037–1045, 2013
31. Haykin SS: Neural Networks And Learning Machines: Pearson Education Upper Saddle River, 2009
32. Virmani J, Kumar V, Kalra N, Khandelwal N: Neural network ensemble based CAD system for focal liver lesions from B-mode ultrasound. J Digit Imaging 27:520–537, 2014
33. Wang J, Kato F, Yamashita H, Baba M, Cui Y, Li R, Oyama-Manabe N, Shirato H: Automatic estimation of volumetric breast density using artificial neural network-based calibration of full-field digital mammography: feasibility on Japanese women with and without breast cancer. J Digit Imaging 30:215–227, 2017
34. Lin Y-J, Chang H-H: Investigation of significant attributes based on image feature and texture analysis for automatic noise reduction in MRI. The 15th international conference on biomedical engineering (ICBME 2013). IFMBE Proc 43:589–592, 2013
35. General Purpose GPU Programming (GPGPU), http://www.gpgpu.org, 2016
36. Park IK, Singhal N, Lee MH, Cho S, Kim C: Design and performance evaluation of image processing algorithms on GPUs. IEEE Transactions on Parallel and Distributed Systems 22:91–104, 2011
37. Gonzalez RC, Woods RE: Digital Image Processing: Pearson, 2010
38. Mallat SG: A theory for multiresolution signal decomposition: the wavelet representation. Pattern analysis and machine intelligence. IEEE Transactions on 11:674–693, 1989
39. Chang T, Kuo C-C: Texture analysis and classification with tree-structured wavelet transform. Image processing. IEEE Transactions on 2:429–441, 1993
40. Haar A: Zur theorie der orthogonalen funktionensysteme. Mathematische Annalen 69:331–371, 1910
41. Haralick RM, Shanmugam K, Dinstein IH: Textural features for image classification. Systems, man and cybernetics, IEEE Transactions on SMC. 3:610–621, 1973
42. Antel SB, Collins DL, Bernasconi N, Andermann F, Shinghal R, Kearney RE, Arnold DL, Bernasconi A: Automated detection of focal cortical dysplasia lesions using computational models of their MRI characteristics and texture analysis. NeuroImage 19:1748–1759, 2003
43. Howarth P, Ruger S: Robust texture features for still-image retrieval. Vision, image and signal processing. IEE Proceedings 152:868–874, 2005
44. Chen W, Giger ML, Li H, Bick U, Newstead GM: Volumetric texture analysis of breast lesions on contrast-enhanced magnetic resonance images. Magnetic Resonance in Medicine 58:562–571, 2007
45. Wei S, Yanling H, Zhizhong L, Peng W: The research of satellite cloud image recognition base on variational method and texture feature analysis. Proc. Industrial Electronics and Applications, 2007 ICIEA 2007 2nd IEEE Conference on: City, 23–25 May 2007 Year
46. Fisher RA, Genetiker S, Genetician S, Britain G, Généticien S: Statistical Methods for Research Workers: Oliver and Boyd Edinburgh, 1970
47. Breiman L, Friedman JH, Olshen RA, Stone CJ: Classification and Regression Trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software, 1984
48. NVIDIA CUDA ZONE, http://www.nvidia.com/cuda, 2016
49. Nickolls J, Buck I, Garland M, Skadron K: Scalable parallel programming with CUDA. Queue 6:40–53, 2008
50. Hecht-Nielsen R: Theory of the backpropagation neural network. Proc. Neural Networks, 1989 IJCNN, International Joint Conference on: City, 0–0 1989 Year
51. Rumelhart DE, Hinton GE, Williams RJ: Learning representations by back-propagating errors. Nature 323:533–536, 1986
52. McClelland JL, Rumelhart DE, Group PR: Parallel distributed processing. Explorations in the microstructure of cognition 2, 1986
53. Levenberg K: A method for the solution of certain problems in least squares. Quarterly of applied mathematics 2:164–168, 1944
54. Marquardt DW: An algorithm for least-squares estimation of nonlinear parameters. Journal of the Society for Industrial & Applied Mathematics 11:431–441, 1963
55. Suzuki K, Abe H, MacMahon H, Doi K: Image-processing technique for suppressing ribs in chest radiographs by means of massive training artificial neural network (MTANN). Medical imaging. IEEE Transactions on 25:406–416, 2006
56. Lewis CD: Industrial and business forecasting methods: a practical guide to exponential smoothing and curve fitting: Butterworth Scientific London, 1982
57. Stone M: Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society Series B (Methodological):111–147, 1974
58. MGH: Internet Brain Segmentation Repository (IBSR), http://www.cma.mgh.harvard.edu/ibsr/, 2015
59. McGill University: BrainWeb: Simulated Brain Database, http://www.bic.mni.mcgill.ca/brainweb/, 2015
60. Abdi H, Williams LJ: Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics 2:433–459, 2010