

RESEARCH ARTICLE

Open Access



An automatic restoration framework based on GPU-accelerated collateral filtering in brain MR images

Heng-Hua Chang*  and Cheng-Yuan Li

Abstract

Background: Image restoration is one of the fundamental and essential tasks within image processing. In medical imaging, developing an effective algorithm that can automatically remove random noise in brain magnetic resonance (MR) images is challenging. The collateral filter has been shown a more powerful algorithm than many existing methods. However, the computation of the collateral filter is more time-consuming and the selection of the filter parameters is also laborious. This paper proposes an automatic noise removal system based on the accelerated collateral filter for brain MR images.

Methods: To solve these problems, we first accelerated the collateral filter with parallel computing using the graphics processing unit (GPU) architecture. We adopted the compute unified device architecture (CUDA), an application programming interface for the GPU by NVIDIA, to hasten the computation. Subsequently, the optimal filter parameters were selected and the automation was achieved by artificial neural networks. Specifically, an artificial neural network system associated with image feature analysis was adopted to establish the automatic image restoration framework. The best feature combination was selected by the paired t-test and the sequential forward floating selection (SFFS) methods.

Results: Experimental results indicated that not only did the proposed automatic image restoration algorithm perform dramatically faster than the traditional collateral filter, but it also effectively removed the noise in a wide variety of brain MR images. A speed up gain of 34 was attained to process an MR image, which completed within 0.1 s. Representative illustrations of brain tumor images demonstrated the capability of identifying lesion boundaries, which outperformed many existing methods.

Conclusions: We believe that our accelerated and automated restoration framework is promising for achieving robust filtering in many brain MR image restoration applications.

Keywords: Collateral filter, GPU, Parallel computing, Image feature, Neural networks, MRI

Background

Since its invention in the 1970s, magnetic resonance imaging (MRI) has been an important imaging technique for noninvasive diagnosis of the human body. The applications of MRI in the brain result in diversified images for further processes such as tissue classification, segmentation and registration [1–3]. However, the random noise in MRI scanners inevitably causes deterioration of

brain MR images [4]. Consequently, noise removal has become a crucial issue in brain MR image post-processing [5]. The Gaussian filter has been extensively adopted in many image processing applications for its simplicity. However, anatomical boundaries are inevitably blurred by this lowpass filter. In addition to the Gaussian filter, one of the well-known techniques for noise removal has been the bilateral filter, which was originally proposed by Tomasi and Manduci [6]. It has been widely utilized in many medical image denoising applications for decades. The bilateral filter is an effective filtering algorithm that can both remove the random noise and preserve edges rather than

* Correspondence: herbertchang@ntu.edu.tw
Computational Biomedical Engineering Laboratory (CBEL), Department of Engineering Science and Ocean Engineering, National Taiwan University, No. 1 Sec. 4 Roosevelt Road, Daan, 10617 Taipei, Taiwan



blurring the lines in images. An iterative version of the bilateral filter for MR image restoration was recently introduced [7].

One remarkable technique of using statistics strategies is the linear minimum mean squared error (LMMSE) estimator [8], which computes the local mean, the local variance, and the local mean square value of the input image. Another famous method is the anisotropic diffusion filter (ADF) [9], in which pixel intensities are averaged from neighbors in a prescribed window, whose dimension and shape are measured at every location. An appropriate function of the image gradient is constructed in accordance with the diffusion coefficient to encourage filtering within a region of interest in preference not to filtering across the boundaries. It is noted that the quality of the denoised image is greatly relevant to the number of iterations. Subsequently, Ferrari [10] proposed a mathematical framework to automatically determine the best number of iterations of the ADF method by utilizing the maximization of some evaluation index. Nevertheless, it has been adopted in many medical image restoration applications [11].

In contrast to many methods that mainly reply on local pixels within a small neighbourhood, the non-local means (NLM) filter detects repeated structures in the global domain [12]. The similarity between pixels is region-based comparison in that pixels far from the kernel center are not penalized due to the distance to the center. Based on an enhanced sparse representation in the transform domain, the block-matching and 3D filtering (BM3D) strategy [13] extended the nonlocal filtering techniques. The enhancement of the sparsity is accomplished by aggregating similar 2D fragments of the image into 3D arrays. Subsequently, the Wiener filter is employed in the collaborative filtering procedure to remove noise.

Motivated by the bilateral filter, the collateral filter [14] was recently proposed that introduced a median filter and an entropy function to the filtering framework. The median filter compensated the Gaussian filter for random noise reduction. It has been shown that the collateral filter is more powerful than the Gaussian filtering, bilateral filtering, and ADF methods in many scenarios [14]. However, comparing to these two filters, the collateral filter is more complicated and time consuming. A contemporary technique called the tensor processing unit (TPU) has shown its excellent computation and energy efficiency over existing frameworks [15]. However, this domain specific architecture is exclusively deployed in Google data centers. One realistic approach to speed up the denoising procedure is through the parallel programming using the compute unified device architecture (CUDA) [16, 17] with the graphics processing unit (GPU).

A GPU is a specialized processor that accelerates graphic operations on personal computers, workstations, and mobile devices. The GPU has more processor cores than a traditional central processing unit (CPU) and each core can process data simultaneously. Therefore, the GPU is well suited to handle tasks that can be parallelized. For example, GPU-based acceleration has been shown effective in segmentation and classification applications [18, 19]. Additionally, CUDA is a programming platform developed by NVIDIA for general purpose computation on GPU (GPGPU) on NVIDIA GPUs [20]. Many of NVIDIA graphics cards support CUDA (e.g., GeForce, Quadro, and Tesla). With CUDA, programmers can manage memory on the CPU as well as the GPU from the host. Once memory spaces on both host and device sides are allocated and data are transferred, the program can launch kernels for execution on the device.

A kernel is a function that can be called from the host and executed on the device. Programmers can design the code in the kernel function for parallel computing. The kernel function defines the operations for a single thread and thousands of threads execute the function code synchronously. A thread is the basic execution unit for parallel computing. After the thread completes its task in the kernel function, the data are transferred back to the host.

Based on the GPU architecture, one pixel in an image can be handled by one thread and thousands of threads can run synchronously. Each thread can access data from different types of memories such as local, global and shared memory. While local memory belongs to one specific thread, global memory is shared by all threads. Indeed, local memory is part of global memory and both is slower than shared memory. Threads are organized in blocks and a grid is a group of blocks. Each block has its shared memory that can be accessed by its own threads. Although shared memory's size is smaller than global memory, the access time of shared memory is faster like cache [21].

The objective of this paper is two-fold. First, to accelerate the computation of the collateral filter through the adoption of the CUDA strategy. Second, to automate the collateral filtering process by the aid of image texture features associated with artificial neural networks [22]. We will show that a fast and automatic denoising system based on CUDA-based collateral filtering provides robust and accurate restoration results over many existing denoising methods.

Methods

CUDA-based collateral filtering

As described earlier, the collateral filter is a more effective denoising method than the bilateral filter in

that it contains three different Gaussian functions [14]. Let (θ_x, θ_y) be the pixel location under consideration in the filtering image and $\Psi_{\theta_x, \theta_y}$ be the neighborhood of (θ_x, θ_y) :

$$\Psi_{\theta_x, \theta_y} = \left\{ (\mu_x, \mu_y) : (\mu_x, \mu_y) \in [\theta_x - N, \theta_x + N] \times [\theta_y - N, \theta_y + N] \right\} \quad (1)$$

where N is the filter radius. The Gaussian functions for the spatial, radiometric and median-filtered components are respectively defined as

$$W_{\theta_x, \theta_y}^S(\mu_x, \mu_y) = \exp \left[-\frac{|(\mu_x, \mu_y) - (\theta_x, \theta_y)|^2}{2\sigma_S^2} \right] \quad (2)$$

and

$$W_{\theta_x, \theta_y}^R(\mu_x, \mu_y) = \exp \left[-\frac{|I(\mu_x, \mu_y) - I(\theta_x, \theta_y)|^2}{2\sigma_R^2} \right] \quad (3)$$

and

$$W_{\theta_x, \theta_y}^M(\mu_x, \mu_y) = \exp \left[-\frac{|I^M(\mu_x, \mu_y) - I^M(\theta_x, \theta_y)|^2}{2\sigma_M^2} \right] \quad (4)$$

where $I(\mu_x, \mu_y)$ is the intensity value at (μ_x, μ_y) and $I^M(\mu_x, \mu_y)$ is the median filtered image at location (μ_x, μ_y) . In addition to median filtering, the collateral filter introduces an entropy function that is utilized to balance the radiometric and median-filtered components using

$$H(\theta_x, \theta_y) = -\sum_{(\mu_x, \mu_y) \in \Psi_{\theta_x, \theta_y}} p(\mu_x, \mu_y) \log p(\mu_x, \mu_y) + [1 - p(\mu_x, \mu_y)] \log [1 - p(\mu_x, \mu_y)] \quad (5)$$

where

$$p(\mu_x, \mu_y) = \frac{\left[1 - \frac{|I(\mu_x, \mu_y) - I^M(\mu_x, \mu_y)|}{MAX_I} \right]^2}{2} + 0.5 \quad (6)$$

where MAX_I is the maximum intensity value of the input image I .

The ensemble weight function that combines all the components is then defined as

$$W_{\theta_x, \theta_y} = W_{\theta_x, \theta_y}^S \left(W_{\theta_x, \theta_y}^R \right)^{\frac{1}{1+H(\theta_x, \theta_y)}} \left(W_{\theta_x, \theta_y}^M \right)^{H(\theta_x, \theta_y)} \quad (7)$$

The collateral filtering of image I at location (θ_x, θ_y) is finally computed using

$$I^C(\theta_x, \theta_y) = \frac{\sum_{(\mu_x, \mu_y) \in \Psi} W_{\theta_x, \theta_y}(\mu_x, \mu_y) [(1-\beta)I + \beta I^M]}{\sum_{(\mu_x, \mu_y) \in \Psi} W_{\theta_x, \theta_y}(\mu_x, \mu_y)} \quad (8)$$

where β is adopted to adjust the weight between the input image I and the median filtered image I^M . A number of techniques have been proposed to accelerate image filter computation. Some researchers used approximation methods to reduce the computation time [23]. A straightforward approach is to take advantage of an additional memory space. Inspired by the strategy in [24], we introduce a new scheme not only saving computation time but also achieving the same restoration results as using the traditional collateral filter.

In the traditional collateral filter, the spatial component is repeatedly computed for every pixel, which takes a lot of time in computing the same pixel distances. Therefore, we redefine (2) as

$$DS(d_x, d_y) = \exp \left[-\frac{d_x^2 + d_y^2}{2\sigma_S^2} \right] \quad (9)$$

where d_x and d_y are the spatial distances with $0 \leq d_x \leq N$ and $0 \leq d_y \leq N$. In our approach, we create a spatial weight buffer WSB , which is defined as

$$WSB = \{ DS(d_x, d_y) : d_x, d_y \in [1, \dots, N] \} \quad (10)$$

We then compute all the spatial weights and store them in this memory buffer WSB according to the filter radius N . Let $WSB(x, y)$ be an element of WSB , (2) can be expressed as $WSB(\mu_x - \theta_x, \mu_y - \theta_y)$.

Subsequently, we implement the complex parts of the collateral filter using CUDA shared memory, which is an efficient way to reduce the reading and writing time comparing to the usage of global memory. We split the image pixels into separate blocks, each denoted as block (i, j) , where $i = 1, 2, \dots, k$ and $j = 1, 2, \dots, k$. For entropy computation, each pixel and its neighbors are required to sum up as shown in (5). To hasten the computation, the entropy of the pixel at (x, y) is computed and stored in the shared memory buffer in advance using

$$HPB_{i,j}(x, y) = p(x, y) \log p(x, y) + [1 - p(x, y)] \log [1 - p(x, y)] \quad (11)$$

where $S \times (i - 1) - N < x \leq S \times i + N$, $S \times (j - 1) - N < y \leq$

$S \times j + N$, and $S \times S$ represents the block size. Once $HPB_{i,j}$ is created, the entropy is defined as:

$$HG_{i,j}(\delta_x, \delta_y) = - \sum_{(\mu_x, \mu_y) \in \Psi_{\delta_x, \delta_y}} HPB_{i,j}(\mu_x, \mu_y) \quad (12)$$

where (δ_x, δ_y) represents a thread in block (i, j) with $1 \leq \delta_x \leq S$ and $1 \leq \delta_y \leq S$. Since accessing shared memory is faster than accessing local memory, each thread only needs to sum up its own value from its neighbors. Consequently, we redefine (7) as:

$$W_{i,j,\delta_x,\delta_y} = WSB_{i,j} \left(W_{i,j,\delta_x,\delta_y}^R \right)^{\frac{1}{1+HG_{i,j}(\delta_x,\delta_y)}} \left(W_{i,j,\delta_x,\delta_y}^M \right)^{HG_{i,j}(\delta_x,\delta_y)} \quad (13)$$

where $WSB_{i,j}$ represents the spatial weight buffer in block (i, j) .

In order to more effectively utilize shared memory, the computation of β in (8) is reformulated so that it is calculated at the beginning in every block (i, j) before applying the filter to each pixel. Finally, the accelerated version of the collateral filter is defined as:

$$I_{i,j}^{AC}(\delta_x, \delta_y) = \frac{\sum_{(\mu_x, \mu_y) \in \Psi_{\delta_x, \delta_y}} W_{i,j,\delta_x,\delta_y}(\mu_x, \mu_y) BETA_{i,j}(\mu_x, \mu_y)}{\sum_{(\mu_x, \mu_y) \in \Psi_{\delta_x, \delta_y}} W_{i,j,\delta_x,\delta_y}(\mu_x, \mu_y)} \quad (14)$$

where $BETA_{i,j}(\mu_x, \mu_y)$ is the value of β at (μ_x, μ_y) . The pseudo code of our GPU-based collateral filter is depicted in Fig. 1. Note that each pixel being processed is assigned to a thread since the process is a slice-by-slice approach, i.e., one 2D image at a time.

Feature extraction

Although the execution time of the traditional collateral filter is hastened using the GPU strategy, it is still laborious to manually adjust the parameters for achieving the best restoration results in a particular brain MR image. Using an artificial neural network system is suitable as it can learn to estimate the optimal parameters in the filtering process. However, neural networks require appropriate input features to obtain the best performance of the predictable model. In our experience, image texture

features play an important role in the neural network automation process. The candidate image texture features adopted in this study are described as follows.

Statistical features

There are several basic statistical features that are computed directly from the image intensity: mean intensity (Mean), standard deviation (SD), variance (VAR), entropy (ENT), and histogram features (skewness, kurtosis, variance, entropy and energy).

Gray-level co-occurrence matrix (GLCM)

The GLCM features were proposed by Haralick et al. [25]. A co-occurrence matrix is defined as a matrix of frequencies at which two pixels, separated by a certain distance and direction, occur in the image. Eight GLCM features [25–27] in four directions ($\theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ$) with distance $d = 1$ are computed:

- (1) Standard deviations in the i -direction (SD_i)
- (2) Standard deviations in the j -direction (SD_j)
- (3) Energy or angular second moment (ASM)
- (4) Contrast (CON)
- (5) Dissimilarity (DIS)
- (6) Homogeneity (HOM)
- (7) Entropy (ENT)
- (8) Correlation (COR)

Gray-level run-length matrix (GLRLM)

The GLRLM features were proposed by Galloway [28]. A run-length matrix is defined as the number of runs with a pixel of a specific gray level and its run length in a certain direction. Eleven GLRLM features in four directions ($\theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ$) are computed:

- (1) Short run emphasis (SRE)
- (2) Long run emphasis (LRE)
- (3) Gray level nonuniformity (GLN)
- (4) Run length nonuniformity (RLN)
- (5) Run percentage (RP)
- (6) Low gray-level run emphasis (LGRE)
- (7) High gray-level run emphasis (HGRE)
- (8) Short run low gray-level emphasis (SRLGE)

```

Collateral_Filter_GPU
1. Create arrays in shared memory for the spatial component WSBi,j, entropy function HGi,j and beta BETAi,j
2. Obtain the current thread(x, y) in block(i, j)
3. Compute the three different values and store them in the arrays using thread(x, y)
4. All threads in block(i, j) wait until completion of handling the arrays
5. Calculate output value for thread(x, y) using the arrays based on (14)
6. Store the result at (x, y) in the output image
End
    
```

Fig. 1 Pseudo code of the GPU-accelerated collateral filter

- (9) Short run high gray-level emphasis (SRHGE)
- (10) Long run low gray-level emphasis (LRLGE)
- (11) Long run high gray-level emphasis (LRHGE)

Tamura texture features

Tamura features [29] were developed based on the human visual and psychological perception. The first three features, which are correlated more closely with human perception, are computed:

- (1) Coarseness (CRS)
- (2) Contrast (CON)
- (3) Directionality (DIR)

Their formulas are briefly provided in Additional file 1.

Noise estimation features

Since the collateral filter parameters are sensitive to noise, noise related features are required. Many noise variance estimation methods that are based on the Rician and Rayleigh distributions are complicated. An easier way to estimate the noise level is to employ a simpler Laplacian operator [30]. A more efficient algorithm [31] with a generic transfer function improved the Laplacian mask for better estimation. Aja-Fernández et al. [32] presented a set of estimators based on the Rayleigh distribution in the background. Moreover, the peak value of the autocorrelation function computed by an image with itself provides the information of noise [33]. Based on the above methods, the following seven features are adopted:

- (1) Laplacian mask (LAP)
- (2) Laplacian of Gaussian (LOG)
- (3) A generic transfer function with Laplacian (GTFLAP)
- (4) Maximum value of some local distribution (AJA)
- (5) A least square fitting of the Rayleigh distribution with the histogram background (BRUM)
- (6) Maximum likelihood function (SIJ)
- (7) Autocorrelation function (ACF)

Feature selection

If a large amount of features are selected as the input arguments, the training of neural networks will be an extremely time-consuming procedure. Consequently, we utilize the sequential forward floating selection (SFFS) method [34] to choose the best combination of features. Before applying the SFFS, we make use of the paired-samples t-test [35, 36] to rank the significance of candidate features. The t-test is applied to each image feature to evaluate the ability for discriminating differences between noise levels, which results in an average p -value. The smaller the p -value, the better the

discrimination. Statistically, those features with an average $p < 0.05$ are selected as candidates for the SFFS process.

Neural networks

The popular back propagation neural network (BPN), proposed by McClelland et al. [37], is utilized for training our neural network system. The main architecture includes input, hidden and output layers. The back propagation process is employed to optimize the weights by minimizing the target error so that the neural network is able to learn the input-output mapping perfectly. In our approach, the inputs are the features selected by the SFFS algorithm and the outputs are the collateral filter parameters. In addition, a brute-force approach is conducted to obtain the optimal filter parameters based on the peak signal-to-noise ratio (PSNR):

$$PSNR = 10 \times \log_{10} \left[\frac{MAX_I^2}{\frac{1}{mn} \times \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [R(i, j) - F(i, j)]^2} \right] \tag{15}$$

where F is the $m \times n$ noise-free image, R is the restored image, and MAX_I is the maximum possible intensity of F .

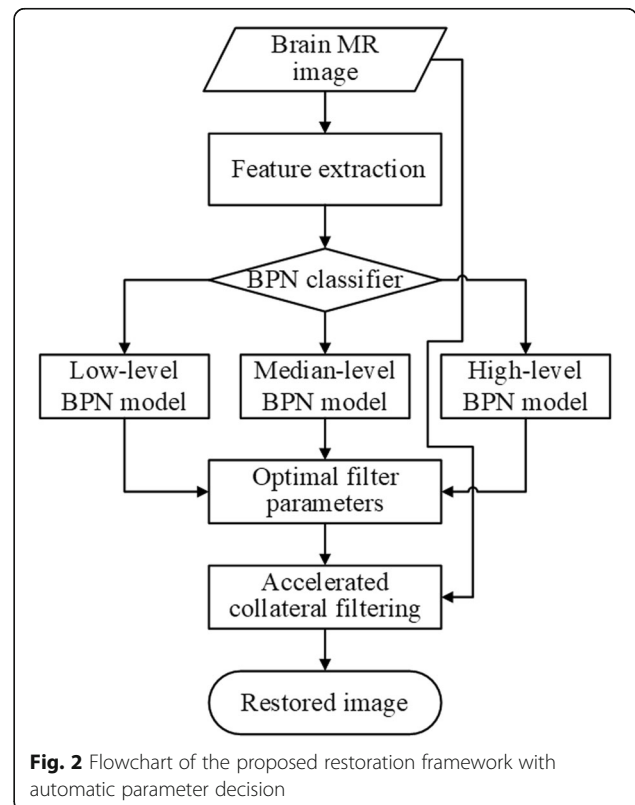


Fig. 2 Flowchart of the proposed restoration framework with automatic parameter decision

Table 1 Execution time of the CPU-based and GPU-based collateral filters

Image number	CPU (s)	GPU without SM (s)	GPU with SM (s)	Speed up
1	0.63	0.14	0.018	34
10	6.06	0.16	0.027	224
100	58.78	0.24	0.108	541
500	260.52	0.59	0.425	612
1000	512.38	1.05	0.871	588

The optimal filter parameters are decided when the highest *PSNR* value is obtained.

Finally, a two stage neural network structure is developed with the first stage for classification and the second stage for prediction. At the first stage, the neural network classifier distinguishes inputs into three different noise levels. Multiple-layer (input, hidden, and output) neural networks with different noise level models at the second stage predict the optimal filter parameters for image restoration. The Levenberg-Marquardt learning algorithm [38] is exploited to independently train each of the three individual BPN models. For each BPN model, the transfer function between the input layer and the hidden layer is the hyperbolic tangent function, whereas the linear transfer function is employed between the hidden layer and the output layer. The overall flow-chart of the proposed methods is depicted in Fig. 2.

Performance evaluation

In addition to the *PSNR* metric in (15), the structural similarity (SSIM) metric [39] is utilized to evaluate the restored images using

Table 2 Paired-samples t-test results based on the average *p*-value with $p < 0.05$

No.	Feature	<i>p</i> -value	No.	Feature	<i>p</i> -value
1	BRUM	0.019017	15	AJA	0.038044
2	SIJ	0.020022	16	GLN(45°)	0.038485
3	SRLGE(90°)	0.021380	17	RP(45°)	0.040108
4	GTFLAP	0.022898	18	RP(135°)	0.040233
5	SRLGE(45°)	0.023014	19	LGRE(0°)	0.041105
6	SRLGE(0°)	0.028114	20	LGRE(90°)	0.041233
7	SRLGE(135°)	0.028445	21	RLN(135°)	0.042278
8	RLN(90°)	0.030137	22	SIJ	0.043803
9	GLN(0°)	0.032553	23	LAP	0.044592
10	RP(0°)	0.032900	24	LGRE(135°)	0.045512
11	RP(90°)	0.033309	25	GLN(135°)	0.046477
12	GLN(90°)	0.033748	26	RLN(45°)	0.046907
13	RLN(0°)	0.034005	27	ACF	0.048627
14	con	0.035178	28	LOG	0.049145

$$SSIM(F, R) = \frac{(2\mu_F\mu_R + C_1)(2\sigma_{FR} + C_2)}{(\mu_F^2 + \mu_R^2 + C_1)(\sigma_F^2 + \sigma_R^2 + C_2)} \quad (16)$$

where σ represents the standard deviation, μ represents the mean intensity, $C_1 = (K_1MAX_I)^2$ and $C_2 = (K_2MAX_I)^2$ with $K_1 = 0.01$ and $K_2 = 0.03$. The higher the *PSNR* and *SSIM* scores the better the restoration results. To more thoroughly evaluate the image restoration ability of the proposed system, the restoration results obtained using the automatic framework are compared with the results obtained using the brute-force method, which is considered as the optimal outcome. Let $PSNR_A$ be the restoration score of the proposed automatic system and

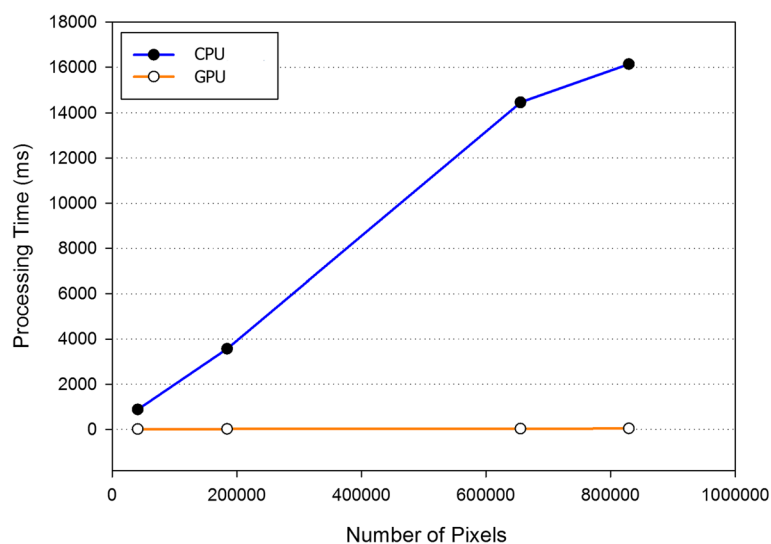


Fig. 3 Plots of the running times of the CPU-based and the GPU-based with shared memory collateral filters

Table 3 Restoration result analyses of the proposed GPU-based collateral filter on different slices

Slice		7	14	17	20	23
Noise						
5%	PSNR _A	33.066	34.131	34.156	33.836	34.736
	PSNR _T	33.229	34.145	34.157	33.855	34.758
	ϵ_r (%)	0.4891	0.0395	0.0031	0.0575	0.0621
7%	PSNR _A	31.278	31.728	31.956	32.015	32.613
	PSNR _T	31.355	31.837	31.960	32.028	32.658
	ϵ_r (%)	0.2453	0.3430	0.0117	0.0415	0.1377

PSNR_T be the restoration score of the brute-force method. The relative error ϵ_r is then defined as

$$\epsilon_r = \frac{|\text{PSNR}_T - \text{PSNR}_A|}{\text{PSNR}_T} \times 100\% \quad (17)$$

Results

We first conducted the experiments on the BrainWeb database [40] as it contains simulated brain MR image data based on normal and multiple sclerosis (MS) models with different thicknesses, noise levels and intensity non-uniformities. Clinical image data

were acquired from the medical image database in the Division of Interventional Neuro Radiology, Department of Radiology, UCLA, Los Angeles, CA, USA. For efficiency tests, the experiments were performed on an Intel® Xeon(R) CPU E5-2620 v3 2.40GHz equipped with a Tesla K40c GPU. Tesla K40c is one NVIDIA's GPU based on the NVIDIA's Kepler architecture. The Tesla K40c GPU contains 15 multiprocessors and each multiprocessor contains 192 cores, which results in 2880 cores. Its memory size is 12 GB and the maximum bandwidth is 288 (GB/sec). Each block has 65,536 registers with 48 KB shared memory and the maximum number of threads is 1024. The CUDA driver version is 7.5. The algorithm was implemented and programmed in MATLAB 2017a (The MathWorks Inc. Natick, MA, USA) associated with C for CUDA-based acceleration.

Table 1 presents the execution performance of the traditional collateral filter and our accelerated version on the same 217×181 brain MR images. The performance of using the shared memory (SM) or not is also indicated. The block size for the shared memory was 16×16 threads and the filter size was 3×3 . When the number of images was increasing, the proposed framework executed the filtering process more effectively and a remarkable acceleration was attained. It was noted that

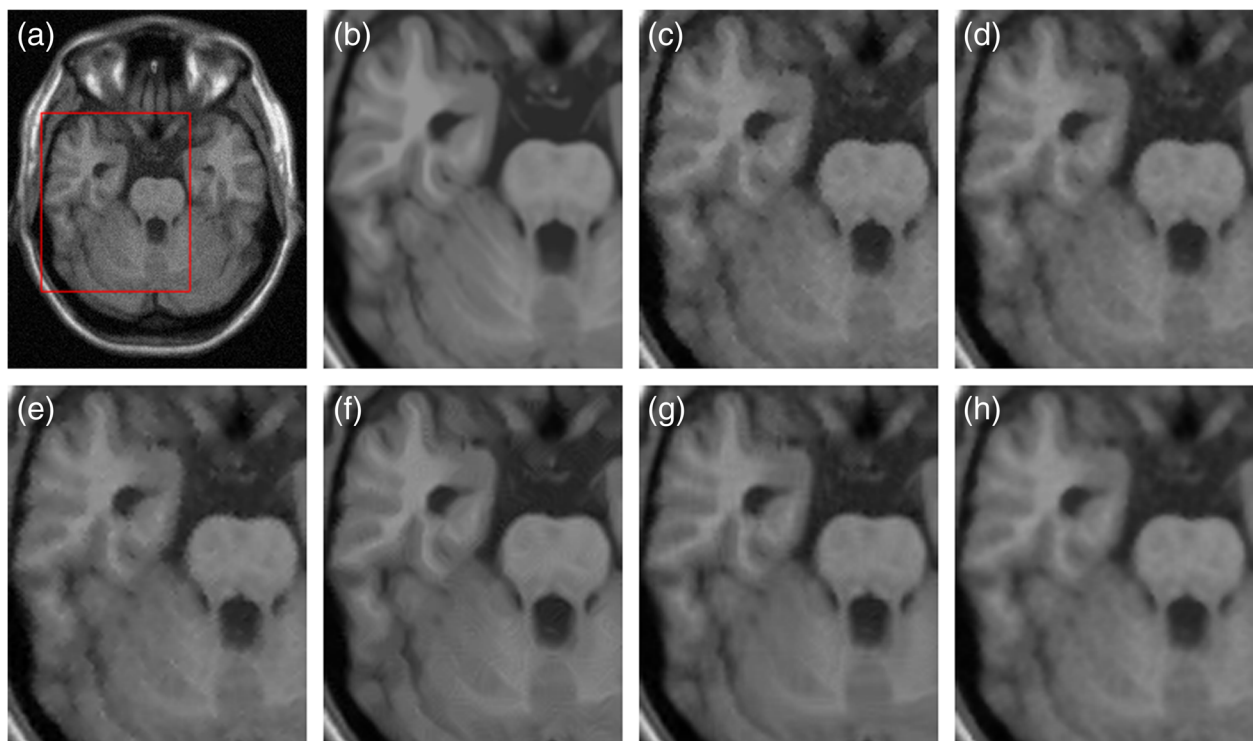


Fig. 4 Comparison of visual restoration results on slice 10 with 5% noise in the 5 mm MS dataset. **a** Noisy MR image. **b** Intact image in magnified view. **c** IBF with PSNR = 32.49 and SSIM = 0.8650. **d** ADF with PSNR = 33.61 and SSIM = 0.8684. **e** LMMSE with PSNR = 32.53 and SSIM = 0.8657. **f** NLN with PSNR = 32.53 and SSIM = 0.8800. **g** BM3D with PSNR = 32.49 and SSIM = 0.8997. **h** Proposed with PSNR = 34.32 and SSIM = 0.8803

the speed up gain raised from 34 for one single image to 541 for processing 100 images. Our proposed GPU-based collateral filter is much faster than the original CPU version. The corresponding running times with respect to different image pixel sizes are depicted in Fig. 3. Obviously, the GPU implementation ran less than 0.1 s for each scenario, whereas the computation time of the CPU version was roughly linearly proportional to the pixel number.

As presented in Table 2, there were 28 selected features from different image feature classes for the input candidates of the BPN system. The order of significance of these adopted image features was based on the average p -value of the t-test outcome with $p < 0.05$. For understanding the optimal features, T1-weighted brain MR images with three different slice thicknesses (1 mm, 3 mm and 5 mm), five noise levels (1, 3, 5, 7 and 9%), and three intensity non-uniformities (0, 20 and 40%) acquired from the BrainWeb database were utilized to evaluate the proposed automatic restoration framework. Each noise level had 1476 images, which were divided into two groups: 984 slices for training and 492 slices for testing. After the SFFS procedure, five optimal features of RLN(0°), RP(135°), RP(90°), AJA, and kurtosis were obtained and they were further employed in the BPN training phase.

In the testing phase, the first stage BPN system classified the input image into three major categories based on the noise level: low (1 and 3%), median (5 and 7%) and high (9%) estimators. The correct classification rate was up to 97.6% overall. The computed five features of the input image were fed into the corresponding second stage BPN model for producing the filter parameter values for denoising. Table 3 summarizes the PSNR scores and the relative error ε_r of the proposed automatic filtering framework and the brute-force method on randomly selected slices of the 5 mm normal MR images with 5 and 7% noise. It is indicated that the PSNR scores of the proposed restoration algorithm were extremely close to the optimal PSNR scores of the brute-force method with negligible errors.

In Fig. 4, we illustrate our automatic restoration outcome of the 10th slice image corrupted by 5% noise level in the 5 mm MS dataset in comparison with the ADF [9], LMMSE [8], NLM [12], iterative bilateral filtering (IBF) [7], and BM3D [13] methods. Comparing to other techniques, the proposed GPU-based collateral filter revealed more uniform intensity profiles and better sharpened edges in the white matter (WM) and gray matter (GM) regions. Quantitative analyses also indicated that our automatic collateral filter provided the highest score of PSNR = 34.32 dB and the second highest SSIM =

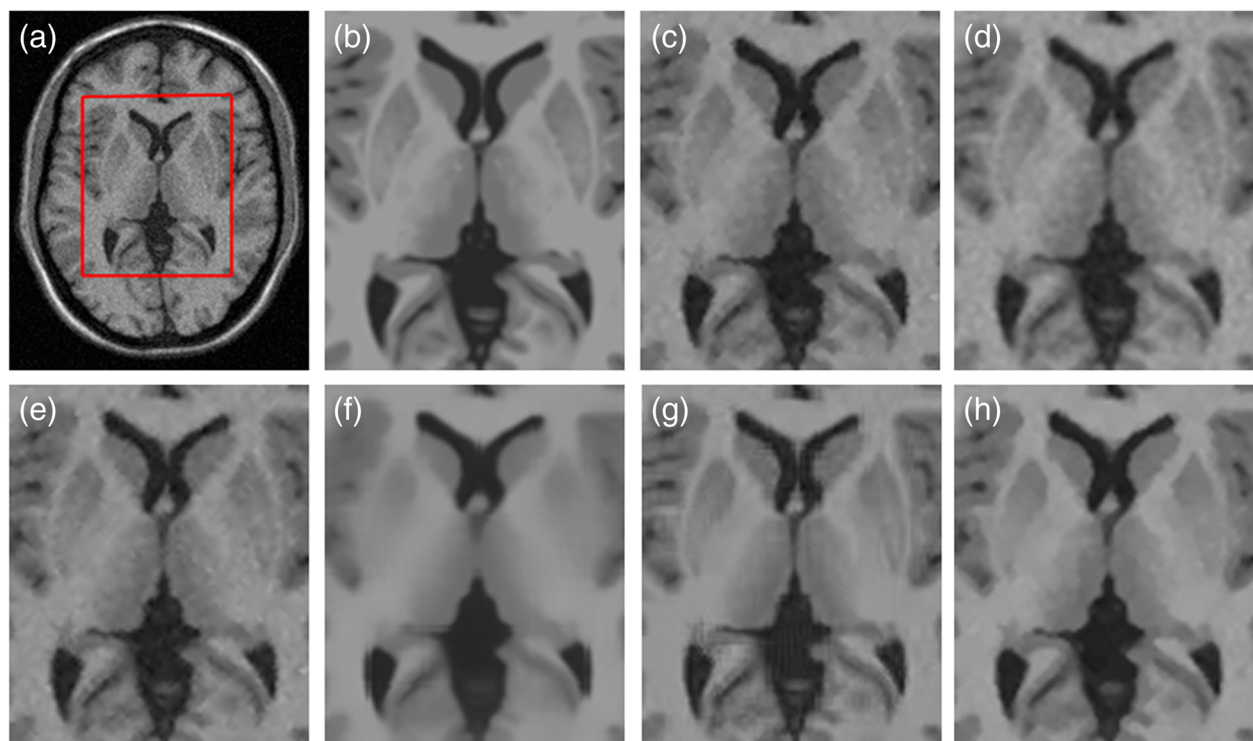


Fig. 5 Comparison of visual restoration results on slice 26 with 7% noise in the 3 mm MS dataset. **a** Noisy MR image. **b** Intact image in magnified view. **c** IBF with PSNR = 28.87 and SSIM = 0.7981. **d** ADF with PSNR = 31.41 and SSIM = 0.8133. **e** LMMSE with PSNR = 29.08 and SSIM = 0.8016. **f** NLM with PSNR = 31.74 and SSIM = 0.8406. **g** BM3D with PSNR = 31.78 and SSIM = 0.8367. **h** Proposed with PSNR = 31.95 and SSIM = 0.8479

0.8803 than other tested methods. Figure 5 depicts the visual restoration results of the 26th slice image corrupted by 7% noise level in the 3 mm MS dataset. It was obvious that, in contrast to other methods, the proposed denoising algorithm achieved a sharper and cleaner representation for the GM and WM with the best PSNR = 31.95 dB and SSIM = 0.8479, which was closer to the intact image.

Figure 6 delineates the restoration results of the 1 mm normal image volume with 9% noise level and 40% intensity non-uniformity in 3D visualization. After applying the ADF, LMMSE, and IBF methods, the brain structures still, more or less, exhibited grainy noise influences. On the other hand, the proposed automatic filter decently wiped out the heavy noise while maintaining noticeable cortical structures as illustrated in Fig. 6. Table 4 presents quantitative analyses on massive 1 mm image volume restoration with various scenarios of anatomical models and noise levels. It was noted that the LMMSE method performed much worse than other

methods in the image volumes with 1% noise. Our automatic denoising framework produced the best restoration results in terms of average PSNR and SSIM in all noise levels. As the noise level was increasing, the advantage of our collateral filter over other methods was more compelling.

For completeness, we demonstrate the capability of our automatic filtering algorithm in restoring clinical brain MR images with diseases. Figure 7 depicts the restoration results of denoising T1-weighted MR images with brain tumors using different methods. While the restoration results obtained using the ADF, LMMSE, and IBF methods were somewhat grainy in the tumor area, the NLM, BM3D and proposed algorithms achieved more suitable smoothness while maintaining sharpness at the tumor boundaries. Another example of filtering distorted PD-weighted MR images with brain tumors is illustrated in Fig. 8. It was indicated that, comparing to other methods, our automatic collateral filter more effectively removed the artefact while disclosing

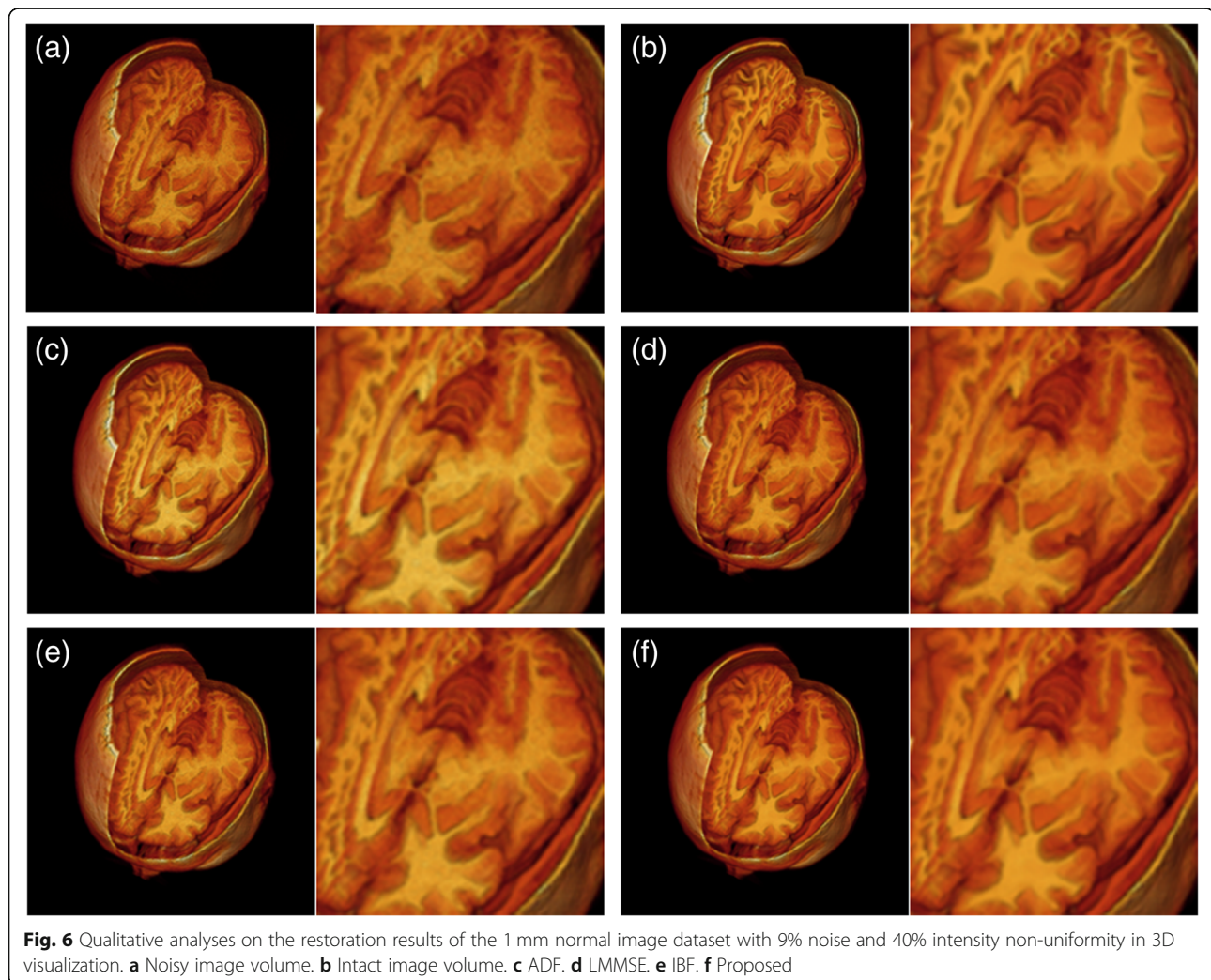


Table 4 Quantitative analyses on the restoration results of the 1 mm image data in terms of average PSNR and SSIM scores using different methods

Type	Noise	LMMSE		IBF		Proposed	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Normal	1%	34.97	0.8513	38.04	0.9766	39.17	0.9816
	3%	33.34	0.8626	33.28	0.8924	34.79	0.9027
	5%	30.65	0.8042	30.51	0.8239	32.42	0.8538
	7%	28.65	0.7535	28.60	0.7711	30.53	0.7959
	9%	27.03	0.7127	27.04	0.7309	29.13	0.7585
MS	1%	35.09	0.8512	38.16	0.9767	39.29	0.9830
	3%	33.48	0.8628	33.40	0.8928	34.87	0.9006
	5%	30.79	0.8046	30.64	0.8244	32.51	0.8540
	7%	28.79	0.7541	28.73	0.7715	30.62	0.7958
	9%	27.17	0.7135	27.16	0.7315	29.21	0.7587

distinct edges between anatomical structures. Finally, in Fig. 9, we show the visual improvement of noisy MR images with low resolution and severe grains using all tested methods. It was obvious that the proposed filtering algorithm more adequately eliminated the noise and identified the lesion in contrast to other techniques.

Discussion

Stimulated by the recent advance in artificial intelligence and parallel computing, we have developed an efficient filtering framework for brain MR

images. An essential property of the proposed automatic restoration algorithm is the two-stage mechanism of the neural network concatenation. A benefit of this strategy is to divide the broad noise range into three narrower classes corresponding to three individual BPN models, where the prediction of the filter parameters can be further clarified. To correlate the image with the BPN models for automation, a wide variety of image texture features were investigated based on the SFFS strategy, from which five best texture features were obtained. Three features of RLN(0°), RP(135°), and RP(90°) belong to the GLRLM category, AJA is one of the noise estimation features, and kurtosis makes use of the histogram information. These five features play an essential role in the effective discrimination between images with different noise levels, anatomical structures, and intensity non-uniformities.

A variety of T1-weighted brain MR images acquired from the BrainWeb database [40] with various scenarios were utilized in the proposed BPN models to obtain the optimal filter parameters. Due to the facility limitations, other MR modalities such as T2-weighted and PD-weighted images were not included in the training dataset in the current study. From the viewpoint of the clinical demand, the proposed automatic filter adequately restored various brain MR images with distinct anatomical structures and modalities such as PD-weighted images (see Figs. 7, 8, 9), which were quite

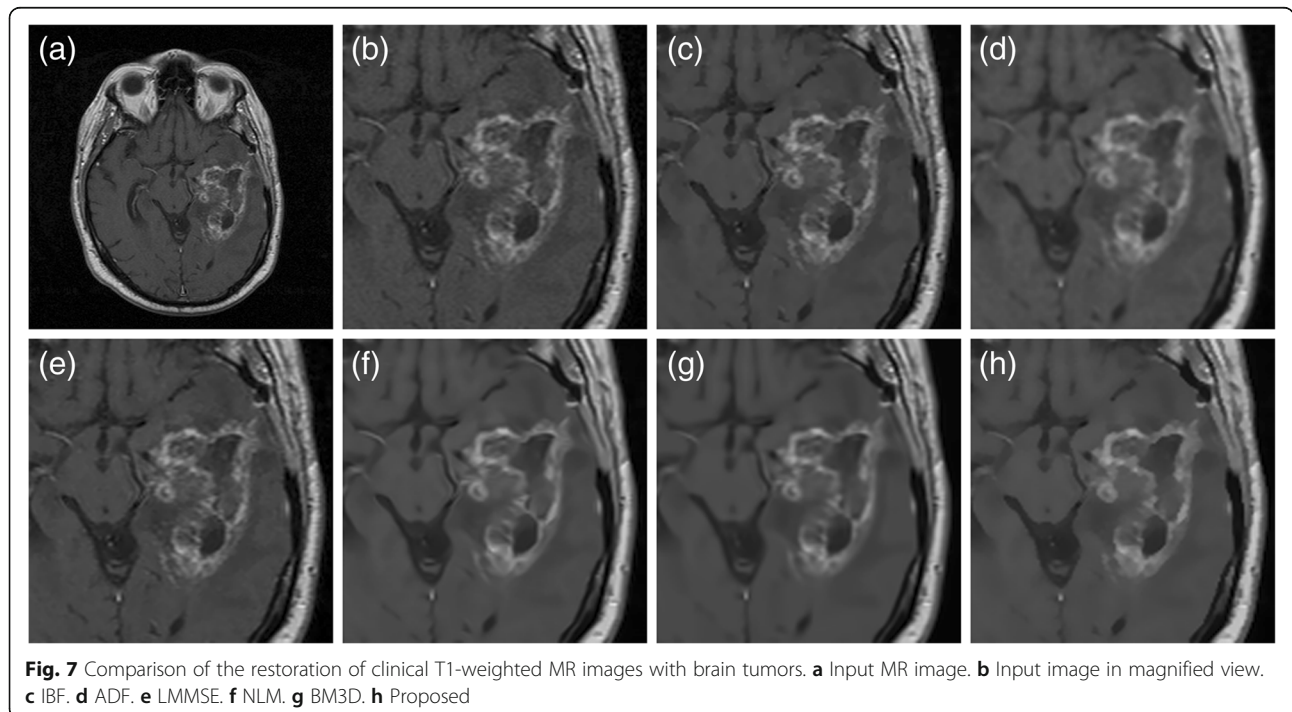


Fig. 7 Comparison of the restoration of clinical T1-weighted MR images with brain tumors. **a** Input MR image. **b** Input image in magnified view. **c** IBF. **d** ADF. **e** LMMSE. **f** NLM. **g** BM3D. **h** Proposed

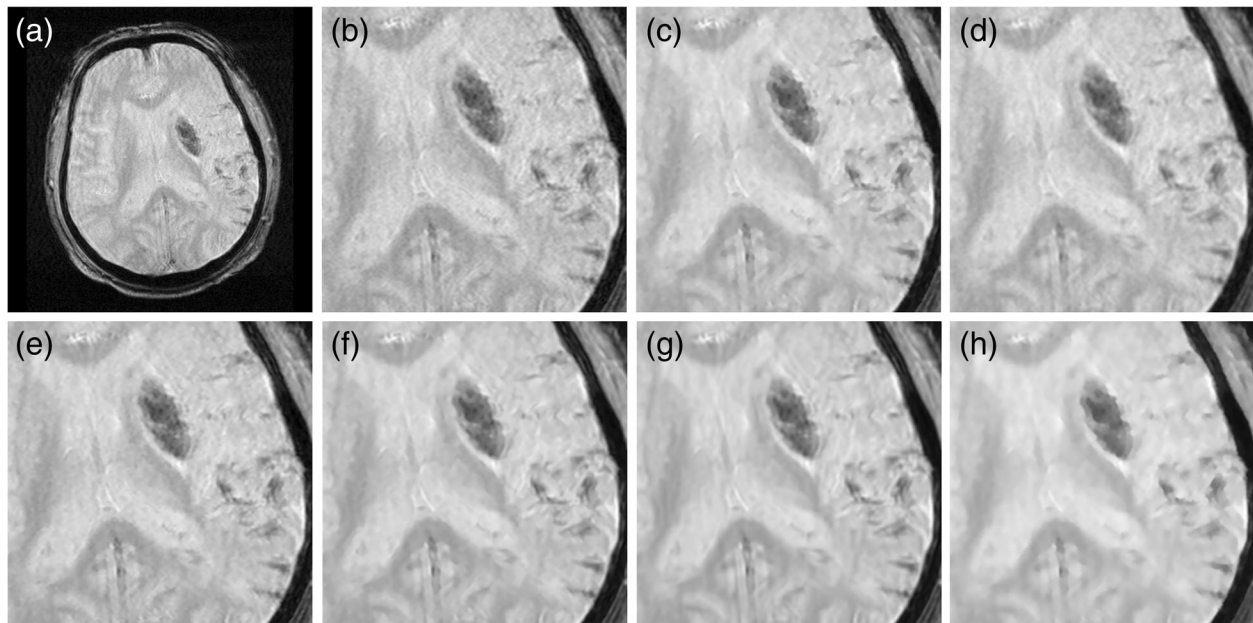


Fig. 8 Comparison of the restoration of distorted PD-weighted MR images with brain tumors. **a** Input MR image. **b** Input image in magnified view. **c** IBF. **d** ADF. **e** LMMSE. **f** NLN. **g** BM3D. **h** Proposed

different from the training BrainWeb dataset. This suggested the propriety of the proposed automation strategy and the robustness of the developed restoration scheme. Nevertheless, to achieve optimal restoration results on massive T2-weighted and PD-weighted brain images, retraining involving these types of images is recommended.

Conclusions

We have described an automatic image restoration algorithm based on the accelerated collateral filter that has been implemented on the GPU architecture. The proposed framework reduced the execution time of the filtering process by taking advantage of shared memory in

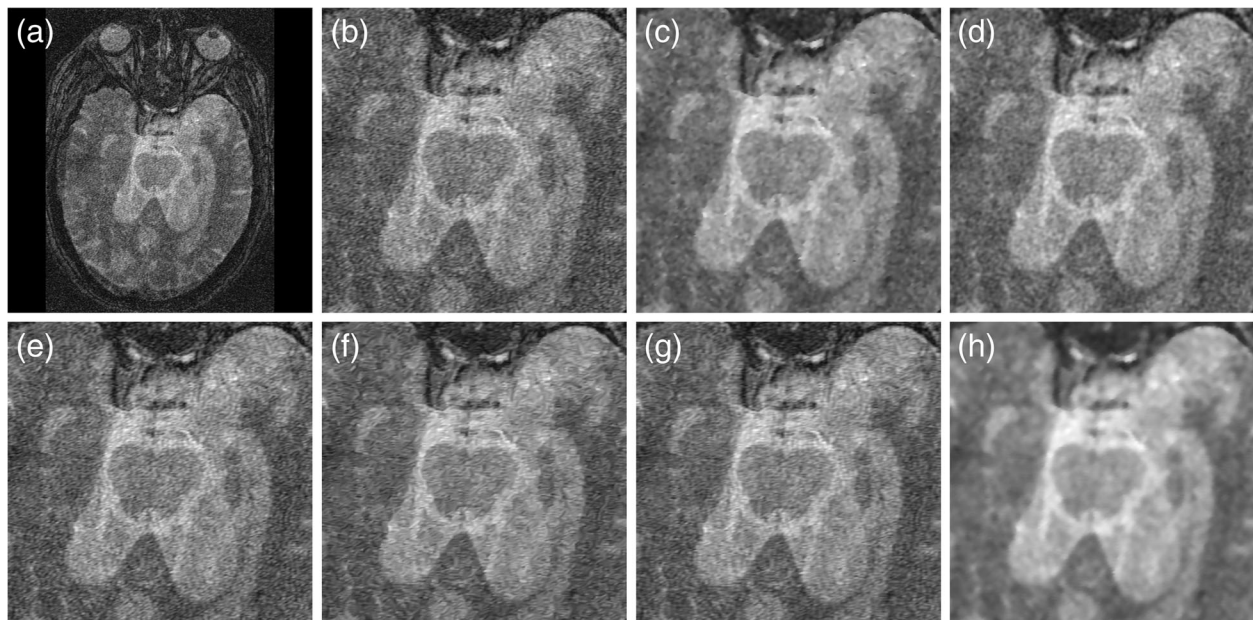


Fig. 9 Comparison of the restoration of noisy MR images with low resolution. **a** Input MR image. **b** Input image in magnified view. **c** IBF. **d** ADF. **e** LMMSE. **f** NLN. **g** BM3D. **h** Proposed

CUDA to decrease memory access latency. This new denoising system fully automated the restoration procedure for brain MR images without the need of adjusting the filter parameters. Experimental results indicated that the proposed automatic restoration framework effectively removed noise in various brain MR images with satisfactory quantity and quality. We also demonstrated its great capability of improving the clinical MR image quality and delineating the tumor boundary that outperformed other tested methods. We believe that our automatic filtering framework is of potential and able to provide a fast and powerful solution in a wide variety of medical image restoration applications.

Additional file

Additional file 1: Formulae. (DOCX 19 kb)

Abbreviations

ADF: Anisotropic diffusion filter; BPN: Propagation neural network; CPU: Central processing unit; CUDA: Compute unified device architecture; GLCM: Gray-level co-occurrence matrix; GLN: Gray level nonuniformity; GLRLM: Gray-level run-length matrix; GM: Gray matter; GPGPU: General purpose computation on GPU; GPU: Graphics processing unit; HGRE: High gray-level run emphasis; IBF: Iterative bilateral filtering; LGRE: Low gray-level run emphasis; LMMSE: Linear minimum mean square error; LOG: Laplacian of Gaussian; LRE: Long run emphasis; LRHGE: Long run high gray-level emphasis; LRLGE: Long run low gray-level emphasis; MRI: Magnetic resonance imaging; MS: Multiple sclerosis; PSNR: Peak signal-to-noise ratio; RLN: Run length nonuniformity; RP: Run percentage; SD: Standard deviation; SFFS: Sequential forward floating selection; SM: Shared memory; SRE: Short run emphasis; SRHGE: Short run high gray-level emphasis; SRLGE: Short run low gray-level emphasis; SSIM: Structural similarity; WM: White matter

Funding

This work was supported by the Ministry of Science and Technology of Taiwan under Grant No. MOST 106–2221-E-002-082. The funder played no role in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

Availability of data and materials

BrainWeb data are fully available without restriction. Due to patient privacy protection, clinical data used for the current study are available from the corresponding author on reasonable request.

Authors' contributions

HC conceived the research, designed the algorithm, analyzed the results, and wrote the manuscript. CL implemented the algorithm, executed the experiments, and contributed to the paper writing. All authors read and approved the final manuscript.

Ethics approval and consent to participate

Clinical images in this article were acquired from the medical image database in the Division of Interventional Neuro Radiology, Department of Radiology, University of California at Los Angeles (UCLA) and are subject to an approved Human Research Subject Protection Protocol.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 28 December 2017 Accepted: 4 January 2019

Published online: 19 January 2019

References

1. Van Leemput K, Maes F, Vandermeulen D, Suetens P. Automated model-based tissue classification of MR images of the brain. *Medical Imaging, IEEE Transactions on*. 1999;18(10):897–908.
2. Pelizzari CA, Chen GT, Spelbring DR, Weichselbaum RR, Chen C-T. Accurate three-dimensional registration of CT, PET, and/or MR images of the brain. *J Comput Assist Tomogr*. 1989;13(1):20–6.
3. Shahid MLUR, Chitiboi T, Ivanovska T, Molchanov V, Völzke H, Linsen L. Automatic MRI segmentation of Para-pharyngeal fat pads using interactive visual feature space analysis for classification. *BMC Med Imaging*. 2017;17(1):15.
4. Macovski A. Noise in MRI. *Magn Reson Med*. 1996;36(3):494–7.
5. Zheng H, Qu X, Bai Z, Liu Y, Guo D, Dong J, Peng X, Chen Z. Multi-contrast brain magnetic resonance image super-resolution using the local weight similarity. *BMC Med Imaging*. 2017;17(1):6.
6. Tomasi C, Manduchi R. Bilateral filtering for gray and color images. *IEEE Proc Int Conf Comput Vis*. 1998:839–46.
7. Riji R, Rajan J, Sijbers J, Nair MS. Iterative bilateral filter for Rician noise reduction in MR images. *SIVIP*. 2015;9(7):1543–8.
8. 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*. 2008;17(8):1383–98.
9. Perona P, Malik J. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans Pattern Anal Machine Intell*. 1990;12(7):629–39.
10. 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*. 2013;51(1–2):71–88.
11. Tricot B, Descoteaux M, Dumont M, Chagnon F, Tremblay L, Carpentier A, Lesur O, Lepage M, Lalonde A. Improving the evaluation of cardiac function in rats at 7T with denoising filters: a comparison study. *BMC Med Imaging*. 2017;17(1):62.
12. Manjón JV, Carbonell-Caballero J, Lull JJ, García-Martí G, Martí-Bonmati L, Robles M. MRI denoising using non-local means. *Med Image Anal*. 2008;12(4):514–23.
13. Dabov K, Foi A, Katkovnik V, Egiazarian K. Image Denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans Image Process*. 2007;16(8):2080–95.
14. Chang H-H, Chu W-C. Collateral filtering of magnetic resonance images. In: *Biomedical Imaging: From Nano to Macro*, 2010 IEEE International Symposium on: 2010. Rotterdam: IEEE; 2010. p. 728–31.
15. Jouppi N, Young C, Patil N, Patterson D. Motivation for and evaluation of the first tensor processing unit. *IEEE Micro*. 2018;38(3):10–9.
16. CUDA ZONE [<http://www.nvidia.com/cuda>]. Accessed 24 May 2017.
17. Li C-Y, Chang H-H. CUDA-based acceleration of collateral filtering in brain MR images. In: *Eighth International Conference on Graphic and Image Processing*. vol. 2017. Tokyo: SPIE; 2017. p. 5.
18. Wang S-H, Sun J, Phillips P, Zhao G, Zhang Y-D. Polarimetric synthetic aperture radar image segmentation by convolutional neural network using graphical processing units. *J Real-Time Image Proc*. 2018;15(3):631–42.
19. Zhang Y-D, Muhammad K, Tang C. Twelve-layer deep convolutional neural network with stochastic pooling for tea category classification on GPU platform. *Multimed Tools Appl*. 2018;77(17):22821–39.
20. Kirk D. NVIDIA CUDA software and GPU parallel computing architecture. *ISMM*. 2007;2007:103–4.
21. Nickolls J, Buck I, Garland M, Skadron K. Scalable parallel programming with CUDA. *Queue*. 2008;6(2):40–53.
22. Frosio I, Egiazarian K, Pulli K. Machine learning for adaptive bilateral filtering. In: *Image Processing: Algorithms and Systems XIII*: 2015; 2015: 939908–939912.
23. Chaudhury KN, Sage D, Unser M. Fast bilateral filtering using trigonometric range kernels. *Image Processing, IEEE Transactions on*. 2011;20(12):3376–82.
24. Yang Q, Tan K-H, Ahuja N. Real-time O(1) bilateral filtering. In: *Computer Vision and Pattern Recognition, 2009 CVPR 2009 IEEE Conference on*: 2009. Miami Beach, FL: IEEE; 2009. p. 557–564.

25. Haralick RM, Shanmugam K, Dinstein IH. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*. 1973;(6): 610–21.
26. George LE, Mohammed EZ. Tissues image retrieval system based on co-occurrence, run length and roughness features. In: *Computer medical applications (ICCMA), 2013 international conference on: 20–22 Jan. 2013* 2013; 2013. p. 1–6.
27. Sompong C, Wongthanavas S. MRI brain tumor segmentation using GLCM cellular automata-based texture feature. In: *Computer science and engineering conference (ICSEC), 2014 international: July 30 2014-Aug. 1 2014* 2014; 2014. p. 192–7.
28. Galloway MM. Texture analysis using gray level run lengths. *Computer graphics and image processing*. 1975;4(2):172–9.
29. Tamura H, Mori S, Yamawaki T. Textural features corresponding to visual perception. *Systems, Man and Cybernetics, IEEE Transactions on*. 1978;8(6): 460–73.
30. Immerkaer J. Fast noise variance estimation. *Comput Vis Image Underst*. 1996;64(2):300–2.
31. Kutty K, Ojha S. A generic transfer function based technique for estimating noise from images. *Int. J. Comput. Appl*. 2012;51(10).
32. Aja-Fernández S, Tristán-Vega A, Alberola-López C. Noise estimation in single-and multiple-coil magnetic resonance data based on statistical models. *Magn Reson Imaging*. 2009;27(10):1397–409.
33. Sim K, Lai M, Tso C, Teo C. Single image signal-to-noise ratio estimation for magnetic resonance images. *J Med Syst*. 2011;35(1):39–48.
34. Pudil P, Novovičová J, Kittler J. Floating search methods in feature selection. *Pattern Recogn Lett*. 1994;15(11):1119–25.
35. Fisher RA, Genetiker S, Genetician S, Britain G, Généticien S. Statistical methods for research workers, vol. In: 14: Oliver and Boyd Edinburgh; 1970.
36. Student. The probable error of a mean. *Biometrika*. 1908:1–25.
37. McClelland JL, Rumelhart DE, Group PR. Parallel distributed processing. *Explorations in the microstructure of cognition*. 1986;2.
38. Marquardt DW. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Ind. Appl. Math*. 1963;11(2):431–41.
39. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process*. 2004;13(4): 600–12.
40. Vijaykumar VR, Vanathi PT, Kanagasabapathy P. Adaptive window based efficient algorithm for removing Gaussian noise in gray scale and color images. *IEEE Proc ICCIMA*. 2007;3:319–23.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

