Commun. Comput. Phys.
doi: 10.4208/cicp.scpde14.30s

Vol. **19**, No. 5, pp. 1357-1374
May 2016

An Anisotropic Convection-Diffusion Model Using Tailored Finite Point Method for Image Denoising and Compression

Yu-Tuan Lin[1,2], Yin-Tzer Shih[1,*] and Chih-Ching Tsai[1]

[1] *Department of Applied Mathematics, National Chung Hsing University, Taichung 40227, Taiwan.*
[2] *Institute of Mathematics, Academia Sinica, Taiwan.*

Received 28 February 2015; Accepted (in revised version) 11 September 2015

Abstract. In this paper we consider an anisotropic convection-diffusion (ACD) filter for image denoising and compression simultaneously. The ACD filter is discretized by a tailored finite point method (TFPM), which can tailor some particular properties of the image in an irregular grid structure. A quadtree structure is implemented for the storage in multi-levels for the compression. We compare the performance of the proposed scheme with several well-known filters. The numerical results show that the proposed method is effective for removing a mixture of white Gaussian and salt-and-pepper noises.

AMS subject classifications: 65M50, 65M75, 68U10, 94A08

Key words: Perona-Malik filter, image denoising, convection-diffusion, tailored finite point method, singular perturbation problem, adaptive grid.

1 Introduction

Image denoising has been one of the most challenging issues since it is very difficult to preserve the edges and the desired textures while removing the noises. For years, many mathematical models for image denoising have been presented, such as the total variation (TV) model by Rudin, Osher and Fatemi [16] and the partial differential equation (PDE) based method pioneered by Perona and Malik [15]. In Perona-Malik (PM) type model, the image is selectively smoothed via the control of the diffusion coefficient depending on the gradient of the pixel intensity values. Thus, the edges as well as the

*Corresponding author. *Email addresses:* ytlin@math.sinica.edu.tw (Y.-T. Lin), yintzer_shih@nchu.edu.tw (Y.-T. Shih), jet.tsai98004@gmail.com (C.-C. Tsai)

 ©2016 Global-Science Press

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press

desired detailed information can be preserved as the noises are reduced through the evolution of the diffusion processing.

Later on, some nonlinear diffusion filters have been intensively studied in [4, 23]. Due to the elegance and the success of PM model, many nonlinear diffusion filters and algorithms [1, 5, 7, 13, 14, 22] have been proposed for edge-preserving smoothing. Shih *et al.* [20] proposed a convection-diffusion (CD) filter on adaptive grids, and the proposed convection term can quickly reduce the required time steps for reaching an admissible image quality. The CD filter works well especially on a mixture of Gaussian and pepper-and-salt noises in comparing with other PDE based filters.

The TFPM was first proposed by Prof. H. Han and then implemented by Han, Huang and Kellogg [8]. The TFPM performs very well in convection dominated convection-diffusion problems, and the essential concept of the TFPM is the selection of appropriate functions which are particular solutions to the differential equation. As a result, one can derive an approximating difference equation of which the coefficients are solved exactly from the selected functions, and the numerical solution is locally tailored to keep some particular properties of the equation. The TFPM can achieve approximation with higher accuracy even in coarse grid. Due to this special feature, many applications including convection-dominated convection-diffusion-reaction problems [19], first order wave equation [12] and steady magnetohydrodynamics duct flow problem [11] have been successfully implemented. More recently, TFPM demonstrates the effectiveness in solving singularly perturbed problems [9, 10]. Here we will present an anisotropic CD (ACD) filter with the advantage of the TFPM for image denoising.

The paper is organized as follows. Section 2 contains a brief introduction to various nonlinear diffusion filters including the proposed ACD filter for image denoising. In Section 3 we present the TFPM for ACD filter. In Section 4 we extend the TFPM on adaptive grids for the image compression. Section 5 presents the numerical results on several test images. Finally, we give some conclusions in Section 6.

2 Nonlinear filters

2.1 PM diffusion filters

First we briefly review the PM type nonlinear diffusion filter. For a given grayscale noisy image with pixel intensity $u_0(x,y)\colon \Omega \mapsto [0,255]$ the regularized PM type filter presented by Catté *et al.* [4] is

$$\begin{cases} u_t = \operatorname{div}(g(|\nabla(G_\sigma * u)|)\nabla u) & \text{in } \Omega\times(0,T), \\ \dfrac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega\times(0,T), \\ u(x,y,0) = u_0(x,y) & (x,y)\in\Omega, \end{cases} \tag{2.1}$$

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press

where Ω is a rectangular domain of the image with boundary $\partial\Omega$, n is the unit outward vector normal to $\partial\Omega$, div is the divergence operator, ∇ is the gradient operator, g is the diffusivity function, G_σ is the Gaussian function with the standard deviation σ, and $G_\sigma * u$ is the convolution of G_σ and u. The convolution guarantees the well-posedness of the equation. Generally, the diffusivity function g, or the edge-stopping function, is smooth and non-increasing. To suppress the diffusion on the edges, it satisfies $g(0)=1$, $g(s)>0$, $\forall s$, and $g(s)\to 0$ as $|s|\to\infty$. Two diffusivity functions proposed by Perona and Malik [15] are $g(s)=1/\left(1+s^2/K^2\right)$ and e^{-s^2/K^2} where K is the gradient threshold parameter. The PM type filter is isotropic since the diffusion coefficient is determined by a scalar-valued function.

In [23], Weickert proposed a filter based on an anisotropic diffusion tensor as

$$\begin{cases} u_t = \operatorname{div}(D\cdot\nabla u) & \text{in } \Omega\times(0,T), \\ \langle D\nabla u, n\rangle = 0 & \text{on } \partial\Omega\times(0,T), \\ u(x,y,0)=u_0(x,y) & (x,y)\in\Omega, \end{cases} \tag{2.2}$$

where D is a positive definite 2 by 2 matrix depending on the regularized structure tensor. D is written by

$$D = PBP^{-1},$$

where

$$P=\begin{bmatrix} v_1 & -v_2 \\ v_2 & v_1 \end{bmatrix}, \quad B=\begin{bmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{bmatrix},$$

and $[v_1\ v_2]^T$ and $[-v_2\ v_1]^T$ are the eigenvectors of D. Different choices on the selected eigenvalue and the corresponding eigenvector leads to a different diffusion tensor. For example, by letting $[v_1\ v_2]^T$ paralleled to ∇u_σ for u_σ the regularized image, one obtains an edge-enhancing diffusion model. The values of κ_1 and κ_2 suggested by Weickert [23] are given by

$$\kappa_1=\alpha, \text{ and } \kappa_2=\begin{cases} \alpha & \text{if } \mu_1=\mu_2, \\ \alpha+(1-\alpha)\exp\left(\frac{-C}{(\mu_1-\mu_2)^2}\right) & \text{if } \mu_1\neq\mu_2, \end{cases}$$

where $C>0$, $0<\alpha\ll 1$, $\mu_{1,2}$, $v_{1,2}$ are the eigenvalues and the corresponding eigenvectors of the regularized structure tensor $J=\nabla u_\sigma\cdot\nabla u_\sigma^T$, respectively.

2.2 CD filter and ACD filter

Shih *et al.* [20] modified the PM model and proposed a CD filter for image denoising as

$$\begin{cases} u_t-\varepsilon\Delta u+\beta\cdot\nabla u=0 & \text{in } \Omega\times(0,T), \\ \dfrac{\partial u}{\partial n}=0 & \text{on } \partial\Omega\times(0,T), \\ u(x,y,0)=u_0(x,y) & (x,y)\in\Omega, \end{cases} \tag{2.3}$$

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press

where the diffusion coefficient $\varepsilon=\frac{1}{1+|\nabla u|^2}$ and the velocity field

$$\beta=(\beta_1,\beta_2)=\gamma\frac{\nabla u^{\perp}}{|\nabla u|_{\epsilon}}. \tag{2.4}$$

Here γ is a constant that controls the magnitude of the convection term and $|\nabla u|_{\epsilon}=\sqrt{u_x^2+u_y^2+\epsilon}$ with $0<\epsilon\ll 1$. The direction of the convection term is orthogonal to the gradient and during the denoising process the image at edges would convect along the tangential direction of the edge. [20] considered an implicit time-stepping scheme with a modified streamline diffusion method (MSD) [18] to avoid the severe oscillation around the edges where ε is small.

Like PM type filter, the diffusion in CD filter is isotropic, which indicates that the capability for preserving details is not satisfactory. In this paper we consider an ACD filter

$$\frac{\partial u}{\partial t}-\mathrm{div}\left(\begin{bmatrix}\varepsilon_1 & 0\\ 0 & \varepsilon_2\end{bmatrix}\cdot\nabla u\right)+\beta\cdot\nabla u=0, \tag{2.5}$$

where ε_1 and ε_2 are diffusion coefficients in the $x-$ and $y-$ directions, respectively, and β is defined in Eq. (2.4). This filter is in fact a simple version of diffusion tensor and it can be implemented straightforward and easily by using the TFPM. For a given $u_{i,j}^k$ the approximations in u_x,u_y are

$$D_x u_{ij}^k=\begin{cases} D_x^+u_{i,j}^k & \text{if } \left|D_x^+u_{i,j}^k\right|\le\left|D_x^-u_{ij}^k\right|,\\ D_x^-u_{i,j}^k & \text{otherwise,}\end{cases} \tag{2.6}$$

$$D_y u_{ij}^k=\begin{cases} D_y^+u_{i,j}^k, & \text{if } \left|D_y^+u_{i,j}^k\right|\le\left|D_y^-u_{ij}^k\right|,\\ D_y^-u_{i,j}^k, & \text{otherwise,}\end{cases} \tag{2.7}$$

where

$$D_x^+u_{i,j}^k=(u_{i+1,j}^k-u_{i,j}^k)/h,\quad D_x^-u_{i,j}^k=(u_{i,j}^k-u_{i-1,j}^k)/h,$$
$$D_y^+u_{i,j}^k=(u_{i,j+1}^k-u_{i,j}^k)/h,\quad D_y^-u_{i,j}^k=(u_{i,j}^k-u_{i,j-1}^k)/h.$$

Similarly, the components of the convection vector β_1 and β_2 are given by

$$\beta_{1,ij}^k=\begin{cases}\beta_{1,ij}^{k\,+}, & \text{if } \left|\beta_{1,ij}^{k\,+}\right|\le\left|\beta_{1,ij}^{k\,-}\right|,\\ \beta_{1,ij}^{k\,-}, & \text{otherwise,}\end{cases}$$
$$\beta_{2,ij}^k=\begin{cases}\beta_{2,ij}^{k\,+}, & \text{if } \left|\beta_{2,ij}^{k\,+}\right|\le\left|\beta_{2,ij}^{k\,-}\right|,\\ \beta_{2,ij}^{k\,-}, & \text{otherwise,}\end{cases}$$

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press

where

$$\beta_{1,ij}^{k\,+}=\max\{\beta_{1,ij}^{k},0\},\quad \beta_{1,ij}^{k\,-}=\min\{\beta_{1,ij}^{k},0\},$$
$$\beta_{2,ij}^{k\,+}=\max\{\beta_{2,ij}^{k},0\},\quad \beta_{2,ij}^{k\,-}=\min\{\beta_{2,ij}^{k},0\}.$$

We consider a modified diffusivity function for improving the performance in removing the salt-and-pepper noise. First, set up a simple detector for salt-and-pepper noise. A pixel is a candidate of salt-and-pepper noise if it satisfies below conditions:

1. $u_{i,j}$ is extremely low or high,
2. $\sum_{m=i-1}^{i+1}\sum_{n=j-1}^{j+1}|u_{m,n}-u_{i,j}|$ is greater than a threshold value.

Note that in the first condition, we do not specify $u_{i,j}=0$ or 255 since this detector will be applied at each iteration during the ACD denoising process.

Next, we use the following diffusivity function to eliminate the salt-and-pepper noise candidates

$$\begin{cases}\varepsilon_1(p)=\left(\dfrac{Gx}{\mu}\right)^2,\ \varepsilon_2(p)=\left(\dfrac{Gy}{\mu}\right)^2 & \text{if } p \text{ is a salt-and-pepper noise candidate,}\\ \varepsilon_1(p)=\dfrac{1}{1+\left(\frac{Gx}{K}\right)^2},\ \varepsilon_2(p)=\dfrac{1}{1+\left(\frac{Gx}{K}\right)^2} & \text{otherwise,}\end{cases} \tag{2.8}$$

where $Gx\equiv D_x u_{i,j}^k$ and $Gy\equiv D_y u_{i,j}^k$ are defined in Eq. (2.6) and Eq. (2.7), μ and K are two threshold parameters for the edges. The salt-and-pepper noise candidates can be removed quickly since the diffusion coefficient is proportional to the value of $|Gx|$ or $|Gy|$.

3 Tailored finite point method for the ACD filter

3.1 The TFPM for the discretization

Let the solution to Eq. (2.5) have the form

$$u=e^{-A\lambda^2 t+\imath\lambda(Bx+Cy+Dt)}, \tag{3.1}$$

where $\imath=\sqrt{-1}$, $A=\varepsilon_1 B^2+\varepsilon_2 C^2$, and $D=\beta_1 B+\beta_2 C$ with B, C, λ being arbitrary. Note that for $\lambda=0$, the solution is a constant. Now we develop an explicit difference scheme on a subdomain with six points as shown in Fig. 1. Consider only two time steps, $t=0$ and $t=\tau$. For time $t=0$, there are five points, the reference point $p_{0,0}=(0,0,0)$ together with four neighboring points in a counterclockwise order: $p_{1,t}=(h_{x_1},h_{y_1},0)$, $p_{2,0}=(h_{x_2},h_{y_2},0)$,

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press

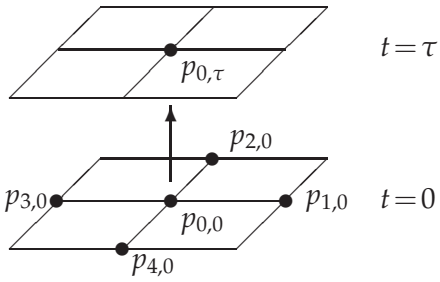


Figure 1: The reference points for ACD-TFPM.

$p_{3,0}=(h_{x_3},h_{y_3},0)$ and $p_{4,0}=(h_{x_4},h_{y_4},0)$, where h_{x_i} and h_{y_i} are the coordinates for the point $p_{i,0}$ in x- and y-directions. We want to find the approximate solution at time $t=\tau$, and $p_{0,\tau}=(0,0,\tau)$.

Next, we choose the solution space consisting of five basis functions. There can be a variety of options. For example, by taking $(B,C)=(\pm1,0)$ and $(0,\pm1)$ we have

$$\begin{aligned}\mathbf{W}_1^5=\{&1,\ e^{-\varepsilon_1\lambda^2 t}\cos(\lambda(x+\beta_1 t)),\ e^{-\varepsilon_1\lambda^2 t}\sin(\lambda(x+\beta_1 t)),\\ &e^{-\varepsilon_2\lambda^2 t}\cos(\lambda(y+\beta_2 t)),\ e^{-\varepsilon_2\lambda^2 t}\sin(\lambda(y+\beta_2 t))\}.\end{aligned}\tag{3.2}$$

Denoting $u_{i,j}^n$ by the numerical approximation of $u(x_i,y_j,t_n)$, we assume the explicit difference scheme is given by

$$u_{i,j}^{n+1}=\alpha_0 u_{i,j}^n+\alpha_1 u_{i+1,j}^n+\alpha_2 u_{i,j+1}^n+\alpha_3 u_{i-1,j}^n+\alpha_4 u_{i,j-1}^n.\tag{3.3}$$

Substituting all the basis functions of Eq. (3.2) into Eq. (3.3) at each points on a subdomain, the coefficients α_i can be determined by the local system

$$\mathbf{A}\boldsymbol{\alpha}=\mathbf{b},\tag{3.4}$$

where

$$\mathbf{A}=\begin{bmatrix}1 & 1 & 1 & 1 & 1\\ 1 & \cos(\lambda h_{x_1}) & \cos(\lambda h_{x_2}) & \cos(\lambda h_{x_3}) & \cos(\lambda h_{x_4})\\ 0 & \sin(\lambda h_{x_1}) & \sin(\lambda h_{x_2}) & \sin(\lambda h_{x_3}) & \sin(\lambda h_{x_4})\\ 1 & \cos(\lambda h_{y_1}) & \cos(\lambda h_{y_2}) & \cos(\lambda h_{y_3}) & \cos(\lambda h_{y_4})\\ 0 & \sin(\lambda h_{y_1}) & \sin(\lambda h_{y_2}) & \sin(\lambda h_{y_3}) & \sin(\lambda h_{y_4})\end{bmatrix},$$

$$\boldsymbol{\alpha}=[\alpha_0,\ \alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4]^T,$$

and

$$\begin{aligned}\mathbf{b}=\{&1,\ e^{-\varepsilon_1\lambda^2\tau}\cos(\lambda\beta_1\tau),\ e^{-\varepsilon_1\lambda^2\tau}\sin(\lambda\beta_1\tau),\\ &e^{-\varepsilon_2\lambda^2\tau}\cos(\lambda\beta_2\tau),e^{-\varepsilon_2\lambda^2\tau}\sin(\lambda\beta_2\tau)\}.\end{aligned}$$

For any λ such that $\det(\mathbf{A})\neq0$, $\boldsymbol{\alpha}$ can be uniquely defined. We can solve the whole system point-wisely.

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press

3.2 Stability analysis

For the simplicity, assume that there are uniformly partitions both on the spatial domain and the time domain with grid size h and time step τ, respectively. By letting $\varepsilon_1=\varepsilon_2=\varepsilon$ and $\beta_1=\beta_2=\bar{\beta}$, one can verify that the coefficients in Eq. (3.3) are given by

$$\begin{aligned}
\alpha_0 &= -\frac{1+\cos(\lambda h)}{1-\cos(\lambda h)}+\frac{2\cos\left(\bar{\beta}\lambda\tau\right)}{1-\cos(\lambda h)}e^{-\varepsilon\lambda^2\tau},\\
\alpha_1=\alpha_2 &= \frac{1}{2(1-\cos(\lambda h))}-\frac{\sin\left(\bar{\beta}\lambda\tau\right)+\sin\left(\lambda\left(h-\bar{\beta}\tau\right)\right)}{2\sin(\lambda h)\left(1-\cos(\lambda h)\right)}e^{-\varepsilon\lambda^2\tau},\\
\alpha_3=\alpha_4 &= \frac{1}{2(1-\cos(\lambda h))}-\frac{-\sin\left(\bar{\beta}\lambda\tau\right)+\sin\left(\lambda\left(h+\bar{\beta}\tau\right)\right)}{2\sin(\lambda h)\left(1-\cos(\lambda h)\right)}e^{-\varepsilon\lambda^2\tau},
\end{aligned} \tag{3.5}$$

where $\lambda\neq 0$. Following Von Neumann linear stability analysis, we write the numerical solution at time $t=n\tau$ as

$$u_{i,j}^n=\zeta^n e^{\imath\gamma ih}e^{\imath\kappa jh}, \tag{3.6}$$

where γ and κ are spatial frequencies, and ζ is the amplification factor. Substituting this expression into Eq. (3.3), we have

$$\begin{aligned}
\zeta^{n+1}e^{\imath\gamma ih}e^{\imath\kappa jh} =& \alpha_0\zeta^n e^{\imath\gamma ih}e^{\imath\kappa jh}+\alpha_1\zeta^n e^{\imath\gamma(i+1)h}e^{\imath\kappa jh}+\alpha_2\zeta^n e^{\imath\gamma ih}e^{\imath\kappa(j+1)h}\\
&+\alpha_3\zeta^n e^{\imath\gamma(i-1)h}e^{\imath\kappa jh}+\alpha_4\zeta^n e^{\imath\gamma ih}e^{\imath\kappa(j-1)h}.
\end{aligned} \tag{3.7}$$

Dividing Eq. (3.7) by $\zeta^n e^{\imath\gamma ih}e^{\imath\kappa jh}$ gives

$$\zeta=\alpha_0+\alpha_1 e^{\imath\gamma h}+\alpha_2 e^{\imath\kappa h}+\alpha_3 e^{-\imath\gamma h}+\alpha_4 e^{-\imath\kappa h}. \tag{3.8}$$

From Eq. (3.5), $\alpha_3=\alpha_1+\frac{\sin\left(\bar{\beta}\lambda\tau\right)}{\sin(h\lambda)}e^{-\varepsilon\lambda^2\tau}$ and $\alpha_4=\alpha_2+\frac{\sin\left(\bar{\beta}\lambda\tau\right)}{\sin(h\lambda)}e^{-\varepsilon\lambda^2\tau}$, we arrive at

$$\begin{aligned}
|\zeta| &= \left|\alpha_0+2\alpha_1\left(\cos(\gamma h)+\cos(\kappa h)\right)+\frac{\sin\left(\bar{\beta}\lambda\tau\right)e^{-\varepsilon\lambda^2\tau}}{\sin(h\lambda)}\left(e^{-\imath\gamma h}+e^{-\imath\kappa h}\right)\right|\\
&\leq |\alpha_0|+4|\alpha_1|+2\left|\frac{\sin\left(\bar{\beta}\lambda\tau\right)e^{-\varepsilon\lambda^2\tau}}{\sin(h\lambda)}\right|.
\end{aligned} \tag{3.9}$$

Note that for $\bar{\beta}=0$, the imaginary parts in Eq. (3.9) vanishes and

$$\begin{aligned}
\zeta &= -1+2e^{-\frac{\varepsilon\pi^2\tau}{4h^2}}+\cos(\gamma h)+\cos(\kappa h)-e^{-\frac{\varepsilon\pi^2\tau}{4h^2}}\cos(\kappa h)-\mathrm{e}^{-\frac{\varepsilon\pi^2\tau}{4h^2}}\cos(\gamma h)\\
&= 1-2\left[\sin^2\left(\frac{\gamma h}{2}\right)+\sin^2\left(\frac{\kappa h}{2}\right)\right]\left(\frac{1-e^{-\varepsilon\lambda^2\tau}}{1-\cos(\lambda h)}\right).
\end{aligned} \tag{3.10}$$

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press

Obviously, $\left[\sin^2\left(\frac{\gamma h}{2}\right)+\sin^2\left(\frac{\kappa h}{2}\right)\right]$ and $\frac{1-e^{-\varepsilon\lambda^2\tau}}{1-\cos(\lambda h)}$ are nonnegative. $|\zeta|<1$ implies

$$\left[\sin^2\left(\frac{\gamma h}{2}\right)+\sin^2\left(\frac{\kappa h}{2}\right)\right]\left(\frac{1-e^{-\varepsilon\lambda^2\tau}}{1-\cos(\lambda h)}\right)<1,$$

which leads to

$$\frac{1-e^{-\varepsilon\lambda^2\tau}}{1-\cos(\lambda h)}<\frac{1}{2}.$$

Therefore, the stability condition is dependent on λ, h and ε, and the restriction on the selected time step is

$$\tau<\frac{\ln 2-\ln(1+\cos(\lambda h))}{\varepsilon\lambda^2}. \tag{3.11}$$

For $1+\cos(\lambda h)\neq 0$, the ACD-TFPM in uniform partitions is conditionally stable. In particular, for $\lambda\to\frac{(1+2m)\pi}{h}$, $m=0,1,2,\cdots$, it tends to be unconditionally stable. For the proper choice by letting $\lambda=\lfloor\frac{\pi}{h}\rfloor$, the time step τ can be a larger number.

4 Implement the TFPM in adaptive grids

4.1 The quadtree representation of images

In the CD filter, the adaptive meshing is based on a strategy that utilizes the saturation condition to coarsen the mesh. With a coarsening indicator, the multi-level grid is constructed and only one level difference for any two neighboring elements in mesh hierarchies is allowed. This kind of meshing is appropriate for Galerkin finite element method. In this work we use a popular refining approach, the quadtree representation for images, in which the cost of mesh generation is lower and the level difference can be two or more.

Finkel and Bentleyis [6] proposed the quadtree data structure where every internal node has four leafs. For image representation, a quadtree is constructed by recursively partitioning the image into four sub-blocks. The leafs of a root is in order of the NW, NE, SW, and SE quadrants as shown in Fig. 2. Only one intensity value represent all the values in the sub-block. Naturally, this technique is immediately used for image compression if various resolution levels are indispensable for keeping a certain degree of quality. For a complex region in an image, further segmentation is needed. The evaluation of the criterion for segmentation plays a key role in the quadtree representation. Since each sub-block is represented by a single value, we have to assure whether a further division into four sub-blocks is needed. One common measurement is the difference between the maximum and the minimum intensity values in a sub-block. If the difference is greater than a threshold, the sub-block must be subdivided; otherwise, the division procedure stops.

For better preservation of edges and details, we incorporate Canny edge detection [3] into the quadtree representation. For any sub-block that contains at least one pixel

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press

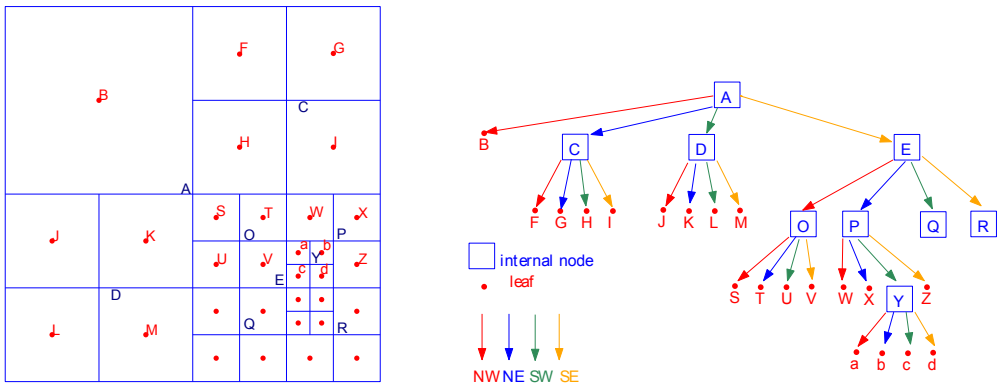


Figure 2: An example of quadtree structure for image representation.

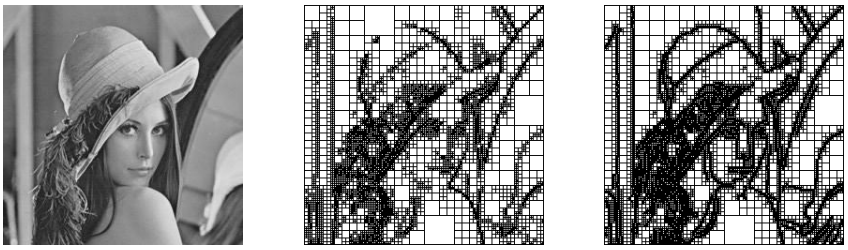


Figure 3: Quadtree structure for 256×256 Lena image. Left: original image; Middle: quadtree structure without using Canny edge detection; Right: quadtree structure after using Canny edge detection.

marked as an edge region, the sub-block is divided into four sub-blocks. Fig. 3 compares the block structures with and without Canny edge detection. The maximum block size is 64×64. Clearly, more details are retained using the Canny edge detection.

The image processing toolbox of Matlab provides an efficient solution for constructing the quadtree decomposition iteratively. Instead of using a tree structure, a sparse matrix **S** stores a quadtree structure as shown in Table 1. We will use this representation for implementing the TFPM in adaptive grids.

Table 1: The sparse matrix **S** with element S_{ij} representing the dimension of block.

upper left corner (S)	dim of block
(1,1)	16
(17,1)	8
(17,9)	8
⋮	⋮
(17,17)	4
⋮	⋮

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press

4.2 The TFPM in adaptive grids

The TFPM possesses the properties of meshless method as the grid connectivity is unnecessary for computation, and it can be easily extended on scattered nodes without many modifications. Here we introduce the TFPM on the multi-level grid points obtained by quadtree decomposition.

Overall, the methodology is similar to the TFPM on uniform partitions except that the determination of the subdomain requires more attention owing to the complexity of the multi-level grid placement. The mean value of the intensities for each sub-block represents all the intensities for the sub-block. The $k-$nearest points searching algorithm, which is commonly used for quadtree search, is not suitable in this case. For instance, the four selected points may be too close in a small region, resulting in a badly ill-conditioned local matrix. For the worst case, these points can be co-linear and thus the local matrix is singular. Accordingly, we seek each nearest point in each quadrant, which guarantees the points in a subdomain are well-distributed for a better computational stability. Fig. 4 shows one example of the subdomain at time step $k-1$.

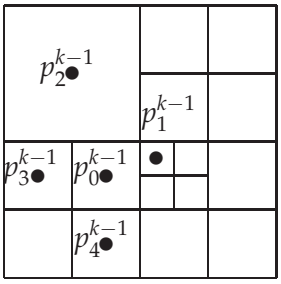


Figure 4: The reference points for TFPM on adaptive meshes at t^{k-1}.

Even though the points in a computational subdomain are selected using the above method, the local matrix can be singular or ill-conditioned under certain circumstances if the above-mentioned basis is applied. We consider two approaches to solve the problem. If the local matrix is singular, we use an alternative basis

$$\mathbf{W}_2^5 = \left\{1,\, e^{-\varepsilon_1\lambda^2 t}\cos\left(\lambda\left(\frac{x+y}{\sqrt{2}}+\beta_1 t\right)\right),\, e^{-\varepsilon_1\lambda^2 t}\sin\left(\lambda\left(\frac{x+y}{\sqrt{2}}+\beta_1 t\right)\right),\right.$$
$$\left. e^{-\varepsilon_2\lambda^2 t}\cos\left(\lambda\left(\frac{x-y}{\sqrt{2}}+\beta_2 t\right)\right),\, e^{-\varepsilon_2\lambda^2 t}\sin\left(\lambda\left(\frac{x-y}{\sqrt{2}}+\beta_2 t\right)\right)\right\}, \tag{4.1}$$

which is obtained by taking $(B,C)=(\pm\frac{1}{\sqrt{2}},\pm\frac{1}{\sqrt{2}})$. This basis is in fact a result of rotating the first basis by an angle of $\frac{\pi}{4}$.

If both of the two bases fail, we exploit the traditional Tikhonov regularization [21], where one can achieve an acceptable approximate solution $\tilde{\mathbf{x}}$ to the linear system $\mathbf{Ax}=\mathbf{b}$

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press

by solving

$$\tilde{\mathbf{x}} = (\mathbf{A}^*\mathbf{A} + \delta\mathbf{I})^{-1}\mathbf{b}, \tag{4.2}$$

where $\mathbf{A}^*$ is the conjugate transpose of matrix $\mathbf{A}$, $\mathbf{I}$ is the identity matrix, and δ is a positive regularization parameter.

5 Numerical experiments

In this section we compare the performances of the ACD-TFPM with some well-known filters for removing a mixture of Gaussian noise and the salt-and-pepper noise. The test images are grayscale pictures. We test 5 different filters including the PM implemented by finite difference method (FD), the adaptive parameterized block-based singular value decomposition (APBSVD) [17] with maximum block size 16, adaptive wavelet thresholding (DWT) [2] and median filter (MF) using 3×3 windows.

For the PM and the ACD, we set $K=\mu=20$ for the diffusivity function in Eq. (2.8). For the ACD model, we let $\gamma=0.01$ in the convection term, and set $\tau=$5E-2 in uniform grids and $\tau=$5E-4 in adaptive grids. Let the parameter $\delta=$1E-8 for Tikhonov regularization. All the computation are performed on an Intel Core PC using MATLAB Version 2014a with double precision arithmetic on processor I7-4790 with 3.60GHz and 32 GB RAM.

For an image of size $M\times N$, the evaluation of image quality is measured by peak signal to noise ratio (PSNR) and mean absolute error (MAE) defined as

$$\begin{aligned} \text{PSNR} &= 10\log_{10}\left\{\frac{255^2 MN}{\sum_{i=1}^{M}\sum_{j=1}^{N}[u_0(i,j)-u(i,j)]^2}\right\}, \\ \text{MAE} &= \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}|u_0(i,j)-u(i,j)|}{MN}, \end{aligned} \tag{5.1}$$

where u_0 is the intensity of the original image and u is that of the recovered image.

Experiment 5.1. Compare ACD/TFPM filter with other filters

We compare the performances of the ACD-TFPM filter with those of the APBSVD, DWT, PM/FD, and MF. It is well-known that the APBSVD, DWT, and PM/FD are excellent filters for reducing Gaussian noises while MF is also good for removing the salt-and-pepper noise. All test images of size 512×512 are corrupted by Gaussian noises with standard deviation $\sigma=10$ and 5% salt-and-pepper noise. On the uniform partitions with $h=1$, we set $\lambda=3$ based on the stability analysis in Section 4. The threshold of salt-and-pepper noise detector is the 80-th percentile of the total image gradients.

The performance of each filter is displayed in Table 2. With the presence of salt-and-pepper noise, the APBSVD, DWT and PM/FD filters are all impractical and the performance of the APBSVD is the worst. MF is much better in this case. Even MF does not work for Gaussian noises, the salt-and-pepper noise at this density level can nearly be

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press

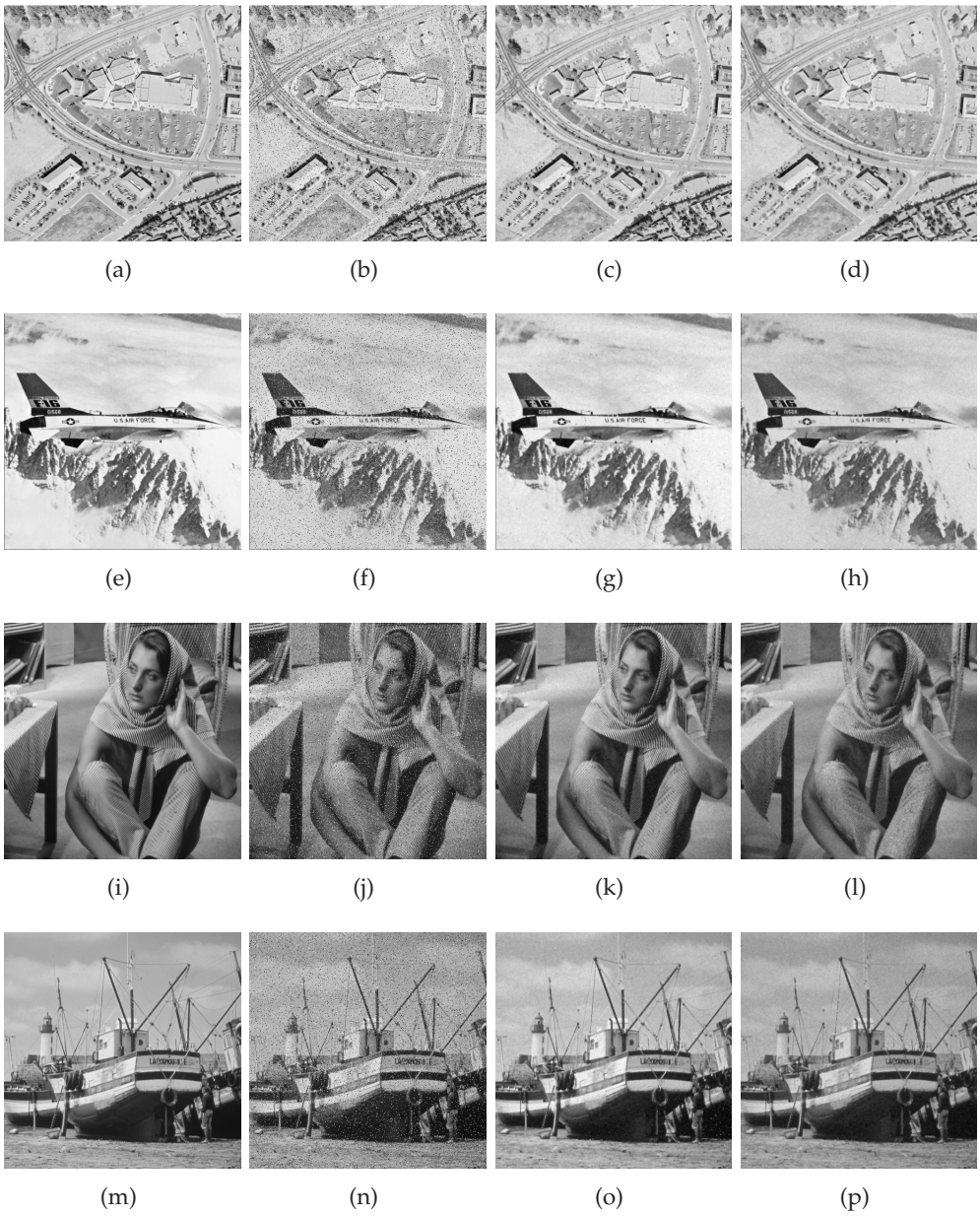


Figure 5: In the first column (a), (e), (i) and (m) are the original images; in the second column (b), (f), (j) and (n) are corrupted images; in the third column (c), (g), (k) and (o) are the recovered images by using ACD-TFPM; in the last column (d), (h), (l) and (p) are the recovered images by using MF.

diminished completely. On the other hand, the ACD-TFPM can deal with both kinds of noises and give the best recovered images. Fig. 5 shows the original images, corrupted images, and images restored by the ACD-TFPM and MF. We can see that most of the

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press

Table 2: Images corrupted by white Gaussian noise ($\sigma=10$) and salt-and-pepper noise (5%).

	Aerial		Airplane		Barbara		Boat	
	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR
Noise	13.91	17.65	13.99	17.53	13.89	18.04	13.96	17.91
APBSVD	13.90	17.76	13.15	17.69	13.61	18.21	13.51	18.08
PM/FD	12.82	17.62	13.29	17.59	13.11	18.11	12.91	18.00
DWT	12.09	19.99	11.32	19.92	11.57	20.91	11.28	20.87
MF	7.38	26.98	4.93	30.94	9.04	24.54	5.76	29.77
ACD-TFPM	**5.66**	**29.47**	**3.94**	**32.77**	**5.53**	**30.10**	**4.56**	**31.78**

salt-and-pepper noises are eliminated by MF; however, the recovered images are darker than original images. The recovered images by the ACD-TFPM contain more detailed information, and they are brighter than those of MF.

Experiment 5.2. Preserve image details: pinstriped texture

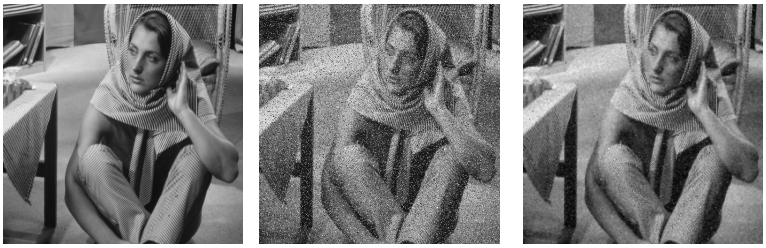
Since the APBSVD, DWT and PM/FD are only effective for Gaussian noise, we consider a two-stage denoising approach, which we will compare to the ACD-TFPM. We first remove the salt-and-pepper by MF and then reduce the Gaussian noises. We use 512×512 Barbara image to investigate the effects of MF+APBSVD, MF+PM/FD, MF+DWT and our ACD-TFPM on some fine textures. The noisy levels are higher than the first experiment with $\sigma=20$ for Gaussian noise and 10% for salt-and-pepper noise. Table 3 reveals the results. PSNR for the corrupted image is 14.64 and MAE is 27.08. The performance of ACD-TFPM remains the best and can reach as high as 25.88. For recovered images by MF+APBSVD, MF+PM/FD and MF+DWT, most of the salt-and-pepper noises disappear and the Gaussian noises are also reduced. Using this strategy, PSNRs are around 23.

Table 3: Barbara image corrupted by white Gaussian noise ($\sigma=20$) and salt-and-pepper noise (10%).

	Noise	MF+APBSVD	MF+PM/FD	MF+DWT	ACD-TFPM
PSNR	14.64	23.63	23.37	23.43	**25.88**
MAE	27.08	10.96	11.64	11.63	**9.01**

Fig. 6 shows the recovered images and Fig. 7 highlights the textures of the trouser in the marked region. We can see that the pinstriped texture can still be recognized in the image denoised by ACD-TFPM, whereas the textures in other restored images are thoroughly destroyed. After removing salt-and-pepper noises by MF, from our observation the delicate texture of the restored image has been damage and affects the later removal of Gaussian noises. ACD-TFPM prevails because the salt-and-pepper and Gaussian noises are treated simultaneously. The comparison of image qualities confirms the effectiveness of our proposed method.

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press

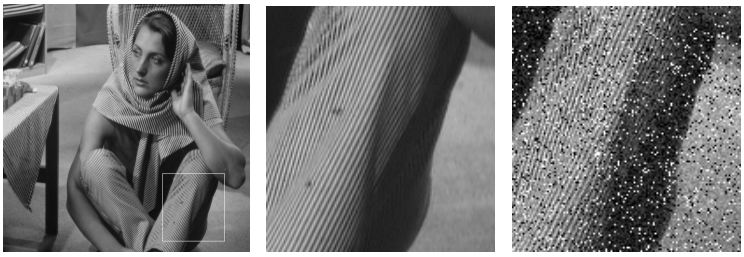


(a) Original image. (b) Noisy image. (c) ACD-TFPM.

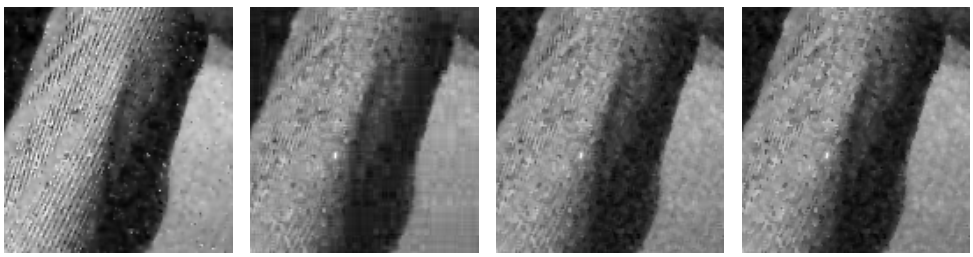


(d) MF+APBSVD (e) MF+DWT (f) MF+PM/FD

Figure 6: Experiment 5.2. The Barbara image is corrupted by $\sigma = 20$ Gaussian noise and 10% salt-and-pepper noise.



(a) Original image. (b) Zoom-in of the original image. (c) Zoom-in of the noisy image.



(d) ACD-TFPM. (e) MF+APBSVD (f) MF+DWT (g) MF+PM/FD

Figure 7: Experiment 5.2. A comparison of the pinstriped textures of the trouser by various filters.

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press

Experiment 5.3. Performance using adaptive grids

In this experiment we demonstrate the performance of the ACD-TFPM on five-level adaptive grids with sub-block sizes from 1×1 to 16×16 established by using the quadtree decomposition. The test images are of size 256×256 and corrupted by Gaussian noises with standard deviation σ=10 and 5% salt-and-pepper noise. The thresholds for quadtree decomposition, Canny edge detection and the salt-and-pepper noise detector are 32, 30 and 60, respectively.

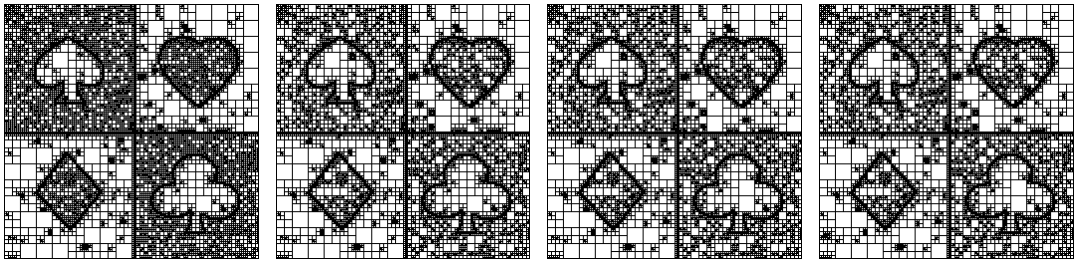
Table 4 shows the required sizes of the images, compression ratios (CR), PSNRs and MAEs at the first 8 iterations for the three test images, Poker, Lena and Peppers. All the required nodes decrease significantly at the second iteration, and then fluctuate moderately for the rest of iterations. For Poker image, CR is less than 20% since this image is much simpler. The PSNR at the second iteration reaches the maximum value, and then goes down gradually. For other real images, CRs are roughly 50%–55%. The PSNR numbers increase slightly and are almost the same.

Table 4: Block structures for 256×256 images corrupted by Gaussian noise with $\sigma=10$ and salt-and-pepper noises (5%). Maximum block size is 8×8. NO denotes the number of required nodes.

	Ite.	1	2	3	4	5	6	7	8
Poker	NO	14353	12793	12784	12784	12733	12754	12688	12439
	CR	21.90%	19.52%	19.51%	19.51%	19.43%	19.46%	19.36%	18.98%
	PSNR	26.82	26.94	26.94	26.93	26.92	26.90	26.88	26.87
	MAE	5.44	5.11	5.14	5.16	5.19	5.21	5.24	5.23
Lena	NO	34576	32485	32542	32527	32485	32557	32404	32233
	CR	52.56%	49.57%	49.66%	49.63%	49.56%	49.68%	49.44%	49.18%
	PSNR	28.23	28.37	28.39	28.40	28.41	28.42	28.44	28.46
	MAE	6.76	6.53	6.53	6.53	6.54	6.54	6.53	6.51
Peppers	NO	35905	34498	34387	34387	34375	34306	34222	34105
	CR	54.79%	52.64%	52.47%	52.47%	52.45%	52.34%	52.21%	52.11%
	PSNR	27.87	27.96	27.97	27.98	28.00	28.01	28.02	28.02
	MAE	7.06	6.92	6.91	6.92	6.92	6.92	6.92	6.92

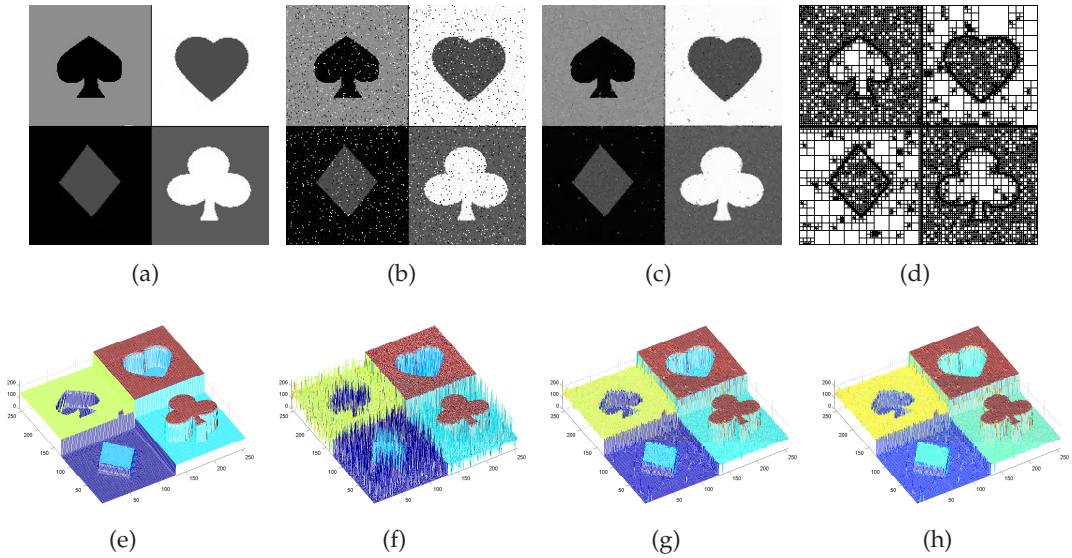
Fig. 8 illustrates the quadtree block structures of Poker at the time steps $t=t_1$, t_3, t_5 and t_7. Most of the block sizes at iteration 1 are smaller since the noises are not suppressed at this step, and after iteration 3 the sub-blocks are bigger due to the reduction of noise. Fig. 9 shows the 3-D plots of the original, corrupted images of Poker and the recovered images at the time steps t_1 and t_2. Fig. 10 shows the recovered images and block structures of Lena and Peppers. Similar to the previous two experiments, there is always a trade-off between removing these two kinds of noises. If we want to reduce more salt-and-pepper noises, the threshold for salt-and-pepper noise detector must be lower, which causes a poorer performance for removing Gaussian noises and vice versa. The selections of threshold parameters need a further study for optimal performances.

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press



(a) $t=t_1$ (b) $t=t_3$ (c) $t=t_5$ (d) $t=t_7$

Figure 8: Experiment 5.3. Block structures for Poker denoising at $t=t_1$, t_3, t_5, and t_7.



(a) (b) (c) (d)

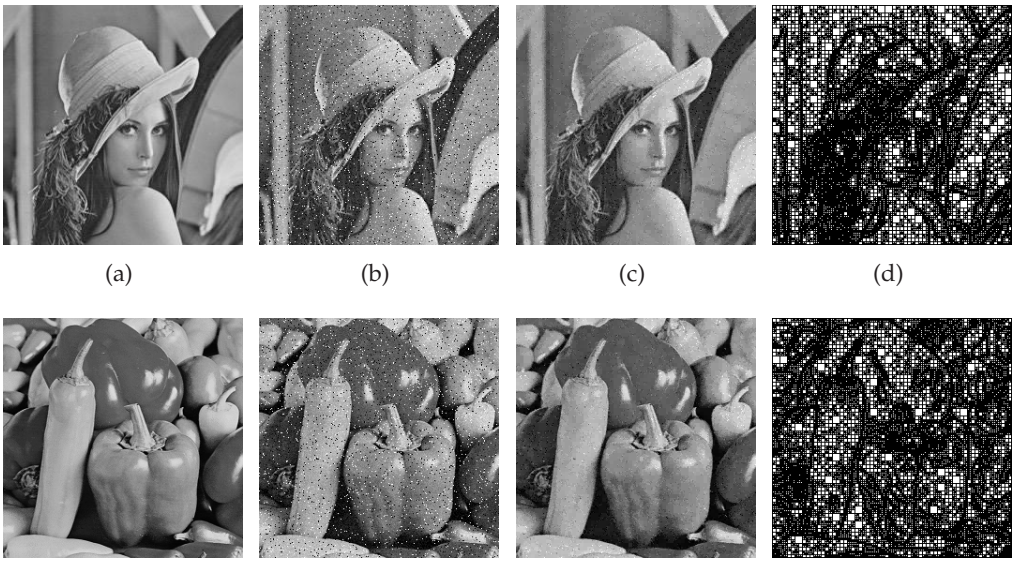
(e) (f) (g) (h)

Figure 9: Experiment 5.3. (a) the original image, (b) the noisy image with $\sigma=10$ Gaussian noise and 5% salt-and-pepper noise, (c) the recovered image by ACD-TFPM, (d) the block structure at $t=t_2$, (e) the 3-D plot of (a), (f) is the 3-D plots of (b), (g) and (h) are 3-D plots for $t=t_1$ and t_2, respectively.

6 Conclusions

We propose an ACD model for image denoising using the TFPM. In addition, we implement our model with four-level grid structure generated by quadtree decomposition to simultaneously perform the denoising and compression. Our numerical experiments show that for a mixture of white Gaussian and salt-and-pepper noises our model obtains higher quality of the recovered images and better capability of detail-preserving in comparison with other well-known filters. In the future study, we will consider the selection of the threshold parameters and other algorithms in adaptive grids to reach higher computational efficiency.

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 10: Experiment 5.3. (a) and (e) are original images, (b) and (f) are noisy images with $\sigma=10$ Gaussian noise and 5% salt-and-pepper noise, (c) and (g) are recovered images by ACD-TFPM, (d) and (h) are the block structures.

Acknowledgments

The authors would like to thank S.-Y. Chiou and Y.-R. Lin for programming parts of the Matlab functions used in this work. This work was supported by the Ministry of Science and Technology of Taiwan under the grants MOST 103-2115-M-005-004-MY2 and MOST 103-2811-M-005-032.

References

[1] D. Barash and D. Comaniciu, A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift, Image Vis. Comput., 22 (2004), 73-81.
[2] S. Chang, Y. Bin and M. Vetterli, Adaptive wavelet thresholding for image denoising and compression, IEEE Trans. Image Process., 9 (2000), 1532-1546.
[3] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Machine Intell., 8 (1986), 679-698.
[4] F. Catté, P. L. Lions, J. M. Morel and T. Coll, Image selective smoothing and edge detection by nonlinear diffusion, SIAM J. Numer. Anal., 29 (1992), 182-193.
[5] S. Chao and D. Tsai, An improved anisotropic diffusion model for detail and edge preserving smoothing, Pattern Recognit. Lett., 31 (2010), 2012-2023.
[6] R. Finkel and J. L. Bentley, Quad trees: a data structure for retrieval on composite keys, Acta Inform., 4 (1974), 1-9.
[7] Z. Guo, J. Sun, D. Zhang and B. Wu, Adaptive Perona-Malik model based on the variable exponent for Image denoising, IEEE Trans. Image Process., 21 (2012), 958-967.

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press

[8] H. Han, Z. Huang, and R. B. Kellogg, A tailored finite point method for a singular perturbation problem on an unbounded domain, J. Sci. Comput., 36 (2008), 243-261.
[9] H. Han, J.J.H. Miller and M. Tang, A parameter-uniform tailored finite point method for singularly perturbed linear ODE systems, J. Comp. Math., 31 (2013), 422-438.
[10] H. Han, Y. T. Shih and C. C. Tsai, Tailored finite point method for numerical solutions of singular perturbed eigenvalue problems, Adv. Appl. Math. Mech., 6 (2014), 376-402.
[11] P. W. Hsieh, Y. T. Shih and S. Y. Yang, A tailored finite point method for solving steady MHD duct flow problems with boundary layers, Commun. Comput. Phys., 10 (2011), 161-182.
[12] Z. Huang and X. Yang, Tailored finite point method for first order wave equation, J. Sci. Comput., 49 (2011), 351-366.
[13] F. Liu and J. Liu, Anisotropic diffusion for image denoising based on diffusion tensors, J. Vis. Commun. Image R., 23 (2012), 516-521.
[14] C. Lopez-Molina, M. Galar, H. Bustince and B. De Baets, On the impact of anisotropic diffusion on edge detection, Pattern Recognit., 47 (2014), 270-281.
[15] P. Perona and J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Trans. Pattern Anal. Machine Intell., 12 (1990), 629-639.
[16] L. I. Rudin, S. Osher, and E. Fatemi, Nonlinear total variation based noise removal algorithms, Phys. D, 60 (1992), 259-268.
[17] Y. Shih, C. Chien and C. Chuang, An adaptive parameterized block-based singular value decomposition for image de-noising and compression, Appl. Math. Comput., 218 (2012), 10370-10385.
[18] Y. Shih and H. C. Elman, Modified streamline diffusion schemes for convection-diffusion problems, Comput. Meth. Appl. Mech. Eng., 147 (1999), 137-151.
[19] Y. Shih, R. B. Kellogg and Y. Chang, Characteristic tailored finite point method for convection-dominated convection-diffusion-reaction problems, J. Sci. Comput., 47 (2011), 198-215.
[20] Y. Shih, C. Rei and H. Wang, A novel PDE based image restoration: convection-diffusion equation for image denoising, J. Comput. Appl. Math., 231 (2009), 771-779.
[21] A. N. Tikhonov, Solution of incorrectly formulated problems and the regularization method, Soviet Math. Dokl., 4 (1963), 10351038.
[22] Y. Wang, J. Guo, W. Chen and W. Zhang, Image denoising using modified Perona-Malik model based on directional Laplacian, Signal Process., 93 (2013), 2548-2558.
[23] J. Weickert, Anisotropic Diffusion in Image Processing, ECMI Series, Teubner-Verlag, Stuttgart, Germany, 1998.

https://doi.org/10.4208/cicp.scpde14.30s Published online by Cambridge University Press