J Math Imaging Vis (2012) 43:135–142
DOI 10.1007/s10851-011-0295-x

Application of Lattice Boltzmann Method to Image Filtering

Wenhuan Zhang · Baochang Shi

Published online: 18 May 2011

Abstract In this paper, lattice Boltzmann D2Q5 (two dimensions and five discrete velocity directions) and D2Q9 (two dimensions and nine discrete velocity directions) models are used to solve Perona-Malik equation, which is widely used in image filtering. A set of images added three types of noise are processed using these models. Then the processed images are compared in aspects of peak signal to noise ratio (PSNR) and visual effect. The comparison show that two models have almost the same filtering effect. Simultaneously, it is validated that D2Q5 model is more efficient. Other findings are: (1) D2Q5 and D2Q9 models are more effective in dealing with some images than others; (2) salt and pepper noise is relatively difficult to remove compared with gaussian noise and speckle noise; (3) lattice Boltzmann method shows good stability in the image filtering.

Keywords Lattice Boltzmann method · Perona-Malik equation · Image filtering · Nonlinear diffusion equation

1 Introduction

Originating from the initial value problem for heat conduction equation, partial differential equations (PDEs) began as a tool to be used in the field of image processing. The heat conduction equation filters images utilizing the diffusion process. However, the edge of images can not be preserved well by this equation because the Laplacian operator in the equation is linear. To solve this problem, Perona and Malik proposed a novel diffusion equation [1] which can maintain the edge effectively. After that, many other models [2–4] are proposed based on the Perona-Malik equation (P-M equation).

To solve P-M equation, many numerical methods can be used, such as finite difference method, finite element method, finite volume method and so on. Among them, the lattice Boltzmann (LB) method is a relatively new numerical method emerging in recent years. It has been extended successfully to solve the reaction-diffusion equation [5], nonlinear evolution equation [6–8], wave equation [9], Poisson equation [10, 11] and convection-diffusion equation [12–14]. Besides, "the lattice Boltzmann method provides the capability of handling complicated boundaries and easily implemented fully parallel algorithms" [15]. This makes it probably be a highly efficient numerical method to solve P-M equation which is a special case of the convection-diffusion equation. That is to say, image can probably be filtered efficiently with the lattice Boltzmann method.

In fact, the lattice Boltzmann method has been applied to image filtering by few researchers [15–20]. In Ref. [15], the lattice Boltzmann D2Q9 model for nonlinear diffusion filtering was firstly proposed and the feasibility of the lattice Boltzmann model for image filtering was verified. After that, the lattice Boltzmann model for diffusion equation was extended to 3D space for volume smoothing and surface denoising in Refs. [16, 17]. In Refs. [18, 19], a new lattice Boltzmann D2Q9 model for medical image smoothing was proposed by introducing a medium between the nodes of the lattice based on Ref. [21]. In Ref. [20], the lattice Boltzmann scheme was used to simulate the well known total variation based restoration model, that is, ROF model. With satisfactory restored images, the computational speed of LB method

W. Zhang
State Key Laboratory of Coal Combustion, Huazhong University of Science and Technology, Wuhan 430074, P.R. China

B. Shi (✉)
School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan 430074, P.R. China
e-mail: shibc@mail.hust.edu.cn

is much faster than the iterative fixed point method which is another method to solve ROF model.

The main reason that authors in Refs. [15–20] investigate the application of lattice Boltzmann method to image filtering is that this method provides the easily implemented fully parallel algorithms which fulfill the intrinsic features of parallelism. The authors have paid much attention to this advantage, but they omit the efficient difference between different lattice Boltzmann models, such as the difference between D2Q5 and D2Q9, which are two commonly used 2-dimensional models. As we know, D2Q5 model has five evolution directions but D2Q9 has nine, this implies D2Q5 model may be more efficient. But what about the difference in filtering effect? The motivation of this paper is to compare the difference in filtering effect between two models with Perona-Malik equation, and simultaneously validate the higher efficiency of D2Q5 model. The difference in filtering effect is evaluated by testing a set of images which show different degrees of contrast, edge details, texture, etc. Considering the noise type may affect result, we add three types of noise to each image respectively. The higher efficiency of D2Q5 model is validated by comparing the time which is used to carry out the same iterations. Before tests, we fix two experimental parameters in reasonable ways, and find that lattice Boltzmann method shows good stability in the image filtering. Finally, through lots of tests, we compare the differences in filtering effect and efficiency between D2Q5 and D2Q9 models, and find that two models have almost the same capacity, but D2Q5 model is more efficient.

The paper is organized as follows. The lattice Boltzmann model for P-M equation is introduced in Sect. 2. Experiments and discussions are presented in Sect. 3. Finally, a brief conclusion is given in Sect. 4.

2 Lattice Boltzmann Model for P-M Equation

The P-M equation can be written as follows [1]:

$$\begin{cases} \frac{\partial u}{\partial t} = \nabla \cdot (\alpha(|\nabla u|)\nabla u) \\ u(\mathbf{x},0) = u_0(\mathbf{x}), \end{cases} \tag{1}$$

where $\mathbf{x}$ is the pixel, $u(\mathbf{x},t)$ is the intensity of an image at position $\mathbf{x}$ and time t, $u_0(\mathbf{x})$ is the intensity of an image at time $t=0$, $\alpha(\cdot)$ is a positive nonincreasing diffusion coefficient. In Ref. [1], Perona and Malik provided two diffusion coefficients, $\alpha(s) = e^{-(s/k)^2}$ and $\alpha(s) = \frac{1}{1+(s/k)^2}$, where k (threshold value) can be preset or altered as image iterates, s is the modulus of image gradient, that is $|\nabla u|$.

Recently, a general lattice Boltzmann model for nonlinear convection-diffusion equation has been proposed by Shi and Guo in [14]. Since P-M equation is a special convection-diffusion equation, we can establish a corresponding D2Qb model with b velocity directions in 2D space. The evolution equation of the model reads

$$u_i(\mathbf{x}+\mathbf{c}_i\Delta t, t+\Delta t) = u_i(\mathbf{x},t) - \frac{1}{\tau}[u_i(\mathbf{x},t) - u_i^{(\mathrm{eq})}(\mathbf{x},t)],$$
$$i = 0,\ldots,b-1, \tag{2}$$

where $\{\mathbf{c}_i, i=0,\ldots,b-1\}$ is the set of discrete velocity directions, b is the direction number, Δt is the time step, τ is the dimensionless relaxation time, $u_i(\mathbf{x},t)$ and $u_i^{(\mathrm{eq})}(\mathbf{x},t)$ are the density distribution function and the equilibrium density distribution function, respectively.

The equilibrium distribution function is taken as

$$u_i^{(\mathrm{eq})}(\mathbf{x},t) = \alpha_i u(\mathbf{x},t). \tag{3}$$

For D2Q9 model,

$$\mathbf{c}_i = \begin{cases} (0,0) \quad i=0, \\ (\cos[(i-1)\pi/2], \sin[(i-1)\pi/2])c \quad i=1,2,3,4, \\ (\cos[(i-5)\pi/2+\pi/4], \sin[(i-5)\pi/2+\pi/4])\sqrt{2}c \\ \quad i=5,6,7,8, \end{cases}$$

where $c = \frac{\Delta x}{\Delta t}$, Δx is the lattice spacing, and

$$\alpha_i = \begin{cases} 4/9 & i=0, \\ 1/9 & i=1,2,3,4, \\ 1/36 & i=5,6,7,8. \end{cases}$$

For D2Q5 model,

$$\mathbf{c}_i = \begin{cases} (0,0) & i=0, \\ (\cos[(i-1)\pi/2], \sin[(i-1)\pi/2])c & i=1,2,3,4, \end{cases}$$

$$\alpha_i = 1/5 \quad i=0,1,2,3,4.$$

Note that $\sum_i \alpha_i = 1$, and $u(\mathbf{x},t)$ is defined by $u(\mathbf{x},t) = \sum_i u_i(\mathbf{x},t)$, so $u_i(\mathbf{x},t)$ and $u_i^{(\mathrm{eq})}(\mathbf{x},t)$ satisfy

$$\begin{aligned} &\sum_i u_i = \sum_i u_i^{(\mathrm{eq})} = u, \\ &\sum_i \mathbf{c}_i u_i^{(\mathrm{eq})} = 0, \qquad \sum_i \mathbf{c}_i\mathbf{c}_i u_i^{(\mathrm{eq})} = c_s^2 u I, \end{aligned} \tag{4}$$

where $c_s^2 = c^2/3$ and $c_s^2 = 2c^2/5$ for D2Q9 and D2Q5 model, respectively.

To derive the macroscopic equation (1), we can use the following Chapman-Enskog expansion procedure:

$$\begin{aligned} &u_i = u_i^{(0)} + \epsilon u_i^{(1)} + \epsilon^2 u_i^{(2)}, \\ &\partial_t = \epsilon\partial_{t_1} + \epsilon^2\partial_{t_2}, \qquad \nabla = \epsilon\nabla_1, \end{aligned} \tag{5}$$

where $u_i^{(0)} = u_i^{(\mathrm{eq})}$, ϵ is a small expansion parameter, $t_1 = t/\epsilon$ and $t_2 = t/\epsilon^2$ are two macroscopic time scales, ∇_1 is the gradient operator in the macroscopic length scale $x_1 = x/\epsilon$.

Summing the first formula in (5) over i and using the first formula in (4), we have

$$\sum_i u_i^{(k)} = 0 \quad (k \ge 1). \tag{6}$$

By applying Taylor expansion to (2), we get

$$D_i u_i + \frac{\Delta t}{2} D_i^2 u_i + \cdots = -\frac{1}{\tau \Delta t}(u_i - u_i^{(\mathrm{eq})}), \tag{7}$$

where $D_i = \partial_t + \mathbf{c}_i \cdot \nabla$. Denote $D_{1i} = \partial_{t_1} + \mathbf{c}_i \cdot \nabla_1$. Substituting (5) into (7) and collecting the terms in order of ϵ and ϵ^2 separately, gives

$$D_{1i} u_i^{(\mathrm{eq})} = -\frac{1}{\tau \Delta t} u_i^{(1)}, \tag{8}$$

$$\partial_{t_2} u_i^{(\mathrm{eq})} + D_{1i} u_i^{(1)} + \frac{\Delta t}{2} D_{1i}^2 u_i^{(\mathrm{eq})} = -\frac{1}{\tau \Delta t} u_i^{(2)}. \tag{9}$$

Substituting (8) to the left side of (9) and combining the terms including $u_i^{(1)}$ on the left side, we can rewrite (9) as

$$\partial_{t_2} u_i^{(\mathrm{eq})} + D_{1i}\left[\left(1 - \frac{1}{2\tau}\right) u_i^{(1)}\right] = -\frac{1}{\tau \Delta t} u_i^{(2)}. \tag{10}$$

Summing (8) and (10) over i and using (4) and (6), we obtain

$$\partial_{t_1} u = 0, \tag{11}$$

$$\partial_{t_2} u + \nabla_1 \cdot \left[\left(1 - \frac{1}{2\tau}\right) \sum_i \mathbf{c}_i u_i^{(1)}\right] = 0. \tag{12}$$

Using (8) and (4) we have

$$\begin{aligned}\sum_i \mathbf{c}_i u_i^{(1)} &= -\tau \Delta t \sum_i \mathbf{c}_i D_{1i} u_i^{(\mathrm{eq})} \\ &= -\tau \Delta t c_s^2 \nabla_1 u.\end{aligned} \tag{13}$$

So, substituting (13) into (12), we obtain

$$\partial_{t_2} u = \nabla_1 \cdot \left\{\left[c_s^2\left(\tau - \frac{1}{2}\right)\Delta t\right] \nabla_1 u\right\}. \tag{14}$$

Combining (14) and (11), and taking $\alpha = c_s^2(\tau - \frac{1}{2})\Delta t$,we have

$$\partial_t u = \nabla \cdot (\alpha \nabla u), \tag{15}$$

that is to say the P-M equation is exactly recovered to order $O(\epsilon^2)$ using the proposed models.

3 Numerical Experiments and Discussions

Our experiments are carried out as follows. Firstly, we select one classic black-and-white image in Fig. 1, which has the size of 256×256, with the value varying in [0, 1]. Secondly, one type of noise is added to the image using Matlab, which is gaussian noise with average value 0 and variance 0.01 or salt and pepper noise with intensity 0.05 or speckle noise with average value 0 and variance 0.04. Thirdly, based on the proposed lattice Boltzmann model for P-M diffusion equation, $u_i^{eq}(\mathbf{x}, 0)$ is computed with (3) where $u(\mathbf{x}, 0)$ is the added noise image. After assigning $u_i(\mathbf{x}, 0)$ with $u_i^{eq}(\mathbf{x}, 0)$, $u_i(\mathbf{x}, \Delta t)$ is computed with (2).

Before the computation, we need to calculate τ in (2), which is related to $\alpha(|\nabla u|)$. The formula to compute ∇u in the internal region of image for D2Q9 and D2Q5 model is

$$\nabla u(\mathbf{x}) = \frac{1}{\beta \Delta x} \sum_{i=0}^{l} \mathbf{e}_i u(\mathbf{x} + \mathbf{c}_i \Delta t), \tag{16}$$

where β is set to be 6.0 and 2.0 for D2Q9 model and D2Q5 model respectively, l is set to be 8 and 4 for D2Q9 model and D2Q5 model respectively, $\mathbf{e}_i = \mathbf{c}_i / c$. On the boundaries, the image gradient is composed of two components, one is with the direction normal to the boundaries, the other one is with the direction tangent to the boundaries. Because we use the adiabatic boundary condition for $u(\mathbf{x}, t)$, i.e., the intensity flux along the outward normal direction of the boundaries is zero, so the normal component of image gradient is zero and we only need to compute the tangent one. We firstly compute the tangent component of image gradient on four boundaries excluding four angular points with the second order forward difference. Secondly, two upper angular points are viewed as points on the upper boundary, the tangent component of image gradient at the upper left and right angular point is calculated with one order forward and backward difference, respectively. The treatment of two lower angular points is in the same way.

By summing $u_i(\mathbf{x}, \Delta t)$ with the first formula in (4), we get $u(\mathbf{x}, \Delta t)$ in the internal region of image. But the values $u(\mathbf{x}, \Delta t)$ and $u_i(\mathbf{x}, \Delta t)$ on the boundaries are still unknown. To solve this problem, we use the adiabatic boundary condition to calculate $u(\mathbf{x}, \Delta t)$ and non-equilibrium extrapolation method [22] to calculate $u_i(\mathbf{x}, \Delta t)$ on the boundaries. The specific implementation of boundary condition is as follows. Firstly, according to the adiabatic boundary condition for $u(\mathbf{x}, t)$, $u(B, \Delta t)$ is set equally to $u(N, \Delta t)$, where points B and N are shown in Fig. 2 and $u(N, \Delta t)$ has been calculated before the boundary treatment. Secondly, based on the first formula in (5), the density distribution function $u_i(\mathbf{x}, t)$ for every pixel and at any time can be decomposed into two parts, i.e., $u_i(\mathbf{x}, t) = u_i^{eq}(\mathbf{x}, t) + u_i^{neq}(\mathbf{x}, t)$, where $u_i^{neq}(\mathbf{x}, t)$ is the non-equilibrium part of $u_i(\mathbf{x}, t)$. The basic idea of non-equilibrium extrapolation method in [22] is to approximate the non-equilibrium part of boundary point with that of the internal point next to it, i.e., $u_i^{neq}(B, \Delta t) \approx u_i^{neq}(N, \Delta t)$, where $u_i^{neq}(N, \Delta t) =$

Fig. 1 Images used in experiments

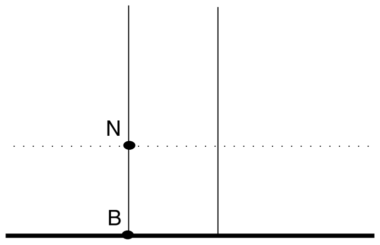


Fig. 2 The *heavy solid line* represents the image boundary, the *dashed line* represents the grid line in the internal region of image. B is the point on the boundary, N is the point next to B in the internal region of image

$u_i(N, \Delta t) - u_i^{eq}(N, \Delta t)$. So we can calculate $u_i(B, \Delta t)$ with the formula $u_i(B, \Delta t) = u_i^{eq}(B, \Delta t) + u_i^{neq}(B, \Delta t) \approx u_i^{eq}(B, \Delta t) + u_i(N, \Delta t) - u_i^{eq}(N, \Delta t)$. Because $u(B, \Delta t)$ and $u(N, \Delta t)$ are equal, $u_i^{eq}(B, \Delta t)$ is equal to $u_i^{eq}(N, \Delta t)$ according to (3). Thus, $u_i(B, \Delta t) \approx u_i(N, \Delta t)$, i.e., density distribution function at the boundary point is fixed with that of internal point next to it. Calculated with the above method, $u(\mathbf{x}, \Delta t)$ and $u_i(\mathbf{x}, \Delta t)$ on the boundaries are well determined. Similar to the computation of image gradient, two upper and lower angular points are again viewed as points on the upper and lower boundary, respectively.

Now we get $u(\mathbf{x}, \Delta t)$ for every pixel at time Δt, i.e., we get the image which has iterated one step. This process continues till the noise added to the image is filtered effectively. After the added noise images are processed, we use Matlab to show the results, and the visual effect of image filtering can be easily distinguished. The processing effect is also evaluated with the quantitative index PSNR, which is defined as

$$PSNR = 10 \log_{10}\left(\sum_{x,y=1}^{256} \frac{1.0^2}{(u(x,y,t) - u_o(x,y))^2}\right), \qquad (17)$$

where $u(x, y, t)$ is the processed version for the added noise image $u(x, y, 0)$ at time t, $u_o(x, y)$ is the original image without noise, (x, y) is the pixel. Generally speaking, the larger PSNR value means the higher fidelity. In our experiments, diffusion coefficient is chosen as $\alpha(s) = \frac{1}{1+(s/k)^2}$.

3.1 Two Parameters k and c

There are two parameters in our experiment which are k in the expression of $\alpha(s)$ and $c = \Delta x/\Delta t$ in (2). Before experiments, they need to be fixed in a reasonable way.

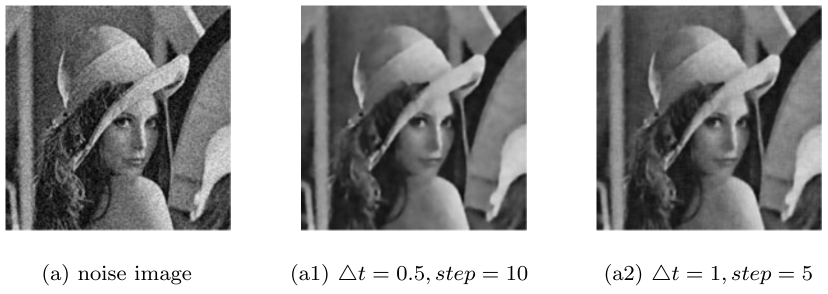

(a) noise image
(a1) $\triangle t = 0.5, step = 10$
(a2) $\triangle t = 1, step = 5$

Fig. 3 (**a**) is the Lenna image added gaussian noise, (**a1**), (**a2**) are the processed versions using D2Q9 model, Δt denotes the time step, *step* denotes the iteration step, the PSNR values for (**a**), (**a1**) and (**a2**) are 20.28, 27.04 and 26.94, respectively

As used by Perona and Malik in Ref. [1], k is fixed either by hand at some fixed value or using the "noise estimator" described by Canny [23]: a histogram of the absolute values of the gradient throughout the image is computed, and k is set equal to the 90% value of its integral at every iteration, i.e., $k = 0.9\sum|\nabla u|/(M \times N)$, where the sum is computed over the whole image region, $M \times N$ is resolution of the image. In our experiments, we fix k with the second method. The reason is that the second method compared with the first one makes the algorithm insensitive to the parameter Δx and self-adaptive to different images and noises. From the formula to calculate ∇u and k, we know $|\nabla u|$ and the k fixed with the second method are both related to Δx, they increases by the same order of magnitude, when Δx decreases. But in the diffusion coefficient $\alpha(|\nabla u|) = 1/(1 + (\frac{|\nabla u|}{k})^2)$, $|\nabla u|$ is divided by k, so the effect of Δx disappears in it, i.e., Δx dose not affect the filtering process. In this sense, the second method is insensitive to the parameter Δx and avoids choosing different k for different Δx compared with the first method. On the other hand, because k is computed with the formula $0.9\sum|\nabla u|/(M \times N)$, so k is self-adaptive to different image added different types of noise, which reduces the procedure to specify different k for different noisy image in the first method. Considering above two factors, we choose the second method to fix k in our experiments.

Besides k, c is also an important parameter. Because $c = \Delta x/\Delta t$ and $\Delta x = 1$ is set in advance without loss of generality, the discussion on c is equivalent to the discussion on Δt. Inferred from $\alpha = c_s^2(\tau - \frac{1}{2})\Delta t$, Δt satisfies

$$\tau = \frac{\gamma \Delta t}{(1 + (\frac{|\nabla u|}{k})^2)} + \frac{1}{2}, \tag{18}$$

where γ is a constant with the value of 3.0 and 2.5 for D2Q9 and D2Q5 model respectively. As we know, the stability condition for lattice Boltzmann method is $\tau > 1/2$, this is equivalent to $\Delta t > 0$ according to (18). Note that lattice Boltzmann method shows good stability in the image filtering. If P-M equation is discretized by forward approximation in time and central differences in space, the stability condition [24] is

$$0 < \Delta t \le (\Delta x)^2/(4\alpha), \tag{19}$$

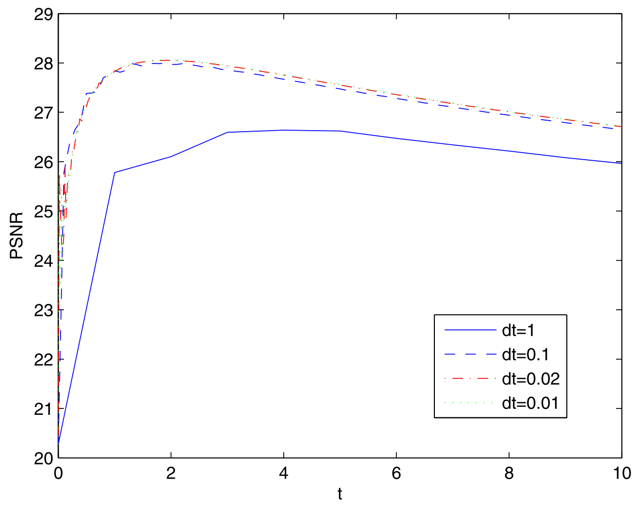


Fig. 4 The PSNR varies as the iteration time increases, the iteration time is $t = step \times dt$, *step* denotes the iteration step, *dt* denotes Δt. Without loss of generality, the test image is Lenna image added gaussian noise and the test model is D2Q5 model

where α is diffusion coefficient in (1) varying in $(0, 1]$. If $\Delta x = 1$ is set, then the stability condition for the above finite difference method is $0 < \Delta t \le 1/4$ [1]. Thus, lattice Boltzmann method allows larger time step to filter an image, see Fig. 3. In the experiments, the time steps Δt are both larger than $1/4$ which is the upper limit for the above finite difference method to maintain stability. The successful noise removal in Fig. 3(a1),(a2) shows that lattice Boltzmann method provides good stability in the image filtering.

Although larger Δt leads to less steps or less computation time, Δt should not be too large to the extent that makes ω approximate 0, where $\omega = 1/\tau$. Otherwise, the computation may result in relatively large numerical error [25]. Figure 4 depicts the PSNR variation of Lenna image added Gaussian noise. We can see the PSNR value for $\Delta t = 1$ is much lower than that of three other cases, the reason is for $\Delta t = 1$ the minimal ω ($\approx$0.33) approximates 0. We can also see the PSNR value for $\Delta t = 0.02$ is nearly the same as that of $\Delta t = 0.01$ but is slightly larger than that of $\Delta t = 0.1$. This relationship still exists in other tests, this implies that the magnitude of PSNR value does not change when Δt is smaller than 0.02.

3.2 Results and Discussions

In the following numerical experiments, different images added different types of noise are processed using D2Q9 and D2Q5 models with $\Delta t = 0.02$. All computation is done in the Inter(R) Core(TM) 2 CPU T5600 @ 1.83 GHz, C programs are optimized with the command-fast in the Inter(R) C++ Compiler 9.0.

As shown in Fig. 4, the PSNR value of the processed Lenna image firstly increase to the maximum and then decrease monotonically. All other tests show the same trend, so we take the maximal PSNR value of each processed image as the characteristic indicator to compute the increased PSNR value. Figure 5 shows the increased PSNR value of the processed image compared with the added noise image. We can see the PSNR difference between D2Q5 and D2Q9 model is not large. And the minimal, maximal and average deviation for two models are 0.01, 0.45 and 0.19. Moreover, we show the processed images in Matlab and find no visual difference between two models. Without loss of generality, the processed rice images are shown in Fig. 6. We can see the images processed with D2Q9 model are as clear as those processed with D2Q5 model. Thus, considering the PSNR value and visual effect, it is believed that two models have almost the same processing effect.

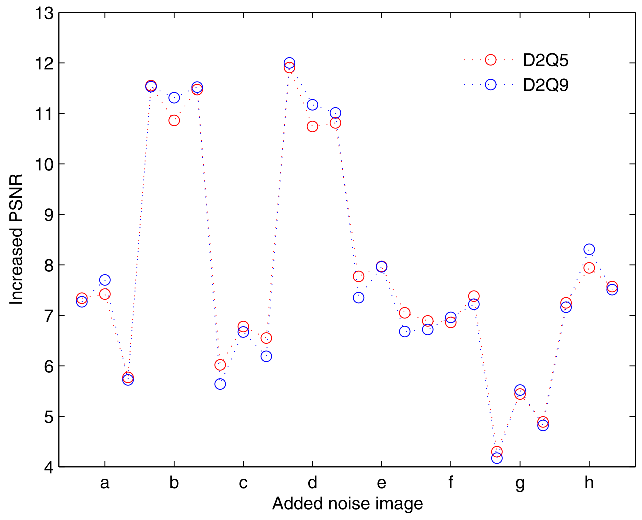


Fig. 5 The increased PSNR value of the processed image compared with the added noise image. The added noise image is the result of adding one type of noise to one image shown in Fig. 1. Corresponding to each image in Fig. 1, three points are draw, which represent images added gaussian noise, salt and pepper noise and speckle noise from left to right

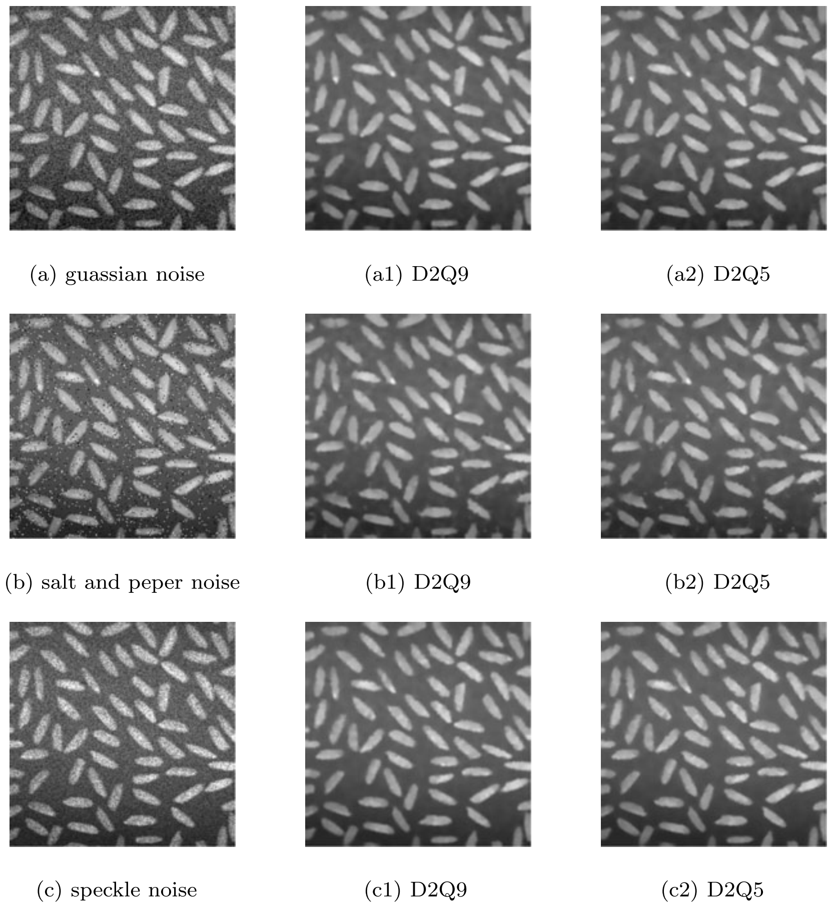


Fig. 6 (**a**), (**b**), (**c**) are rice images added gaussian noise, salt and pepper noise and speckle noise, respectively. (**a1**), (**b1**), (**c1**) are images processed with D2Q9 model corresponding to (**a**), (**b**), (**c**). (**a2**), (**b2**), (**c2**) are images processed with D2Q5 model corresponding to (**a**), (**b**), (**c**)

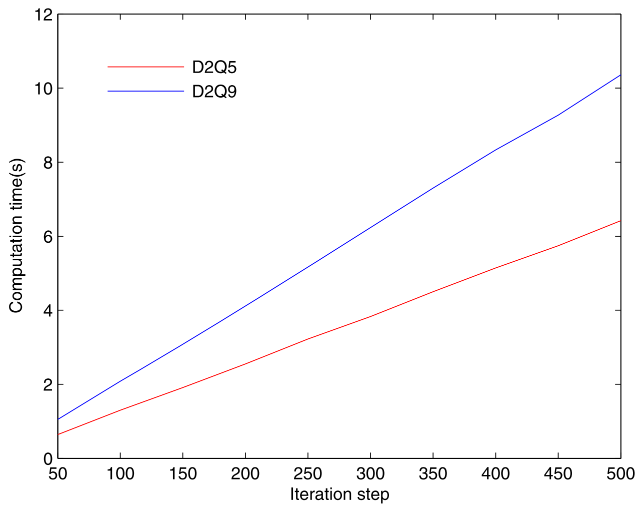


Fig. 7 The computation time corresponding to the iteration step for D2Q5 model and D2Q9 model

In addition, the first step to determine the maximal PSNR value of processed images is calculating P-M equation with sufficiently long time. In this process, D2Q5 model cost much less computation time than that of D2Q9 model, see Fig. 7. We can see the variation of computation time corresponding to the iteration step for two models is nearly linear. The difference between two models becomes larger and larger as the iteration step increases. The reason is that D2Q5 model only needs evolution of five directions but D2Q9 needs nine. Through computation, we know the slope of red line and blue line in Fig. 7 is 0.013 and 0.021 respectively. That is to say, D2Q9 model cost 1.62 times more time than that of D2Q5 model to iterate to the same step, so D2Q5 model is more efficient. If the resolution of image is very high, the benefit of D2Q5 model compared with D2Q9 model will be enlarged. So considering the filtering effect and efficiency, it is concluded that D2Q5 model is better than D2Q9 model.

In addition to above discussion, we can also obtain the following conclusions:

(1) Our lattice Boltzmann models which are used to solve P-M non-linear diffusion equation are effective in image denoising. Comparing the images added noise and the images filtered, it is found that noise of images has been removed effectively and the edge of images has been protected well. Comparing with the image added noise, the PSNR values of processed images increase at least by 4.17, at most by 12.00, average by 7.85. The PSNR values of some images are close to or more than 30 which generally means the image is clear enough.

(2) Our models are more effective in dealing with house and rice images than other six images, no matter what type of noise is added, see Fig. 5. PSNRs of processed house (b) and rice (d) images increase by more than 10, while PSNRs of other images increase average by 6.69. Moreover, structure of house and rice images is protected better than other six images. It is believed that our models can filter the image similar with house image or rice image very well.

(3) Salt and pepper noise is relatively difficult to remove compared with gaussian noise and speckle noise. Observing the processed versions of images which have been added salt and pepper noise, it is found that some noise points still exist in images. We can remove more noise points if the iteration steps are increased. However, if we do so, the cost will be that more details of images are lost and more contrast of images are reduced. The reason may be that P-M equation is insufficient to obtain satisfactory denoising results for images in which the brightness gradient generated by the noise is greater than that of the edges [1].

4 Conclusions

In this paper, we apply the lattice Boltzmann method to solve P-M equation for image filtering. D2Q5 and D2Q9 models are applied to filter eight images with three different types of noise. It is found that: (1) D2Q5 and D2Q9 models have almost the same filtering effect but D2Q5 model is more efficient (2) these models are more effective in dealing with house and rice images than other six images (3) salt and pepper noise is relatively difficult to remove compared with gaussian noise and speckle noise (4) lattice Boltzmann method shows good stability in the image filtering. It is worthy to note, that the higher efficiency of D2Q5 model will be very useful in dealing with large amount of image data. And above conclusions are important references when we choose lattice Boltzmann method to filter an image added different types of noise.

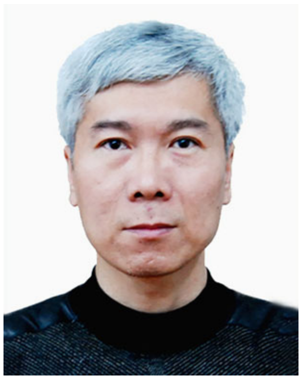
Acknowledgements One of authors (Wenhuan Zhang) would like to thank Prof. Songfa Xie, Guowei Wei for useful discussion and thank Prof. Zhuangzhi Yan for sharing his paper [18]. He also would like to thank Dr. Zhenhua Chai, Jianhua Lu, Fumei Rong and Master Futong Hu, Haipeng Wang for their help. The work is supported by National Natural Science Foundation of China (Grant Nos. 51006040, 51006039 and 60773195) and the Fundamental Research Funds for the Central Universities, HUST (Grant Nos. 2010MS131 and 2010QN057).

References

1. Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990)
2. Catte, F., Lions, P.L., Morel, J.M., Coll, T.: Image selective smoothing and edge detection by nonlinear diffusion. SIAM J. Numer. Anal. **29**, 182–193 (1992)
3. Alvarez, L., Lions, P.L., Morel, J.M.: Image selective smoothing and edge detection by nonlinear diffusion. SIAM J. Numer. Anal. **29**, 845–866 (1992)

4. You, Y.L., Kaveh, M.: Fourth-order partial differential equations for noise removal. IEEE Trans. Image Process. **9**, 1723–1730 (2000)
5. Chen, S., Dawson, S.P., Doolen, G.D., Janecky, D.R., Lawniczak, A.: Lattice methods and their applications to reacting systems. Comput. Chem. Eng. **19**(6–7), 617–646 (1995)
6. Zhang, C.Y., Tan, H.L., Liu, M.R., Kong, L.J.: A lattice Boltzmann model and simulation of KdV-Burgers equation. Commun. Theor. Phys. **42**(2), 281–284 (2004)
7. Ma, C.: A new lattice Boltzmann model for KdV-Burgers equation. Chin. Phys. Lett. **29**(9), 2313–2315 (2005)
8. Chai, Z.H., Shi, B.C., Zheng, L.: A unified lattice Boltzmann model for some nonlinear partial differential equations. Chaos Solitons Fractals **36**, 874–882 (2008)
9. Yan, G.: A lattice Boltzmann equation for waves. J. Comput. Phys. **161**, 61–69 (2000)
10. Hirabayashi, M., Chen, Y., Ohashi, H.: The lattice BGK model for the Poisson equation. JSME Int. J. Ser. B **44**(1), 45–52 (2001)
11. Chai, Z.H., Shi, B.C.: A novel lattice Boltzmann model for the Poisson equation. Appl. Math. Model. **32**, 2050–2058 (2008)
12. Guo, Z.L., Shi, B.C., Wang, N.C.: Fully Lagrangian and lattice Boltzmann methods for the advection-diffusion equation. J. Sci. Comput. **14**(3), 291–300 (1999)
13. Deng, B., Shi, B.C., Wang, G.C.: A new lattice Bhatnagar-Gross-Krook model for convection-diffusion equation with a source term. Chin. Phys. Lett. **22**(2), 267–270 (2005)
14. Shi, B.C., Guo, Z.L.: Lattice Boltzmann model for nonlinear convection-diffusion equations. Phys. Rev. E **79**, 016701 (2009)
15. Jawerth, B., Lin, P., Sinzinger, E.: Lattice Boltzmann models for anisotropic diffusion of images. J. Math. Imaging Vis. **11**, 231–237 (1999)
16. Zhao, Y.: Lattice Boltzmann based PDE solver on the GPU. Vis. Comput. **24**(5), 323–333 (2008)
17. Zhao, Y.: GPU-Accelerated surface denoising and morphing with lattice Boltzmann scheme. In: IEEE International Conference on Shape Modeling and Applications (SMI), pp. 19–28. (2008)
18. Chen, Y., Yan, Z.Z., Qian, Y.H.: An anisotropic diffusion model for medical image smoothing by using the lattice Boltzmann method. APCMBE 2008. IFMBE Proc. **19**, 255–259 (2008)
19. Chen, Y., Yan, Z.Z., Qian, Y.H.: The lattice Boltzmann method based image denoising. Acta Electron. Sinica (in Chinese) **37**(3), 574–580 (2009)
20. Chang, Q.S., Yang, T.: A lattice Boltzmann method for image denoising. IEEE Trans. Image Process. **18**(12), 2797–2802 (2009)
21. Qian, Y.H.: Fractional propagation and the elimination of staggered invariants in lattice-BGK models. Int. J. Mod. Phys. C **8**, 753–761 (1997)
22. Guo, Z.L., Zheng, C.G., Shi, B.C.: Non-equilibrium extrapolation method for velocity and pressure boundary conditions in the lattice Boltzmann method. Chin. Phys. **11**(4), 366–374 (2002)
23. Canny, J.: computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**, 679–698 (1986)
24. Ames, W.F.: Numerical Methods for Partial Differential Equations. Academic Press, New York (1977)
25. Wolf-Gladrow, D.: A lattice Boltzmann equation for diffusion. J. Stat. Phys. **79**(5/6), 1023–1032 (1995)

Wenhuan Zhang received his B.S. degree in Statistics in 2008 from Huazhong University of Science and Technology (HUST), China. From 2008 to 2009 he was a postgraduate student of Computational Mathematics at this University. Since 2009, he has been working on his Ph.D. degree in Thermal Power Engineering from HUST. His research interests include image filtering and image segmentation.



Baochang Shi is a Professor of Mathematics at the Huazhong University of Science and Technology (HUST) in PR China. He received his B.S. degree in Applied Mathematics, M.S. degree in Optimization and Ph.D. degree in Systems Engineering from the HUST. His research interests include image processing, fast transform and Lattice Boltzmann methods for fluid flow.