A Fast Wavelet Multilevel Approach to Total Variation Image Denoising

Kossi Edoh and John Paul Roop

Department of Mathematics
NC A&T State University
Greensboro, NC, USA
kdedoh, jproop@ncat.edu

Research partially supported by NSA Grant BAA-002-06

Abstract

In this paper we present an adaptive multilevel total variational (TV) method for image denoising which utilizes TV partial differential equation (PDE) models and exploits the multiresolution properties of wavelets. The adaptive multilevel TV method provides fast adaptive wavelet-based solvers for the TV model. Our approach employs a wavelet collocation method applied to the TV model using two-dimensional anisotropic tensor product of Daubechies wavelets. The algorithm inherently combines the denoising property of wavelet compression algorithms with that of the TV model, and produces results superior to each method when implemented alone. It exploits the edge preservation property of the TV model to reduce the oscillations that may be generated around the edges in wavelet compression. In contrast with previous work combining TV denoising with wavelet compression, the method presented in this paper treats the numerical solution in a novel way which decreases the computational cost associated with the solution of the TV model. We present a detailed description of our method and results which indicate that a combination of wavelet based denoising techniques with the TV model produces superior results, for a fraction of the computational cost.

***Keywords**: Wavelets, Collocation, Adaptive Grid Refinement, Image Denoising*

1. Introduction

In this paper, we present an adaptive multilevel total variation (TV) method for image denoising which utilizes TV partial differential equation (PDE) model and exploits the multiresolution properties of wavelets. The purpose of this paper is to develop a fast method which combines TV denoising with denoising from wavelet compression, which is known to produce results which are superior to either method alone. In contrast with previous work combining TV denoising with wavelet compression [1], the method presented in this paper treats the numerical solution in a novel way which decreases the computational cost associated with the solution of the TV model.

Adaptive multilevel TV methods can prove useful in applications in which noisy images are utilized to produce immediate feedback. Such applications include but are not limited to real-time sensor image data, biometric imaging (e.g. iris recognition, fingerprint recognition), and images of fast moving military objects. In this paper we combine wavelets and TV PDE image denoising techniques in a natural way to produce denoising results which have improved image quality and compression ratio compared to regular wavelet image denoising, and for a fraction of the computational cost compared to other state-of-the-art techniques.

The theory of wavelets and variational partial differential equations (PDEs) have been used for a number of years in image processing. Both techniques are promising in the area of images denoising. We exploit multiscale phenomena in order to provide efficient and accurate image denoising techniques [2]. Image denoising and compression using the thresholding of wavelet coefficients has been a widely used technique. Among the pioneering work on wavelet shrinkage denoising are those by Weaver et. al. [3] and Donoho-Johnstone [4]. The computational challenge lies in how to determine a methodology for removing insignificant wavelet coefficients, while keeping the significant ones. Various techniques have been adopted such as hard and soft thresholding [5]. Higher dimensional wavelets such as shearlets and curvelets, and those obtained from more sophisticated tensor products of univariate wavelets are been used to improve on wavelet image denoising to reduce effects such as the Gibbs phenomena - the oscillations near discontinuities, c.f. [6].

PDE models have been used in recent decades in image processing by solving the PDEs in the image domain. Since the image is considered as a continuous function, sharp edges and other 2-dimensional phenomena can be modeled into this equation through the application of concepts such as curvature and gradients. As a result these methods are able to produce sharp edges and improved images. Some PDE models that have been used include the Perona-Malik anisotropic diffusion method [7], Rudin-Osher-Fatemi total variational method [8], Mumford-Shah model [9], and Alvarez-Guichard-Lions model [10].

The localization property of wavelets in space and scale makes them suitable for adaptive methods for solving PDEs. Images with sharp edges can be approximated with few number of wavelet bases also known as the degrees of freedom. While at the same time preserving the edges and preventing the effects of Gibbs phenomena. The approximation error of wavelets can be obtained by using the fact that polynomials of up to a certain degree are in the span of the wavelets basis. A review of this can be found in [11].

Two widely used wavelet-based multilevel methods for solving PDEs are those involving Galerkin [12], and collocation procedures [13, 14]. In Galerkin methods, the vanishing moments of orthogonal wavelet bases are used to approximate the derivatives. Unlike the collocation methods, the Galerkin-based methods are known to have difficulties with nonlinear terms, boundary conditions, and the geometries of the computational domain [15]. The collocation-based methods [14, 16] are often used to avoid these difficulties. In the collocation methods the derivatives can be computed in the wavelet space using methods such as matrix derivatives [14], finite difference operators [17], and the computation of the derivative in the image space and projecting the results in the wavelet space[18].

In this paper we focus on improving the speed and the efficiency of existing algorithms for image denoising. In [1] the authors used wavelet-collocation methods to solve the underlying PDEs by projection back and forth from wavelet space to image space to approximate the derivatives and nonlinearities in the TV operator. In this article, we approximate a differential operator in a particular coordinate direction by performing wavelet projections in that coordinate direction alone. This increases the speed of the method since we can exploit sparsity structures in order to perform the finite differencing. We draw an important distinction between our method and that in [1], which is that the nonlinearity is evaluated in the wavelet domain as opposed to the image domain, which along with our method of finite differencing, allows for efficient computations. Although there are marked differences between treating the nonlinearity in in the image domain versus the wavelet domain, our computations do indicate an improvement of the peak signal-to-noise ratio (PSNR) over the TV denoising method and the wavelet denoising method.

In addition, we want to contrast our method with the Wavelet-Optimized Finite Difference Methods (WOFD), (WOFD2), in e.g. [13]. In this method, wavelet thresholding is utilized to generate a sparse grid, and then the PDE is approximated on the sparse grid in the physical space using usual sparse PDE differencing techniques. This method is useful from the standpoint of typical PDEs in mathematical physics as problems from nonlinearities and boundary conditions can be avoided. However, from our computational experiments, such methods are not necessarily appropriate for approximating the TV model, as one would lose the superior ability of wavelets to compress and interpolate image data.

Among the existing state-of-the-art methods are Dabov-Foi-Katkovnik-Egiazarian collaborative filtering [19], Portilla-Strela-Wainwright-Simoncelli [20], Kervrann-Boulanger [21], and Buades-Coll-Morel nonlocal means [22]. In the experiments we compared our method with that of nonlocal means [22]. We used the Matlab nonlocal means toolbox for our computations. It is significant to note that the proposed method can be applied to other PDE models in image processing such as Perona-Malik filtering [7].

Our results indicate that using wavelets to compress TV denoised images results in a higher compression ratio than the regular wavelet thresholding methods. Superior denoised images are obtained from the adaptive multilevel TV method when compared to those obtained from wavelet or TV denoising alone. In addition, we note that solving the PDE in the wavelet domain is less expensive than solving the PDE in the image domain on the full grid. We note that the denoised images obtained from using the method of nonlocal means are superior, but this algorithm is vastly more computationally intensive.

The paper is organized as follows: Section two introduces the total variation model and discusses the numerical technique used to solve the associated PDE. Section three reviews the background behind Daubechies-type wavelets and indicates how wavelet coefficients may be used to generate sparse grids for use in numerical PDE computations. Section four presents results from several numerical experiments involving the TV model, wavelet-based image denoising, and the wavelet multilevel solution of the TV model. Finally, Section five provides concluding remarks and gives direction for future research.

2. The TV Model

In this section, we discuss the numerical implementation of the TV model. Let Ω be a closed bounded domain in R^2. The solution u is obtained by finding a stationary solution to the following initial value problem

$$\frac{\partial u}{\partial t} - \nabla \cdot \left(\frac{\nabla u}{|\nabla u_\varepsilon|} \right) = \lambda(f-u), \;\; in\ \Omega, \tag{1}$$

$$u(x,y,0) = f, \;\; in\ \Omega, \tag{2}$$

where f is the original (noisy) image, $|\nabla u_\varepsilon| := \sqrt{u_x^2 + u_y^2 + \varepsilon^2}$ is used to regularize the gradient, and u is subject to boundary conditions. In the case of image processing, several different types of boundary conditions are implemented. In the articles [23, 8], zero Neumann boundary conditions are utilized. However, it is also possible to implement other boundary conditions. In the case of evaluating finite difference quotients associated with the spatial discretization of (1), one can implement the ``boundary conditions" by constant extension, periodic extension, or reflection.

In this paper, we utilize the fairly naive solution approach which is to discretize in time using the forward Euler finite difference method and apply second order finite difference estimates to approximate the differential operators found in (1), although more sophisticated models and solution techniques have been utilized in the literature. We derive the approximation scheme by first applying the divergence operator through the second term in (1) and discretizing each of the resulting differential operators.

Set $u^0 := f$ *and for* $n = 1,2,\dots$ *solve:*

$$u^{n+1} = u^n + \Delta t\big(A(u^n) + \lambda(f - u^n)\big), \tag{3}$$

where

$$A(u) := \frac{(u_x^2 + \varepsilon^2)u_{yy} + (u_y^2 + \varepsilon^2)u_{xx} - 2u_x u_y u_{xy}}{(u_x^2 + u_y^2 + \varepsilon^2)^{3/2}}. \tag{4}$$

We implement each of the differential operators in (4) as follows. The first order derivatives are discretized by the symmetric difference quotient, the second order derivatives are discretized by the usual second order difference quotients, while the mixed derivative term is discretized by successively applying the first order symmetric difference quotients. That is:

$$u_x \approx \big(u(x+h_x, y) - u(x-h_x, y)\big)/2h_x,\; u_y \approx \big(u(x, y+h_y) - u(x, y-h_y)\big)/2h_y,$$

$$u_{xx} \approx \big(u(x+h_x, y) - 2u(x, y) + u(x-h_x, y)\big)/h_x^2,$$

$$u_{yy} \approx \big(u(x, y+h_y) - 2u(x, y) + u(x, y-h_y)\big)/h_y^2,$$

$$u_{xy} \approx \big(u(x+h_x, y+h_y) - u(x+h_x, y-h_y) - u(x-h_x, y+h_y) + u(x-h_x, y-h_y)\big)/4h_x h_y.$$

3. Wavelet Multilevel Methods

This section describes the multiresolution properties of wavelets and how they are used in the implementation of wavelet-based adaptive-collocation methods for solving PDEs. Wavelets can be localized in space and scale and are designed to represent functions, data and operators in different frequency components. Here we focus our attention on Daubechies-type wavelets [24] and discuss the finite dimensional transform. We show how wavelets can be used to generate a sparse grid for the wavelet computational domain and how this is effectively used to compute the derivatives in the image domain. The section includes the generation of 2-dimensional wavelets using tensor products.

3.1. Daubechies-based Wavelet Approximation

The Daubechies-based wavelets are defined in [24] and they involve a scaling function ϕ and a wavelet function ψ. A dilation equation for a compactly supported scaling function satisfies the equation

$$\phi(x) = \sqrt{2}\sum_{k=0}^{2p-1} h_k \phi(2x - k), \tag{5}$$

where the support of $\phi(x)$ is $[0, 2p-1]$ and p is the order of the scaling function. The corresponding wavelet function is given by

$$\psi(x) = \sqrt{2}\sum_{k=0}^{2p-1} g_k \phi(2x-k), \tag{6}$$

with support of $\psi(x)$ on the interval $[-p+1, p]$. The constants h_k and g_k are the low-pass and high-pass filter coefficients respectively. The function ψ satisfies the conditions

$$\begin{gathered}\int x^n \psi(x)dx = 0, \\ 0 \le n \le p,\end{gathered} \tag{7}$$

known as the vanishing moments. The number of wavelets vanishing moments p is related to the approximation error of wavelet basis functions. The following functions

$$\begin{gathered}\phi_{j,k}(x) = 2^{j/2}\phi(2^j x-k), \quad and \\ \psi_{j,k}(x) = 2^{j/2}\psi(2^j x-k), \quad j,k \in Z\end{gathered} \tag{8}$$

are elements of the scaling and wavelets orthonormal basis respectively. The index j is the dilation parameter and the index k the translation parameter. The translated scaling functions and the corresponding translated wavelet functions span the spaces denoted V_j and W_j respectively and are given by

$$\begin{gathered}V_j = span\{2^{j/2}\phi(2^j x-k), k \in Z\}, \forall j \in Z, \\ W_j = span\{2^{j/2}\psi(2^j x-k), k \in Z\}, \forall j \in Z,\end{gathered} \tag{9}$$

where $V_{j+1} = V_j \oplus W_j$ and $V_j \subset V_{j+1} \subset \ldots \subset L^2$. Hence it can be deduced that $L^2(R) = \oplus_{j\in Z} W_j$ and for any $f(x) \in L^2$

$$f(x) = \sum_{j\in Z}\sum_{k\in Z} d_{j,k}\psi_{j,k}(x). \tag{10}$$

Without loss of generality we define V_j and W_j as the spaces corresponding to a finite interval and given by

$$\begin{gathered}V_j = span\{\phi_{j,k}(2^j x-k), k = 0\ldots 2^j-1\}, \forall j \in Z, \\ W_j = span\{\psi_{j,k}(2^j x-k), k = 0\ldots 2^j-1\}, \forall j \in Z,\end{gathered} \tag{11}$$

where V_{j+1} can be expressed as

$$V_{j+1} = V_0 \oplus W_0 \oplus W_1 \oplus \ldots \oplus W_j.$$

Let

$$\begin{gathered}L^2(R) = span\{\phi_{0,k}(x),\ k = 0.\ldots,2^j-1\} \\ \oplus\{\oplus_{j=0}^{\infty} span\{\psi_{j,k},\ k = 0,\ldots,2^j-1\}\}.\end{gathered}$$

A continuous function $f(x) \in L^2(R)$ can be approximated for N large by

$$f(x) \approx P_{V_0} f(x) = \sum_k f_{0,k}\phi_{0,k}(x) + \sum_{j=0}^{N}\sum_k d_{j,k}\psi_{j,k}(x). \tag{12}$$

3.2. Wavelets in Two Spatial Dimensions

Let the computational domain be a rectangular box. The anisotropic tensor product for the two-dimensional wavelets is defined by

$$\Psi_{j_1,k_1;j_2,k_2} = \psi_{j_1,k_1}(x)\psi_{j_2,k_2}(y), \quad j_1, j_2, k_1, k_2 \in Z, \tag{13}$$

and form an orthogonal basis in $L^2(R^2)$. To maintain multiresolution properties of the wavelets we define

$$\Phi_{j,k,n}(x,y) = \phi_{j,k}(x)\phi_{j,n}(y), \tag{14}$$

such that at level j, $V_j = V_j \otimes V_j$ spans $L^2(R^2)$. The corresponding orthogonal complement of V_j is

$$\mathbf{W}_j = (W_j \otimes W_j) \oplus (W_j \otimes V_j) \oplus (V_j \otimes W_j). \tag{15}$$

The detail space is then spanned by three wavelets

$$\begin{aligned} \Psi_1(x,y) &= \psi(x)\psi(y), \\ \Psi_2(x,y) &= \psi(x)\phi(y), \\ \Psi_3(x,y) &= \phi(x)\psi(y), \end{aligned} \tag{16}$$

where Ψ_1 is the diagonal, Ψ_2 the vertical, and Ψ_3 the horizontal. This method leads to a linear order of successively finer approximations spaces as in the univariate case. We use this particular method to generate the scaling and wavelet coefficients utilized in our experiments. Note that more sophisticated 2-dimensional wavelet representations exist. It would be interesting for future study to see how effective exotic wavelets are when utilized in wavelet-collocation PDE methods.

3.3. The Adaptive Multilevel TV Method

In this section, we describe the adaptive multilevel TV method in which a solution of the TV model is obtained by discretizing the corresponding PDE with respect to the wavelet coefficients. Borrowing the notation from [1], we rewrite the usual representation of a function f in terms of scaling and wavelet functions,

$$f(x) = \sum_k \langle f, \phi_{0,k} \rangle \phi_{0,k}(x) + \sum_{j=0}^{N} \sum_k \langle f, \psi_{j,k} \rangle \psi_{j,k},$$

as a single basis with appropriate indexing

$$f(x) = \sum_{j,k} \alpha_{j,k} \varphi_{j,k}(x),$$

where the ``wavelet" coefficients are found by the formula

$$\alpha_{j,k} = \langle f, \varphi_{j,k}(x) \rangle. \tag{17}$$

In general, we think of an adaptive multilevel method in the following way. In order to approximate the solution to the partial differential equation

$$u_t = \mathsf{A}(u) + \lambda(f - u), \tag{18}$$

where A is some abstract, potentially nonlinear, operator, we instead approximate the solution to the system of ordinary differential equations (ODEs)

$$(\beta_{j,k})_t = \mathsf{A}(\beta_{j,k}) + \lambda(\alpha_{j,k} - \beta_{j,k}), \ \forall j,k, \tag{19}$$

where by $\alpha_{j,k}$, we denote the wavelet coefficients associated with the original ``noisy" image f (17), and by $\beta_{j,k}$, denote the wavelet coefficients associated with the solution to the system of ODEs (18).

The implementation however is not as straightforward as one might think. One has to interpret the mathematical operators within the operator A in terms of the wavelet coefficients. One would naturally expect with the compression and interpolation properties of wavelets, the numerical solution of such a system would require far less computational effort than a full scale brute force computational routine.

Specifically, for the TV model, the operator A has representation

$$\mathsf{A}(u) := \nabla \cdot \left(\frac{\nabla u}{|\nabla u_\varepsilon|} \right),$$

which contains both partial differential operators and nonlinearities.

In [1], the authors treated the numerical solution of (19) by solving

$$\beta_{j,k}^{(n+1)} = \beta_{j,k}^{(n)} + \Delta t(\mathsf{A}(\beta_{j,k}^{(n)}) + \lambda(\alpha_{j,k} - \beta_{j,k}^{(n)})), \ \forall j,k, \tag{20}$$

where

$$\mathsf{A}(\beta_{j,k}) := \int \nabla \cdot \left(\frac{\nabla u}{|\nabla u_\varepsilon|} \right) \varphi_{j,k} dx,$$

which means that for every time step, the authors transform the wavelet coefficients back into the full spatial domain, calculate the operator, and then transform back. The authors were able to obtain results superior to that than either the TV model or wavelet compression alone can provide. In this paper, we consider an adaptive multilevel solution technique which is fully implemented in wavelet space and therefore can be optimized for computational speed in addition to solution efficacy.

In this article, we provide a solution technique which utilizes the time iteration scheme (20) along with the spatial discretization

$$\begin{aligned} \mathsf{A}(\beta_{j,k})) &:= [((\beta_{j,k})_x^2 + \varepsilon^2)(\beta_{j,k})_{yy} \\ &+ ((\beta_{j,k})_y^2 + \varepsilon^2)(\beta_{j,k})_{xx} \\ &- 2(\beta_{j,k})_x(\beta_{j,k})_y(\beta_{j,k})_{xy}] \\ &/\left[((\beta_{j,k})_x^2 + (\beta_{j,k})_y^2 + \varepsilon^2)^{3/2}\right] \end{aligned} \tag{21}$$

which is the same as (4), discretized entirely in the wavelet space. The principal differences between our method and the method in [1] is that the nonlinearity is treated in wavelet space (eliminating the need to transform all the way back to the spatial domain) and that partial derivative calculations exploit the sparsity structure from wavelet compression.

What we have left to explain is the numerical approximation of the differential operators in (21). Essentially, we follow the approach in [25, 26]. A computational procedure for implementing the anistropic tensor product wavelet transform is simpler to think about than even the theoretical consideration. Essentially, one takes the transform in one spatial dimension, and then takes the transform in the other spatial dimension(s). One interesting fact to notice is that compact finite difference operators in one spatial dimension and wavelet transforms in any of the other spatial dimensions commute. Therefore, we discretize each of the derivatives in (21) by taking the inverse transform in one coordinate direction, evaluating the finite difference stencil, and then transforming back into the tensor product wavelet space. Note that for the mixed derivative term, we evaluate the partial with respect to x first and then the partial with respect to y (the same effect as the discretization mentioned in Section 2.)

For example, consider the discretization of one of the terms in (21), $(\beta_{j,k})_x$. In order to obtain this, one performs the following operations:

1. Transform $\beta_{j,k}$ by performing P_x^{-1} (the inverse transform in the ``x" coordinate direction.)
2. Evaluate the finite difference quotient $u_x \approx \left(u(x{+}h_x, y) - u(x{-}h_x, y)\right)/2h_x$.
3. Find $(\beta_{j,k})_x$ by performing P_x (the forward transform in the ``x" coordinate direction.)

Note that the numerical values obtained from this procedure are exactly the same as when transforming all the way back to the spatial domain, as in [1].

Because of the fact that wavelet transforms and finite difference stencils commute in different spatial directions, we achieve the same (in infinite precision arithmetic) values for the difference quotients as when transforming all the way back to the full spatial domain. However, when considering wavelet compression, sparsity structures can be exploited in which the code can be optimized to run extremely fast. For a detailed implementation of this, see [25, 26], and the references therein.

4. Numerical Experiments

In this section, we present a series of numerical experiments which show that image denoising performed with the adaptive multilevel TV method in (20), (21) produces numerical results which are as good if not better than those produced with either the usual wavelet threshold denoising or the TV model on the full grid alone. For each numerical experiment, we begin with a noisy image and present numerical results associated with four denoising algorithms: TV denoising, denoising by wavelet hard thresholding, denoising by the adaptive multilevel TV method, and denoising by nonlocal means. We see that in terms of providing the best quality denoised image, nonlocal means (NL) performs best. However, the adaptive multilevel TV method provides a significant reduction in computational time. We consider three distinct cases to illustrate the advantages and disadvantages of each of the computational algorithms. For the first experiment, we consider the Lenna image corrupted with a relatively small amount of noise. The Lenna image possesses many details, so that wavelet denoising and the adaptive multilevel TV model are not significanly better than the TV model. For the second experiment, we consider an image of concentric circles corrupted with a higher amount of noise. We notice that the higher amount of noise will challenge the algorithm, and the lack of details means that wavelet denoising and the adaptive multilevel TV method will outperform the TV model. For the final experiment, we consider an image with large amount of background and a rocket in the lower right portion of the image. For such images, the adaptive multilevel TV method produces PSNR levels which are comparable with the superior nonlocal means algorithm.

In order to utilize the grid adaptation strategy implemented in [13], images must be of size $N \times N$, where N is one more than a power of two. For our experiments, we utilize images which are 257×257 pixels, thus the highest possible level of wavelet decomposition is 8. In our experiments, however, our wavelet multilevel method was based on a four level wavelet decomposition using Daubechies wavelets of order four. In order to measure the effectiveness of each of our denoising strategies, we utilize the PSNR defined by

$$PSNR := 10\log_{10}\left(\frac{\sum_{i,j=0}^{256} 255^2}{\sum_{i,j=0}^{256}(g_{i,j}-u_{i,j})^2}\right), \tag{22}$$

where g denotes an original image, and u denotes the image after noise is added and then one of our noise reduction strategies is applied.

We implemented the adaptive wavelet multiscale TV model in a non-optimized way in C++ and in an optimized way using the software package AWFD (available online at http://wissrech.iam.uni-bonn.de/research/ projects/AWFD/index.html) [25, 26]. In Experiments One, Two and Three, we illustrate the speed and efficacy of adaptive multilevel TV method by utilizing the AWFD source code as a black box and implementing the adaptive wavelet multiscale total variation denoising model using only functionality which comes prepackaged with the software. The numerical method implemented was an explicit Euler method with differential operators approximated as described in Section 2. One of the drawbacks of the AWFD software is that the choice of boundary conditions and refinement strategies are limited to those prepackaged with the code. Thus we augmented the first three computational experiment with a fourth experiment in which we illustrate the differences between boundary conditions and refinement strategies in the adaptive multilevel TV method. Should this method achieve prominence in an applied/engineering area, one would have to write a more sophisticated optimized code which is designed for image processing applications as opposed to the AWFD code, which was intended for use with spline based multilevel solutions for PDEs arising in mathematical physics. In addition, our nonlocal means calculations were performed using nonlocal means toolbox in Matlab by Gabriel Peyré (available online at http://www.mathworks.com) [22]. Note that the nonlocal means toolbox is implemented in C and ported to Matlab using a ``mex" library, so the computation times are comparable. Code was compiled using the GNU g++ compiler and executed on a desktop computer with 3.19 GHz processor and 1.99 GB of RAM. Computational times reflect real time running times on this machine.

Experiment 1: Lenna. The lenna image was first corrupted with additive white noise at a rate of 29.6 PSNR. We then executed 200 time steps of the TV algorithm with $\Delta t = 0.2$, $\lambda = 0.3$, and $\varepsilon = 1$.

First, we implemented the TV model for the full grid, then utilized the adaptive multilevel TV method for various values of the threshold. The grid was readjusted after each ten time steps. It is important to note that these thresholds are fairly high because we treated the computational domain as $\Omega = [0,256]\times[0,256]$; thus the finest mesh size was $h = 1$. For this experiment we did not implement any boundary conditions, just utilizing the wavelet coefficients provided in the AWFD code. This is the equivalent of using constant extension for boundary conditions. Numerical results for this experiment are presented in the top panel of Table 1 and pictures corresponding to this experiment are given in Figure 1. It is important to note that the combination of the two techniques, wavelet denoising and total variation denoising provide PSNR which is higher than either method alone can achieve. We also compared the results of our experiment with nonlocal means for Gaussian with variance 0.03. The nonlocal means algorithm outperformed any of the other three methods in terms of PSNR, but it is more computationally intensive.

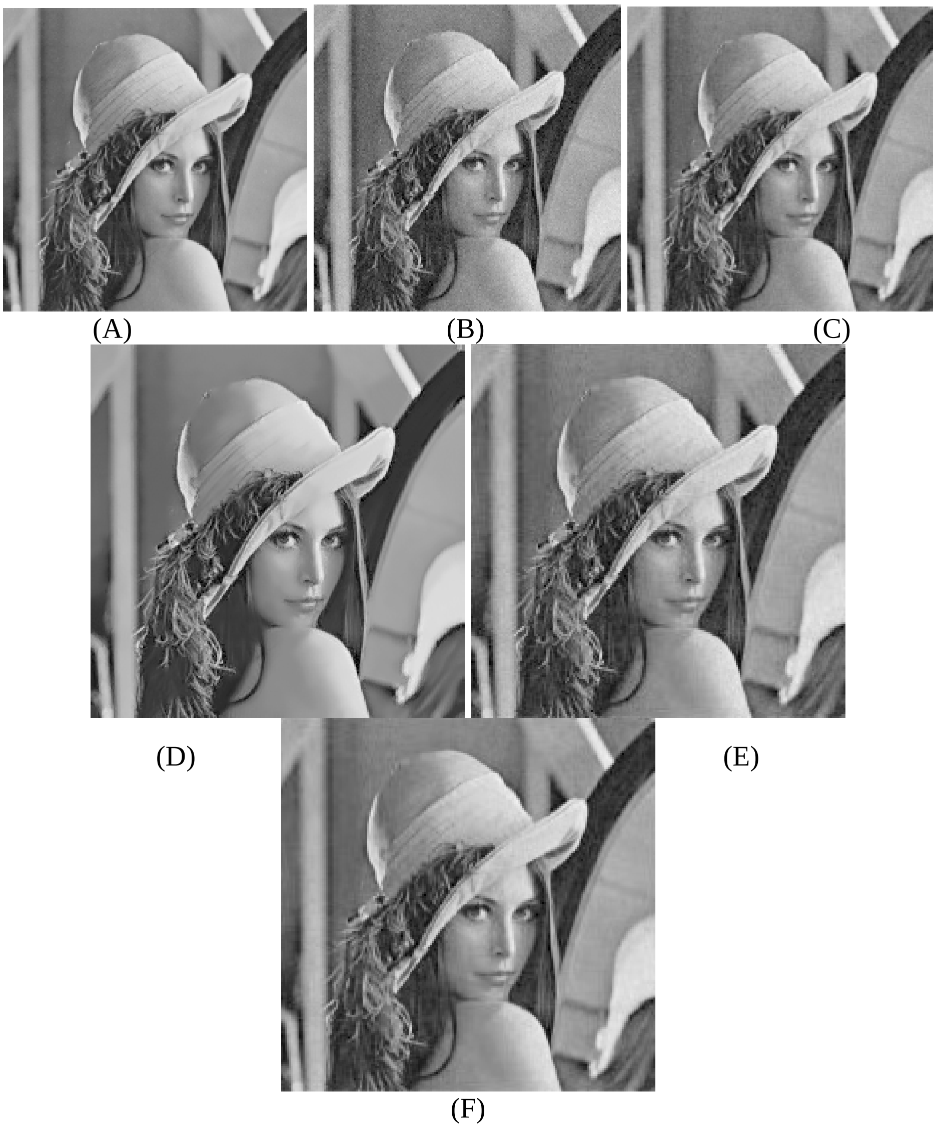

Figure 1. Pictures for various image denoising algorithms applied to the lenna image in Experiment 1. (A) Original image; (B) noisy image (29.6 PSNR); (C) denoised using TV model with $\lambda = 0.3$; (D) denoised using nonlocal means for Gaussian with variance 0.03; (E) denoised using wavelet thresholding, threshold = 40.0; (F) denoised using adaptive multilevel TV method.

Experiment 2: Circles. The circles image was first corrupted with additive white noise at a rate of 19.83 PSNR. We chose to add noise at this higher rate in order to challenge the algorithm. We then executed 200 time steps of the TV algorithm with $\Delta t = 0.2$, $\lambda = 0.1$, and $\varepsilon = 1$. (We utilize a lower value of λ because we expect the results of our algorithm to be further from the original guess.) First, we implemented the TV model for the full grid, then utilized the adaptive multilevel TV method for

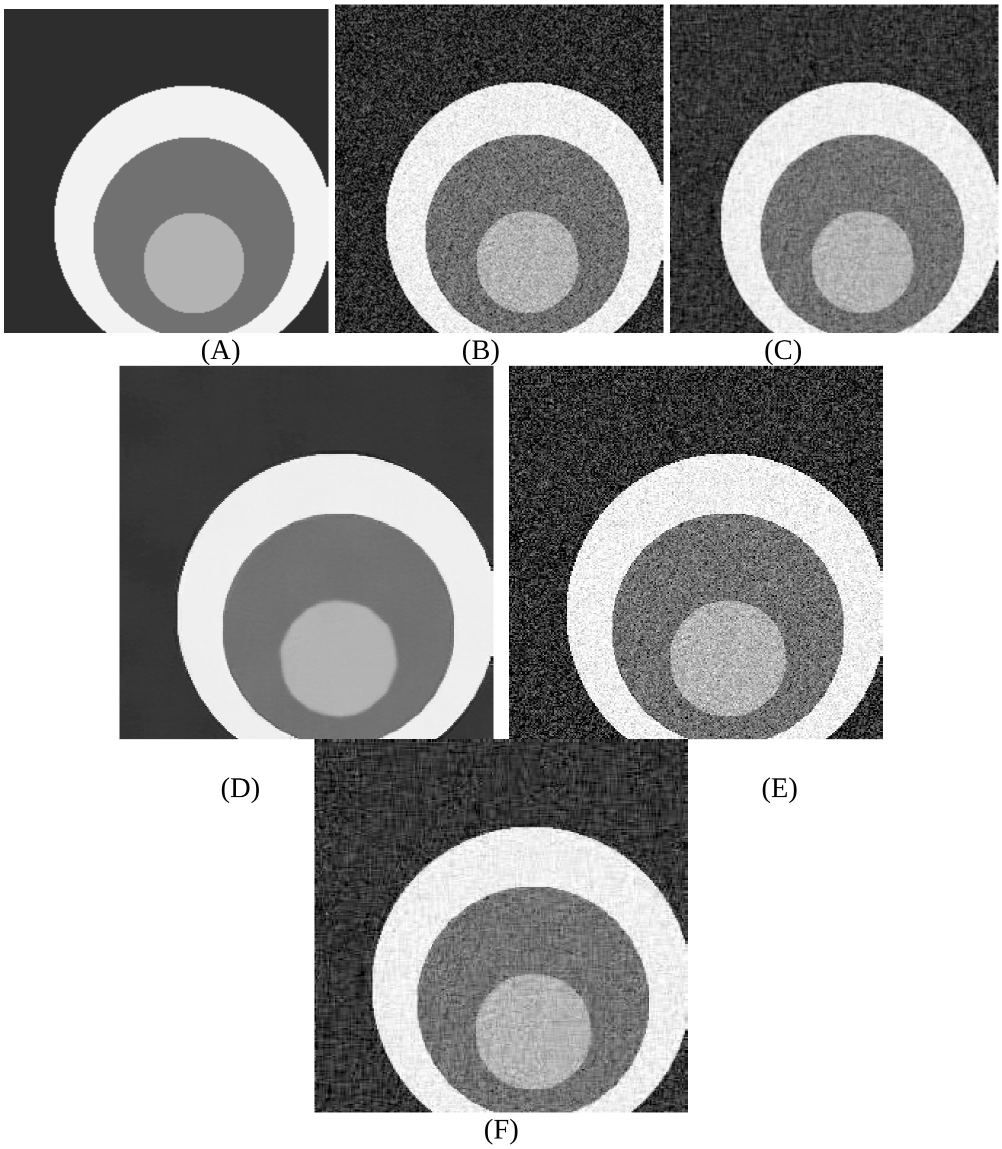

Figure 2. Pictures for various image denoising algorithms applied to the cameraman image in Experiment 2. (A) Original image; (B) noisy image (19.83 PSNR); (C) denoised using TV model with $\lambda = 0.1$; (D) denoised using nonlocal means for Gaussian with variance 0.10; (E) denoised using wavelet thresholding, threshold = 40.0; (F) denoised using adaptive multilevel TV method.

various values of the threshold. The grid was readjusted after each ten time steps. Numerical results for this experiment are presented in the second panel of Table 1 and pictures corresponding to this experiment are given in Figure 2. As in the previous experiment, we compared the results of our experiment with nonlocal means for Gaussian with variance 0.10.

Experiment 3: Rocket. This experiment is different as we hope that the wavelet denoising part of the algorithm will remove grid points associated with the background portion of the image. Also, for this experiment, the image was corrupted with additive white noise at a rate of 23.91 PSNR. We then executed 200 time steps of the TV algorithm with

$\Delta t = 0.2$, $\lambda = 0.2$, and $\varepsilon = 1$. First, we implemented the TV model for the full grid, then utilized the adaptive multilevel TV method for various values of the threshold. The grid was readjusted after each ten time steps. For this experiment we implemented two different types of boundary conditions. First, we did not enforce any boundary conditions, just utilizing the wavelet coefficients.

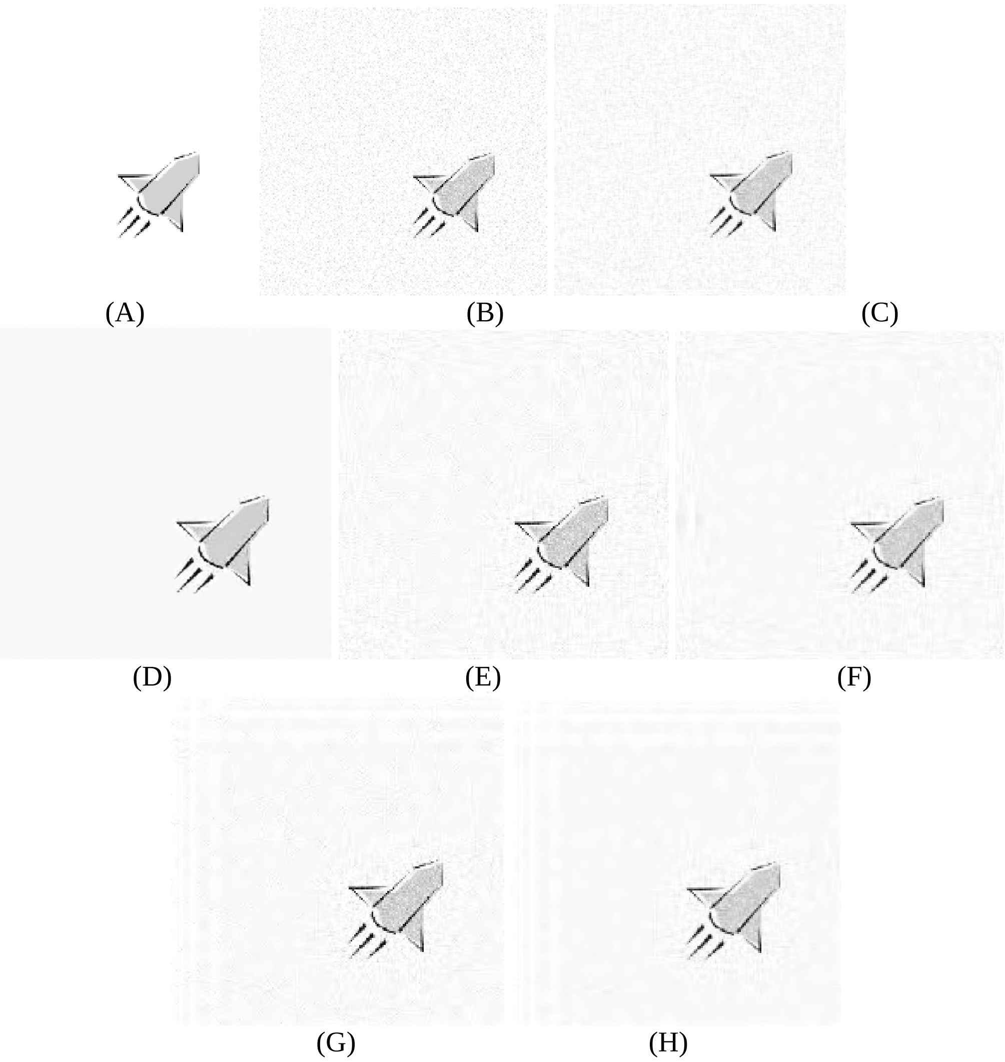

Figure 3. Pictures for various image denoising algorithms applied to the rocket image in Experiment 3. (A) Original image; (B) noisy image (23.91 PSNR); (C) denoised using TV model with $\lambda = 0.2$; (D) denoised using nonlocal means for Gaussian with variance 0.05;(E) denoised using wavelet thresholding with constant extension, threshold = 40.0; (F) denoised using adaptive multilevel TV method with constant extension; (G) denoised using wavelet thresholding with periodic extension, threshold = 40.0; (H) denoised using adaptive multilevel TV method with periodic extension.

Threshold	PSNR (WD)	PSNR (ATV)	DOF	CPU
TV-MODEL	--	32.66	66049	78.7s
NL-MEANS	--	33.75	--	277.9s
10.0	29.61	32.65	54057	90.9s
20.0	30.45	32.80	38139	69.0s
30.0	31.85	32.72	30414	57.9s
40.0	32.16	32.45	25358	50.2s

(A) Lenna (Experiment 1)

Threshold	PSNR (WD)	PSNR (ATV)	DOF	CPU
TV-MODEL	--	24.29	66049	84.3s
NL-MEANS	--	31.54	--	278.8s
10.0	19.83	24.29	64280	112.0s
20.0	19.84	24.31	56613	100.5s
30.0	19.84	24.39	46722	89.7s
40.0	19.96	24.74	35973	76.0s
50.0	20.46	25.26	27113	61.9s
100.0	25.98	26.69	10368	28.8s

(B) Circles (Experiment 2)

Threshold	PSNR (WD)	PSNR (ATV)	DOF	CPU
TV-MODEL	--	26.61	66049	78.8s
NL-MEANS	--	28.73	--	278.1s
10.0	23.91	26.61	55824	94.1s
20.0	23.99	26.87	32681	66.0s
30.0	25.08	27.56	15199	37.6s
40.0	26.81	27.76	10066	24.6s
50.0	27.54	27.74	8991	22.0s
100.0	27.73	27.59	6348	17.8s

(C) Rocket -- Constant Extension (Experiment 3)

Threshold	PSNR (WD)	PSNR (ATV)	DOF	CPU
TV-MODEL	--	27.09	66049	76.1 s
NL-MEANS	--	28.73	--	278.1s
10.0	24.14	27.10	53787	89.3s
20.0	24.25	27.54	26984	57.6s
30.0	25.80	28.37	9692	28.7s
40.0	27.72	28.62	6412	17.5s
50.0	28.42	28.66	5722	15.4s
100.0	28.56	28.45	3901	12.6s

(D) Rocket -- Periodic Extension (Experiment 3)

Table 1. (A) Data from Experiment 1 using the lenna image; (B) data from Experiment 2 using the cameraman image; (C) data from Experiment 3 using the rocket image using constant extension; (D) data from Experiment 3 using the rocket image using periodic extension. The first column gives the threshold, (first row for the TV model and second row for nonlocal means), the second column gives PSNR obtained from denoising by wavelet hard thresholding, the third column gives PSNR obtained from adaptive multilevel TV method, the fourth column gives degrees of freedom, and the final column gives CPU times.

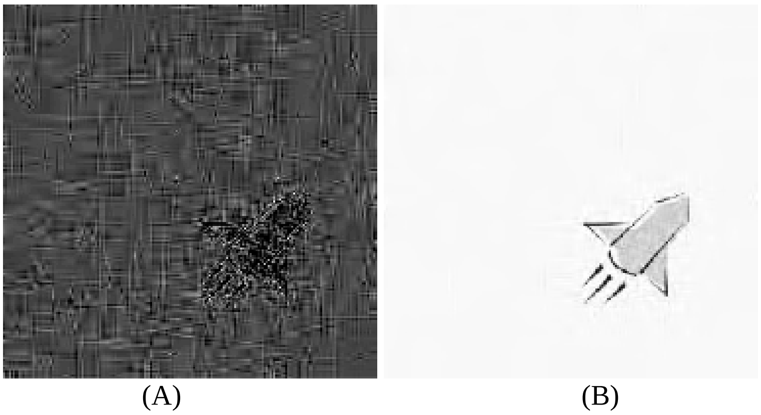

Figure 4. Pictures of adaptive multilevel TV method in Experiment 4 using mirror conditions. (A) Remaining noise in adaptive multilevel TV method with mirror extension, threshold = 20; (B) adaptive multilevel TV method with mirror extension, threshold = 40.

Threshold	PSNR (ATV)	DOF	CPU
Full	26.47	65536	21.2s
10.0	26.71	29492	17.2s
20.0	27.39	10006	14.0s
30.0	27.77	4557	12.7s
40.0	27.84	3441	12.6s
50.0	27.78	2890	12.4s
100.0	26.95	1495	12.1s

Table 2. Data from Experiment 4, adaptive multilevel solution using the rocket image with mirror extension. The first column gives the threshold, (first row for full grid), the second column gives PSNR obtained from adaptive multilevel TV method, the third column gives degrees of freedom, and the final column gives CPU times.

This is the equivalent of using constant extension boundary conditions. Second, we implemented periodic boundary conditions, which produce superior results as there is a region of white space around the interesting portion of the image. Numerical results for this experiment are presented in bottom panel of Table 1 and pictures corresponding to this experiment are given in Figures 3 and 4. For this example, the wavelet denoising procedure works well, but as before when combining this with the TV model, the results are much improved. As in the previous experiment, we compared the results of our experiment with nonlocal means for Gaussian with variance 0.05. Notice that for this experiment, the PSNR produced by the adaptive multilevel TV algorithm rivals that of the nonlocal means algorithm, for less computational cost.

Experiment 4: Rocket (continued). For the fourth and final computational experiment, we implemented the adaptive multilevel TV method in self-written C++ code. For this experiment we did not optimize the code for speed, but instead focused upon distinct refinement strategies and boundary conditions. In the AWFD source code, the authors implement a refinement strategy in which collocation points are included if neighboring collocation points have wavelet coefficients greater than the threshold both within a level and within the next finest level. In our code, we only included collocation points if the neighboring wavelet coefficients are greater than the threshold within a certain level. We observed that no computational accuracy was lost in implementing this less conservative marking strategy.

In addition, we compared three distinct types of boundary conditions. Boundary conditions are utilized both in the definition of the wavelet transforms and in the discretization of the finite difference operators. We implemented the adaptive multilevel TV model using the same parameters as in Experiment 3 using boundary conditions specified by constant extension, periodic extension, and mirrored extension. In the literature, mirror boundary conditions are considered preferable for image processing applications.

Table 2 gives PSNR, degrees of freedom, and computation times associated with this experiment using mirror boundary conditions. Notice that the PSNR levels achieved by our implementation are comparable to those achieved in Experiment 3, without as many boundary artifacts as produced by the AWFD code. The PSNR obtained using each of the three types of boundary conditions are comparable. For example, when implementing the adaptive multilevel TV method with wavelet threshold value of 20.0, constant extension yields a PSNR of 27.34, periodic extension yields a PSNR of 27.41, and mirror extension yields a PSNR of 27.39. We also include Figure 4, in which we show the remaining noise after the adaptive multilevel TV method is applied with threshold 20, and the denoised image with threshold 40. Notice that the image in Figure 4(B) is qualitatively better than the images in Figure 3 (F), (H).

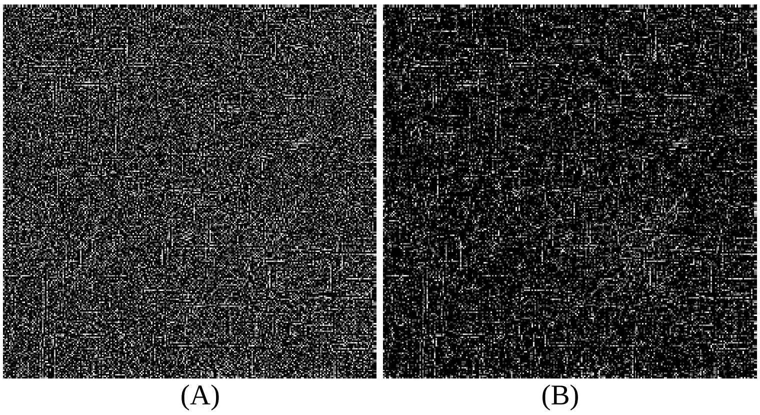

Figure 5. Pictures of nonlinear terms in the WRF model. (A) Nonlinear term (magnified by five) as treated in the wavelet space from Experiment 4 (noisy rocket image) ; (B) Difference (magnified by five) in evaluating nonlinear term in the image space versus evaluating the nonlinear term in the wavelet space.

When utilizing wavelet collocation methods for solving partial differential equations, difficulties may arise in the evaluation of nonlinear terms in the wavelet space [15]. Again, in the article [1], the authors treat the nonlinearity by transforming back to the image space. This technique preserves in principal the stationary solution of the TV model. In this paper, we have treated the nonlinear term in the wavelet space. Figure 5 illustrates the difference in this evaluation. In panel (A), the nonlinear term as calculated in our numerical scheme for the noisy rocket image is displayed, and is magnified by five to further see the details. In panel (B), the difference (in absolute value) of the treatment of the nonlinear term in the wavelet space and image space is displayed, and is magnified by five. There is a significant difference between the way in which the nonlinear term is evaluated for each of our methods and that of [1]. Certainly, the mathematical treatment of linear transformations over nonlinearities proves prohibitive when considering wavelet collocation techniques for the solution of partial differential equations. However, it is notable that our technique does improve on the PSNR obtained from either the TV denoising method or wavelet denoising method.

5. Conclusion

In this article, we have seen that implementing the total variation model for image denoising in an adaptive multiscale setting exploits in essence what is wavelet image denoising (and compression), which removes those high frequency coefficients which are less than some predefined threshold. We have observed that not only does the adaptive multilevel TV method reduce computation time in solving the TV model, but that the numerical results produced are superior to those produced by each of the wavelet denoising and TV model alone.

Direct extensions of this work include using the wavelet multilevel idea to discretize other PDEs useful in image processing, such as the Perona-Malik equation and other equations which rely on differening diffusion-like terms. A significant challenge would be to increase the computational speed for other state-of-the-art denoising algorithms. One of the restrictions of our approach is that it relies on biorthogonal wavelet bases, and cannot be trivially extended to more exotic geometric transformations like shearlets and curvelets [6].

Although the nonlinearity is treated in the wavelet space, our method shows improvement over the TV denoising method or wavelet denoising method, and improves the computational speed of the TV method while reducing Gibbs oscillations.

References

[1] T. Chan and H. Zhou, ``Total variation wavelet thresholding", *J. Sci. Comput.*, vol. 32, no. 2, pp. 315-341, 2007.

[2] E. Tadmor, S. Nezzar and L. Vese, ``A multiscale image representation using hierarchical (BV, L^2) decompositions," *Multiscale Model. Simul.*, vol. 2, no. 4, pp. 554 - 579, 2004.

[3] J.B. Weaver, Y. Xu, D.M. Healy Jr. and L.D. Cromwell, ``Filtering noise from images with wavelets transforms," *Magnetic Resonance in Medicine*, vol. 21, no. 2, pp. 288 - 295, 1991.

[4] D. Donoho and I. Johnstone, ``Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425-455, 1994.

[5] S. Mallat, ``A Wavelet Tour of Signal Processing," *Academic Press*, San Diego, CA, 1998.

[6] K. Guo and D. Labate, ``Optimally sparse multidimensional representation using shearlets", *SIAM J Math. Anal.,*, vol. 39, pp. 298-318, 2007.

[7] P. Perona and J. Malik, ``Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 7, pp. 629-639, 1990.

[8] L. Rudin, S. Osher and E. Fatemi, ``Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259-268, 1992.

[9] D. Mumford and J. Shah, ``Optimal approxiamtions by piecewise smooth functions and associated variational problems," *Comm. Pure Appl. Math.*, vol. XLVII, pp. 577-685, 1989.

[10] L. Alvarez, F. Guichard, P.L. Lions and J. M. Morel, ``Axioms and fundamental equations of image processing," *Arch. Ration. Mech. Anal.*, vol. 123, no. 3, pp. 199-257, 1993.

[11] R. DeVore, ``Nonlinear approximation", *Acta Numer.*, vol. 7, pp. 51 - 150, 1998.

[12] E. Bacry, S. Mallat, and G. Papanicolaou, ``A wavelet based space-time adaptive numerical method for partial differential equations," *RAIRO Model. Math Anal. Numer.*, vol. 26, no. 7, pp. 793-834, 1992.

[13] L. Jameson, ``A wavelet-optimized very high order adaptive grid and order numerical method," *SIAM J. Sci. Comput.*, vol. 19 no. 6, pp. 1980-2013, 1998.

[14] O.V. Vasilyev and S. Paolucci, ``A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain," *J. Comput. Phys.*, vol. 125, no. 2, pp. 498-512, 1996.
[15] S. Bertoluzza, G. Naldi, and J. C. Ravel, ``Wavelet methods for the numerical solutions of boundary value problems on the interval," in *Wavelets: Theory, Algorithms, and Applications*, C.K. Chui, L. Montefusco, and L. Puccio (eds), Academic Press, Inc., New York, pp. 425-448, 1994.
[16] M. Holmström, ``Solving hyperbolic PDEs using interpolating wavelets," *SIAM J. Sci. Comput.*, vol. 21, no. 2, pp. 405-420, 1999.
[17] A. Harten, ``Adaptive multiresolution schemes for shock computations," *J. Comput. Phys.*, vol. 115, pp. 319 - 338, 1994.
[18] O.V. Vasilyev and S. Paolucci, ``A fast adaptive wavelet collocation algorithm for multidimensional PDEs," *J. Comput. Phys.*, vol. 125, no. 2, pp. 16 - 56, 1997.
[19] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, ``Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Trans. Image Proc.*, vol. 16, no. 8, pp. 2080-2095, 2007.
[20] J. Portilla V. Strela, J. Wainwright, E.P. Simoncelli, ``Image denoising using Gaussian scale mixtures in the wavelet domain," *IEEE Trans. Image Proc.*, vol. 12, no. 11, pp. 1338-1351, 2002.
[21] C. Kervrann and J. Boulanger, ``Local adaptivity to variable smoothness for exemplar-based image regularization and representation," *Inter. Jour. of Computer Vision*, vol. 79 no. 1, pp. 45-69, 2008.
[22] A. Buades, B. Coll, and J.M. Morel, ``A review of image denoising algorithms, with a new one," *SIAM Journal on Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490-530, 2005.
[23] A. Marquina and S. Osher, ``Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal," *SIAM J. Sci. Comput.*, vol. 22, pp. 387-405, 2000.
[24] I. Daubechies, ``Orthonormal basis of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol. 41, pp. 909-996, 1988.
[25] M. Griebel, ``Adaptive sparse grid multilivel methods for elliptic PDEs based on finite differences," *Computing*, vol. 61, pp. 151-180, 1998.
[26] M. Griebel and F. Koster, ``Adaptive wavelet solvers for the unsteady incompressible Navier-Stokes equations," in *Advances in Mathematical Fluid Dynamics*, Malek et. al. (eds), Springer-Verlag, New York, 2000.

Authors

Kossi Edoh received his B.Sc. degree in Mathematics from University of Cape Coast, Ghana, in 1986, and his M.Sc. and Ph.D. degrees from Simon Fraser University, Canada, in 1991 and 1995.

He has taught Computer Science at Elizabeth City State University, and Montclair State University in the US for eight years as an assistant professor. He is currently an associate professor of Mathematics at North Carolina Agricultural and Technical State University, USA. His research interests include information security and numerical dynamical systems.

John Paul Roop received his B.Sc. degree in Mathematics from Roanoke College, USA, in 1999, and his M.Sc. and Ph.D. degrees in Mathematical Sciences from Clemson University, USA in 2001 and 2004.

He served as a postdoctoral associate in Mathematics at Virginia Polytechnic and State University from 2004 to 2006, and is currently an assistant professor of Mathematics at North Carolina Agricultural and Technical State University, USA. His research interests include finite element methods and adaptive/multiscale discretization techniques for partial differential equations.