A Two-Level Method for Image Denoising and Image Deblurring Models Using Mean Curvature Regularization

Faisal Fairag , Ke Chen , Carlos Brito-Loeza and Shahbaz Ahmad

Abstract

The mean curvature (MC)-based image denoising and image deblurring models are used to enhance the quality of the denoised images and deblurred images respectively. These models are very efficient in removing staircase effect, preserving edges and other nice properties. However, high order derivatives appear in the Euler-Lagrange equations of the MC-based models which create problems in developing an efficient numerical algorithm. To overcome this difficulty, we present a robust and efficient Two-Level method for MC-based image denoising and image deblurring models. The Two-Level method is consist of solving one small problem and one large problem. The small problem is a nonlinear system, having high order derivative, on Level-I (image having small number of pixels). The large problem is one less expensive system, having low order derivative, on Level-II (image having large number of pixels). The derivation of the optimal regularization parameter of Level-II is studied and formula is presented. Numerical experiments on digital images are presented to exhibit the performance of the Two-Level method.

Key words: denoising, deblurring, regularization, variational models, two-level methods, ill-posed problem

1 Introduction

The mean curvature (MC) [3, 15, 22, 23] based regularization functionals are widely use in all image processing problems. In image denoising and image deblurring, the MC-based models are very effective. These models not only preserve edges but also remove staircase effect in the recovery of digital images. These models have other nice properties as well. However, high order derivatives appear in the Euler-Lagrange equations of the MC-based models, which create problems in developing an efficient numerical algorithm. Furthermore MC-based models produce complicated nonlinear system of equations. The Jacobian matrix of such system has a block banded matrix structure with large

Department of Mathematics and Statistics, King Fahd University of Petroleum & Minerals, Saudi Arabia (ffairag@kfupm.edu.sa).

Center for Mathematical Imaging Techniques, The University of Liverpool, UK (k.chen@liv.ac.uk).

Computational Learning and Imaging Research Center, Autonomous University of Yucatán, Merida, Mexico, (carlos.brito@correo.uady.mx).

Department of Mathematics and Statistics, King Fahd University of Petroleum & Minerals, Saudi Arabia (shahbazahmad@kfupm.edu.sa).

bandwidth. The mean curvature-based models are effective, but due to high nonlinearity and high order derivative, efficient numerical solution is a crucial issue. To overcome these difficulties, in this paper we introduce a Two-Level method.

The Two-Level method is widely studied in other research areas [7, 10, 11, 12, 13] beyond image processing. These methods are so attractive for ill-conditioned and large nonlinear systems. The essence of these methods is that they need the solution of only a small nonlinear system of equations on Level-I (coarse mesh) and one linear system of equations on Level-II (fine mesh). For example, Jintao Xu [19, 20] first introduced Two-Level method to solve semilinear PDEs. Then the method have been examined for the Navier-Stokes (NS) equations in [9, 10, 14]. Other works on Two-Level method can be found here [5, 6, 8]. In these works, they used the coarse solution to find a best scaling between the coarse mesh size H and the fine mesh size h in order to obtain same order of convergence. In order to obtain a better quality solution on the fine mesh (Level-II) a good quality solution at coarse mesh (Level-I) is required. This requirement is the limitation of the Two-Level method.

In this paper, we will present and examine a Two-Level method for image denoising and image deblurring problems. In this method, we solve the problem at two different levels, one after the other. At Level-I we solve a nonlinear integral differential equation (image denoising and image deblurring) on a coarse mesh. At this level, the regularization functional is mean curvature functional. It means we are solving computationally expensive problem just on a coarse mesh at Level-I. Then the Level-I solution will be interpolated. At Level-II, we need to solve an integral differential equation (image denoising and image deblurring) which is linear with less computationally expensive regularization functional. At Level-II, instead of using MC we will use either total variation (TV) [16, 18] or Tikhonov [1, 17]. It means at Level-II, we have to do less expensive work. Because expensive part of the work is already done at Level-I on a coarse mesh of a small size. In our work, we have also used the coarse solution to calculate the optimal regularization parameter for the Level-II. Moreover, this optimal parameter is used to solve a linear system with low order derivative regularizer such as Tikhonov and TV. At Level-II, the mean curvature term is approximated using the coarse solution and kept in the right hand side.

The contributions of this paper include the following: (i) our work presents a robust, effective and less expensive numerical method for MC-based image denoising and image deblurring problems, (ii) presents a better treatment for the computationally expensive mean curvature-based regularization functionals. The proposed method will apply similarly to other image reconstruction problems. The paper is organized in different sections. The first section includes introduction while the second section includes problem description of image denoising and image deblurring models. In the third section, we present primal-dual form of mean curvature-based image denoising and image deblurring models. The cell descritization and cell-centered finite difference method (CCFD) method are also presented in third section. In forth section, we present One-Level method. In fifth section, we present our proposed Two-Level method and the parameters selection procedure. The numerical results are in the sixth section. The conclusions about the proposed Two-Level method is discussed in the last section of the paper.

2 Problem Description

The focus of the paper is on image denoising and image deblurring problems, so we start by presenting their concise description. Mathematically, the relationship between u (original image) and z (recorded image) is as follows;

$$z = \vec{K}u + \varepsilon \tag{1}$$

where ε is the noise function. In case of image denoising, $\vec{K} = I$ is an identity operator but in case of deblurring, $\vec{K}$ is a Fredholm-integral operator (first kind);

$$(\vec{K}u)(\mathbf{x}) = \int_\Omega k(\mathbf{x}, \mathbf{y})u(\mathbf{y})\ d\mathbf{y}, \quad \mathbf{x} \in \Omega$$

known as the blurring operator. The kernel $k(\mathbf{x}, \mathbf{y})$ satisfies the property of translation invariance i.e. $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y})$. Let Ω denotes a square in R^2. $u \in \Omega$ is an image intensity function. $\mathbf{x} = (x, y)$ defines the position in Ω. Let $|\mathbf{x}| = \sqrt{x^2 + y^2}$ is an Euclidean norm and $\|.\|$ is $L^2(\Omega)$ norm.

The problem (1) is not stable [1, 17, 18]. One remedy is to make a use of regularization functionals. For example, the Tikhonov regularization functional [1],

$$J_{Tik}(u) = \int_\Omega u^2\ \mathbf{dx},$$

the total variation (TV) regularization functional [16],

$$J_{TV}(u) = \int_\Omega |\bigtriangledown u|\ \ \mathbf{dx},$$

and mean curvature (MC) regularization functional [22],

$$J_{MC}(u) = \int_\Omega |\bigtriangledown . \frac{\bigtriangledown u}{\sqrt{|\bigtriangledown u|^2 + \beta^2}}|\mathbf{dx}.$$

The information of other regularization functionals can be found here [1, 3, 4, 15, 17, 18, 21, 23]. Then the problem (1) takes the form, find u that minimize the

$$T(u) = \frac{1}{2}\|\vec{K}u - z\|^2 + \alpha J(u) \tag{2}$$

where J is a regularization functional and $\alpha > 0$ is a regularization parameter. The problem (2) is well-posed [16, 18]. Then the Euler-Lagrange equations of (2) are,

$$\vec{K}^*(\vec{K}u - z) + \alpha L(u)u = \ 0 \text{ in } \Omega, \tag{3}$$

$$\frac{\partial u}{\partial n} = 0 \text{ in } \partial\Omega, \tag{4}$$

where $\vec{K}^*$ represents the adjoint operator of $\vec{K}$. In case of image denoising, $\vec{K}^* = I$ is an identity operator. By Tikhonov regularization, $L(u)u = I(u)u$, where $I(u)$ is the identity operator. By TV regularization,

$$L(u)u = - \bigtriangledown .(\frac{1}{\sqrt{|\bigtriangledown u|^2 + \beta^2}} \bigtriangledown u),$$

where $\beta > 0$ is used to avoid non-differentiability at zero. By mean curvature regularization, $L(u)u$ has the complicated form, i.e.

$$L(u)u = \bigtriangledown.[\frac{\bigtriangledown\kappa}{\sqrt{|\bigtriangledown u|^2 + \beta^2}} - \frac{\bigtriangledown\kappa.\bigtriangledown u}{(\sqrt{|\bigtriangledown u|^2 + \beta^2})^3}\bigtriangledown u]$$

where

$$\kappa(u) = \bigtriangledown.\frac{\bigtriangledown u}{\sqrt{|\bigtriangledown u|^2 + \beta^2}},$$

and we also need $\kappa(u) = 0$ in $\partial\Omega$. In case of TV and MC, (3) is a nonlinear integral differential equation.

3 Primal-Dual Form of MC-Based Model

To avoid confusion between the Euler-Lagrange equations of MC-based image denoising and image deblurring problems, from now to onward, we will treat them as,

$$\vec{\Lambda}^*(\vec{\Lambda}u - z) + \alpha_{MC}\bigtriangledown.[\frac{\bigtriangledown\kappa}{\sqrt{|\bigtriangledown u|^2 + \beta^2}} - \frac{\bigtriangledown\kappa.\bigtriangledown u}{(\sqrt{|\bigtriangledown u|^2 + \beta^2})^3}\bigtriangledown u] = 0 \text{ in } \Omega, \quad (5)$$

$$\frac{\partial u}{\partial n} = 0 \text{ in } \partial\Omega, \quad (6)$$

$$\kappa(u) = 0 \text{ in } \partial\Omega, \quad (7)$$

where

$$\kappa(u) = \bigtriangledown.\frac{\bigtriangledown u}{\sqrt{|\bigtriangledown u|^2 + \beta^2}},$$

and α_{MC} is a parameter related to MC functional J_{MC}. Here if $\Lambda = I$ then (5)-(7) are the Euler-Lagrange equations of image denoising problem and if $\Lambda = K$ then (5)-(7) are the Euler-Lagrange equations of image deblurring problem.

The above equations can be expressed as first order nonlinear system,

$$\vec{\Lambda}^*\vec{\Lambda}u + \alpha_{MC}\bigtriangledown.\vec{p} - \alpha_{MC}\bigtriangledown.\vec{t} = \vec{\Lambda}^* z, \quad (8)$$

$$-w + \bigtriangledown.\vec{v} = 0, \quad (9)$$

$$\sqrt{|\bigtriangledown u|^2 + \beta^2}\,\vec{v} - \bigtriangledown u = 0, \quad (10)$$

$$\sqrt{|\bigtriangledown u|^2 + \beta^2}\,\vec{p} - \bigtriangledown w = 0, \quad (11)$$

$$\sqrt{|\bigtriangledown u|^2 + \beta^2}\,\vec{t} - (\bigtriangledown w.\vec{v})\vec{v} = 0, \quad (12)$$

where

$$\vec{v} = \frac{\bigtriangledown u}{\sqrt{|\bigtriangledown u|^2 + \beta^2}}, w = \bigtriangledown.\vec{v}, \vec{p} = \frac{\bigtriangledown w}{\sqrt{|\bigtriangledown u|^2 + \beta^2}} \text{and } \vec{t} = \frac{(\bigtriangledown w.\vec{v})\vec{v}}{\sqrt{|\bigtriangledown u|^2 + \beta^2}}.$$

After discretization, it is possible to linearize the discrete system by fixing $u = u^m$. The elimination of w, $\overrightarrow{v}$, $\overrightarrow{p}$ and $\overrightarrow{t}$ from (8)-(12), yield the following fixed point iteration scheme,

$$(\vec{\Lambda}^*\vec{\Lambda} + \alpha_{MC} L(u^m))u^{m+1} = \vec{\Lambda}^* z, m = 0, 1, 2, ... \tag{13}$$

where

$$L(u^m)u^{m+1} = \triangledown \cdot [\frac{1}{\sqrt{|\triangledown u^m|^2 + \beta^2}}(\triangledown(\triangledown \cdot \frac{\triangledown u^{m+1}}{\sqrt{|\triangledown u^m|^2 + \beta^2}}))$$
$$- \frac{1}{(\sqrt{|\triangledown u^m|^2 + \beta^2})^3}(\triangledown(\triangledown \cdot \frac{\triangledown u^{m+1}}{\sqrt{|\triangledown u^m|^2 + \beta^2}}). \triangledown u^{m+1}) \triangledown u^{m+1}].$$

3.1 Cell Discretization

The domain $\Omega = (0, 1) \times (0, 1)$ of the problem is partitioned by $\delta_x \times \delta_y$, where

$$\delta_x : 0 = x_{1/2} < x_{3/2} < x_{5/2} < ... < x_{N_x - 1/2} < x_{N_x + 1/2} = 1,$$
$$\delta_y : 0 = y_{1/2} < y_{3/2} < y_{5/2} < ... < y_{N_x - 1/2} < y_{N_x + 1/2} = 1.$$

The N_x is the number of equispaced partitions in the x or y directions. The (x_i, y_j) denotes centers of the cells, where

$$x_i = (i - \frac{1}{2})h \quad i = 1, 2, 3, ..., N_x$$
$$y_j = (j - \frac{1}{2})h \quad j = 1, 2, 3, ..., N_x$$

where $h = \frac{1}{N_x}$. The $(x_{i \pm \frac{1}{2}}, y_j)$ and $(x_i, y_{j \pm \frac{1}{2}})$ are representing midpoints of cell edges, where

$$x_{i \pm \frac{1}{2}} = x_i \pm \frac{h}{2} \quad i = 1, 2, 3, ..., N_x$$
$$y_{j \pm \frac{1}{2}} = y_j \pm \frac{h}{2} \quad j = 1, 2, 3, ..., N_x.$$

The set

$$e_{ij} = \left\{ (x, y) : x \in [x_i - \frac{1}{2}, x_i + \frac{1}{2}], y \in [y_j - \frac{1}{2}, y_j + \frac{1}{2}] \right\}$$

represents a cell with (x_i, y_j) as a center. Let

$$\chi_i(x) = \begin{cases} 1 & x \in (x_i - \frac{1}{2}, x_i + \frac{1}{2}) \\ 0 & \text{otherwise}, \end{cases}$$

$$\chi_j(y) = \begin{cases} 1 & y \in (y_i - \frac{1}{2}, y_i + \frac{1}{2}) \\ 0 & \text{otherwise}. \end{cases}$$

And

$$\phi_i(x_l + \frac{1}{2}) = \delta_{il},$$
$$\phi_k(y_j + \frac{1}{2}) = \delta_{jk}.$$

Approximation of u and w are

$$u(x,y) \cong \overline{U}(x,y) = \sum_{i=1}^{N_x}\sum_{j=1}^{N_x} u_{ij}\chi_i(x)\chi_j(y)$$

and

$$w(x,y) \cong \overline{W}(x,y) = \sum_{i=1}^{N_x}\sum_{j=1}^{N_x} w_{ij}\chi_i(x)\chi_j(y),$$

respectively, where $u_{ij} = \overline{U}(x_i, y_j)$ and $w_{ij} = \overline{W}(x_i, y_j)$. The representation of the data z is

$$z(x,y) \cong \overline{Z}(x,y) = \sum_{i=1}^{N_x}\sum_{j=1}^{N_x} z_{ij}\chi_i(x)\chi_j(y)$$

where z_{ij} can be calculated at cell averages. By applying midpoint quadrature approximation, we have

$$(Ku)(x_i, y_j) \cong [K_h U]_{(ij)}.$$

Denote $\overrightarrow{v} = (v^x, v^y)$, $\overrightarrow{p} = (p^x, p^y)$ and $\overrightarrow{t} = (t^x, t^y)$. The approximation of x and y components of $\overrightarrow{v}$ are

$$\overline{V}^x(x,y) = \sum_{i=1}^{N_x-1}\sum_{j=1}^{N_x} v^x_{ij}\phi_i(x)\chi_j(y) \quad \text{and} \quad \overline{V}^y(x,y) = \sum_{i=1}^{N_x-1}\sum_{j=1}^{N_x} v^y_{ij}\phi_i(y)\chi_j(x)$$

respectively. $\overline{V} = [\overline{V}^x \ \overline{V}^y]^t$ denotes the discretization of $\overrightarrow{v}$. Similarly, approximation of the components of $\overrightarrow{p}$ and $\overrightarrow{t}$ are

$$\overline{P}^x(x,y) = \sum_{i=1}^{N_x-1}\sum_{j=1}^{N_x} p^x_{ij}\phi_i(x)\chi_j(y) \quad , \quad \overline{P}^y(x,y) = \sum_{i=1}^{N_x-1}\sum_{j=1}^{N_x} p^y_{ij}\phi_i(y)\chi_j(x)$$

and

$$\overline{T}^x(x,y) = \sum_{i=1}^{N_x-1}\sum_{j=1}^{N_x} t^x_{ij}\phi_i(x)\chi_j(y) \quad , \quad \overline{T}^y(x,y) = \sum_{i=1}^{N_x-1}\sum_{j=1}^{N_x} t^y_{ij}\phi_i(y)\chi_j(x)$$

respectively. The $\overline{P} = [\overline{P}^x \ \overline{P}^y]^t$ and $\overline{T} = [\overline{T}^x \ \overline{T}^y]^t$ denote the discretization of the vectors $\overrightarrow{p}$ and $\overrightarrow{t}$ respectively.

3.2 The Cell-Centered Finite Difference Method

Here, we consider the cell-centered finite difference (CCFD) method for MC-based image denoising and image deblurring problems. With lexicographical ordering of the unknowns,

$$U = [\overline{U}_{11} \quad \overline{U}_{12} \quad ... \quad \overline{U}_{N_xN_x}]^t, \quad W = [\overline{W}_{11} \quad \overline{W}_{12} \quad ... \quad \overline{W}_{N_xN_x}]^t,$$

$$V = [\overline{V}^x_{11} \quad \overline{V}^x_{12} \quad ... \quad \overline{V}^x_{N_x-1N_x-1} \quad ... \quad \overline{V}^y_{11} \quad \overline{V}^y_{12} \quad ... \quad \overline{V}^y_{N_x-1N_x-1}]^t,$$

$$P = [\overline{P}^x_{11} \quad \overline{P}^x_{12} \quad ... \quad \overline{P}^x_{N_x-1N_x-1} \quad ... \quad \overline{P}^y_{11} \quad \overline{P}^y_{12} \quad ... \quad \overline{P}^y_{N_x-1N_x-1}]^t,$$

and

$$T = [\overline{T}^x_{11} \quad \overline{T}^x_{12} \quad ... \quad \overline{T}^x_{N_x-1N_x-1} \quad ... \quad \overline{T}^y_{11} \quad \overline{T}^y_{12} \quad ... \quad \overline{T}^y_{N_x-1N_x-1}]^t.$$

Now by applying CCFD method to (8)-(12) together with midpoint quadrature for the integral term one obtains the following system,

$$\Lambda_h^*\Lambda_h U - \alpha_{MC}A_h W + \alpha_{MC}B_h^* P - \alpha_{MC}B_h^* T = \Lambda_h^* Z, \quad (14)$$

$$-I_h W + B_h^* V = O, \quad (15)$$

$$D_h V + B_h U = O, \quad (16)$$

$$D_h P + B_h W = O, \quad (17)$$

$$D_h T - C_h V = O. \quad (18)$$

Here Λ_h, A_h and I_h are matrices of size $N_x^2 \times N_x^2$. The B_h is of size $2N_x(N_x - 1) \times N_x^2$. The C_h and D_h are matrices of size $2N_x(N_x-1) \times 2N_x(N_x-1)$. So we have the following system

$$\begin{bmatrix} \Lambda_h^*\Lambda_h & -\alpha_{MC}A_h & O & \alpha_{MC}B_h^* & -\alpha_{MC}B_h^* \\ O & -I_h & B_h^* & O & O \\ B_h & O & D_h & O & O \\ O & B_h & O & D_h & O \\ O & O & -C_h & O & D_h \end{bmatrix} \begin{bmatrix} U \\ W \\ V \\ P \\ T \end{bmatrix} = \begin{bmatrix} \Lambda_h^* Z \\ O \\ O \\ O \\ O \end{bmatrix}$$

For image denosing problem, the matrix $\Lambda_h = I_h$ is the identity matrix. For image deblurring problem, the matrix $\Lambda_h^*\Lambda_h = K_h^* K_h$ is symmetric positive semidefinite matrix. The matrix K_h is block Toeplitz with Toeplitz blocks (BTTB) matrix.

$$K_h = h \begin{bmatrix} k(0) & k(-h) & \dots & k(-(n-1)h) \\ k(h) & k(0) & \dots & k(-(n-2)h) \\ & & \ddots & \\ k((n-1)h) & k((n-2)h) & \dots & k(0) \end{bmatrix}$$

The matrix A_h is a diagonal matrix having following structure,

$$A_h = \frac{2}{\beta h}(A_1 + A_2),$$

where both A_1 and A_2 are of size $N_x^2 \times N_x^2$.

$$A_1 = \tilde{I} \otimes E \quad \text{and} \quad A_2 = E \otimes \tilde{I}$$

where $\otimes$ is a tensor product. The identity matrix $\tilde{I}$ and E are of size $N_x \times N_x$.

$$E = \begin{bmatrix} 1 & & & & & \\ & 0 & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & 0 & \\ & & & & & 1 \end{bmatrix}.$$

The matrix B_h has the following structure,

$$B_h = \frac{1}{h}\begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$$

where both B_1 and B_2 are of size $N_x(N_x-1)\times N_x^2$.

$$B_1 = F\otimes \tilde{I} \quad \text{and} \quad B_2 = \tilde{I}\otimes F$$

where F is of size $(N_x-1)\times N_x$.

$$F = \begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & -1 & \\ & & & & 1 & -1 \end{bmatrix}.$$

The matrix C_h is a diagonal matrix and its entries are obtained by the discretization of the expression $(\bigtriangledown w.\overrightarrow{v})$.

$$C_h = \begin{bmatrix} C^x & 0 \\ 0 & C^y \end{bmatrix}$$

where the size of C^x is $(N_x-1)\times N_x$ and the size of C^y is $N_x\times(N_x-1)$. The matrix D_h is also a diagonal matrix with positive diagonal entries and the diagonal entries are obtained by the discretization of the expression $\sqrt{|\bigtriangledown u|^2+\beta^2}$.

$$D_h = \begin{bmatrix} D^x & 0 \\ 0 & D^y \end{bmatrix}$$

where the size of D^x is $(N_x-1)\times N_x$ and the size of D^y is $N_x\times(N_x-1)$. Note that on horizontal and vertical edges of each cell e_{ij}, the values of the all unknowns are not available, so average operators can be used to approximate their values.

Now if we eliminate W, V, P and T from (14)-(18), then we have the following primal system,

$$(\Lambda_h^*\Lambda_h + \alpha_{MC}L_h(U))U = \Lambda_h^* Z, \tag{19}$$

where

$$L_h = (B_h^* D_h^{-1} B_h)^2 + A_h(B_h^* D_h^{-1} B_h) + B_h^* D_h^{-1} C_h D_h^{-1} B_h. \tag{20}$$

The first and the last term in L_h is symmetric positive semidefinite [18] but the middle term is not symmetric. By lexicographical ordering of the unknowns, L_h is block pentadiagonal. The diagonal blocks are pentadiagonal matrices and the off-diagonal blocks, just below and above the main diagonal blocks, are tridiagonal matrices. The remaining blocks are diagonal matrices. Similar primal forms exist for TV [16, 18] and Tikhonov [1, 17] based models. For TV , we have

$$(\Lambda_h^*\Lambda_h + \alpha_{TV} B_h^* D_h^{-1} B_h(U))U = \Lambda_h^* Z, \tag{21}$$

and for Tikhonov, we have

$$(\Lambda_h^* \Lambda_h + \alpha_{Tik} I_h(U))U = \Lambda_h^* Z, \tag{22}$$

where $B_h^* D_h^{-1} B_h$ and I_h are the matrices arise from the discretization of J_{TV} and J_{Tik} respectively.

In the literature, one can find many numerical techniques which have been applied to mean curvature-based nonlinear minimization image denoising and image deblurring problems. Among them, Augmented Lagrangian method [23], Time Marching scheme [22], Conjugate Gradient (CG) method [18] and Multigrid method [2], etc. However, these numerical techniques get quite slow convergence due to ill-conditioned and large nonlinear system. Moreover the use of MC functional in the image denoising and image deblurring problems adds more complexity to the nonlinear systems. MC regularization functional is too much computationally expensive, that is why, existing numerical methods performs very poorly.

4 One-Level Method

Here, we will introduce an algorithm to solve primal form (19) of MC-based nonlinear image denoising and image deblurring problems. We call this procedure One-Level method. First we apply discrete version of the fixed point iteration to (19). So we have a following linear system;

$$(\Lambda_h^* \Lambda_h + \alpha_{MC} L_h(U^m))U^{m+1} = \Lambda_h^* Z. \tag{23}$$

The properties of our system (19), mentioned in the previous section, suggest that iterative method like Generalized Minimal Residual (GMRES) method, is suitable for (23). Unfortunately, GMRES method can get quite slow convergence rate due to ill-conditioned system. One remedy for this problem is preconditioning. That is, we have to use Preconditioned Generalized Minimal Residual (PGMRES) method. In order to make PGMRES method effective, preconditioning matrix P, must be symmetric positive definite. For this , we have used the following simple preconditioning matrix P,

$$P = I + \alpha_{MC} diag(L), \tag{24}$$

where I is an identity matrix and $diag(L)$ is a diagonal matrix whose entries are the diagonal entries of matrix L. While applying PGMRES method to (23), the inversion of P, will be required. Since our preconditioning matrix P, is a diagonal matrix, so inversion can be done easily. Rapid convergence has shown in PGMRES method due to our preconditioning matrix P, in the numerical results below. We summarized the One-Level method, in Algorithm 1.

To make a more robust numerical method for MC-based nonlinear image denoising and image deblurring problems, now we present a Two-Level method.

5 Two-Level Method

The Two-Level method consists of solving two different problems at two different levels. At Level-I (coarse mesh) we solve a nonlinear integral differential

On mesh Ω_h ,
Initial iteration U^0,
for $m = 1; max$

$$A^m = \Lambda_h^* \Lambda_h + \alpha_{MC} L_h(U^m),$$

$$b^m = \Lambda_h^* Z,$$

Use PGMRES method to solve

$$A^m U^{m+1} = b^m,$$

with

$$P_h = I_h + \alpha_{MC} diag(L_h(U^m)),$$

end
The L_h is a matrix arise from the discretization of MC regularization functional J_{MC} and α_{MC} is a parameter related to J_{MC}.

Algorithm 1: The One-Level Method

equation (image denoising and image deblurring). At this level, the regularization functional is mean curvature functional J_{MC}. It means we are solving computationally expensive problem just on a coarse mesh at Level-I. Then we interpolate our solution for Level-II. For better results one can use spline interpolation. Then at Level-II (fine mesh), we solve a linear integral differential equation (image denoising and image deblurring). At Level-II, we will use less computationally expensive regularization functional, that is, instead of using MC we will use either TV or Tikhonov. It means at Level-II, we have to do less expensive work. Because expensive part of the work is already done at Level-I on a coarse mesh of a small size. Moreover, at Level-II, the mean curvature term is approximated using the coarse solution and kept in the right hand side. The effectiveness of Two-Level method is shown in the section of numerical results. The Two-Level method is summarized in Algorithm 2. To simplify notation,we drop the superscript representing fixed point iteration count.

Step 1. Solve the problem (19);

$$(\Lambda_H^* \Lambda_H + \alpha_{MC} L_H(U_H))U_H = \Lambda_H^* Z \quad (25)$$

for U_H on a coarse mesh Ω_H by using Algorithm 1. The L_H is a matrix arise from the discretization of MC regularization functional J_{MC} and α_{MC} is a parameter related to J_{MC}.

Step 2. Obtain U_{Hh} by interpolating U_H on a fine mesh Ω_h.

Step 3. Solve the following problem;

$$(\Lambda_h^* \Lambda_h + \widetilde{\alpha} \widetilde{L}_h(U_{Hh}))U_h = \Lambda_h^* Z - \alpha_{MC} L_h(U_{Hh})U_{Hh} + \widetilde{\alpha} \widetilde{L}_h(U_{Hh})U_{Hh} \quad (26)$$

for U_h on a fine mesh Ω_h. The $\widetilde{L}_h$ is a matrix arise from the discretization of regularization functional J (Tikhonov or TV) and $\widetilde{\alpha}$ is a parameter related to J. The relation between coarse mesh size and fine mesh size is $h = 2H$.

Algorithm 2: The Two-Level Method

5.1 Parameter Selection Procedure

To gain a quality image, the optimum value of the parameters also play a vital role in image denoising and image deblurring problems. So for optimal accuracy in the numerical results, we also need optimum values of α_{MC} of Level-I and $\widetilde{\alpha}$ of Level-II. For optimum value of α_{MC} we refer the reader to [3, 22, 23].

In order to obtain an optimal $\widetilde{\alpha}$, we start by assuming that Two-Level solution U_h is close to the One-Level solution U_h^1 on Ω_h. That is

$$U_h \cong U_h^1.$$

Then from (26), we have

$$\begin{aligned}
&(\Lambda_h^*\Lambda_h + \widetilde{\alpha}\widetilde{L}_h(U_{Hh}))^{-1}(\Lambda_h^*Z - \alpha_{MC}L_h(U_{Hh})U_{Hh} + \widetilde{\alpha}\widetilde{L}_h(U_{Hh})U_{Hh}) \cong U_h^1 \\
&\Rightarrow \Lambda_h^*Z - \alpha_{MC}L_h(U_{Hh})U_{Hh} + \widetilde{\alpha}\widetilde{L}_h(U_{Hh})U_{Hh} \cong (\Lambda_h^*\Lambda_h + \widetilde{\alpha}\widetilde{L}_h(U_{Hh}))U_h^1 \\
&\Rightarrow \Lambda_h^*Z - \alpha_{MC}L_h(U_{Hh})U_{Hh} + \widetilde{\alpha}\widetilde{L}_h(U_{Hh})U_{Hh} \cong \Lambda_h^*\Lambda_h U_h^1 + \widetilde{\alpha}\widetilde{L}_h(U_{Hh})U_h^1 \\
&\Rightarrow \Lambda_h^*Z - \alpha_{MC}L_h(U_{Hh})U_{Hh} - \Lambda_h^*\Lambda_h U_h^1 \cong \widetilde{\alpha}\widetilde{L}_h(U_{Hh})(U_h^1 - U_{Hh}). \qquad (27)
\end{aligned}$$

Now take a norm on both sides, so

$$\|\Lambda_h^*Z - \alpha_{MC}L_h(U_{Hh})U_{Hh} - \Lambda_h^*\Lambda_h U_h^1\| \cong \widetilde{\alpha}\|\widetilde{L}_h(U_{Hh})(U_h^1 - U_{Hh})\|.$$

So we have

$$\widetilde{\alpha} \cong \frac{\|\Lambda_h^*Z - \alpha_{MC}L_h(U_{Hh})U_{Hh} - \Lambda_h^*\Lambda_h U_h^1\|}{\|\widetilde{L}_h(U_{Hh})(U_h^1 - U_{Hh})\|}. \qquad (28)$$

In the above formula, practically it is not possible to use U_h^1 and $(U_h^1 - U_{Hh})$. So their approximate values can be used. In this paper, we are also using their approximating values. That procedure is summarized in Algorithm 3.

Step 1. Calculate the residual r_h on a fine mesh Ω_h

$$r_h = \Lambda_h^*Z - (\Lambda_h^*\Lambda_h + \alpha_{MC}L_h(U_{Hh}))U_{Hh}.$$

Step 2. On the coarse mesh Ω_H , restrict r_h for r_H by using interpolation.
Step 3. Solve the error equation for E_H on the coarse mesh Ω_H

$$(\Lambda_H^*\Lambda_H + \alpha_{MC}L_H(U_H))E_H = r_H.$$

Step 4. Calculate $\widetilde{\alpha}$ by replacing U_h^1 by U_{Hh} and $(U_h^1 - U_{Hh})$ by E_H in formula (28);

$$\widetilde{\alpha} \cong \frac{\|\Lambda_h^*Z - \alpha_{MC}L_h(U_{Hh})U_{Hh} - \Lambda_h^*\Lambda_h U_{Hh}\|}{\|\widetilde{L}_H(U_H)E_H\|}. \qquad (29)$$

Algorithm 3: Algorithm for $\widetilde{\alpha}$

6 Numerical Results

Here, we include numerical results obtained by Two-Level method. We are presenting six examples. In Example 1, Example 2 and Example 4, we have applied Two-Level method to image deblurring problem. In Example 3, Example 5 and Example 6, we have applied Two-Level method to image denoising problem. In all examples, the Level-I calculation were obtained with $\frac{N_x}{2} \times \frac{N_x}{2}$ grid points on Ω_H and Level-II calculation were obtained with $N_x \times N_x$ grid points on Ω_h. So $h = 2H$. In all experiment, we have taken different N_x and the resulting system has N_x^2 unknowns. The optimum value of α_{MC} is used according to [3, 22, 23] and the optimum values of $\widetilde{\alpha}$ are calculated by Algorithm 3. In all examples, for the stopping criteria of a numerical method we have used tolerance $tol = 1e-7$.

For numerical computations, we have used MATLAB software and numerical results are obtained using a Intel(R) Core(TM) i7-4510U CPU @ 2.00 GHz 2.60 GHz. All the results based on the data given in the images are analyzed and presented in the tabular form.

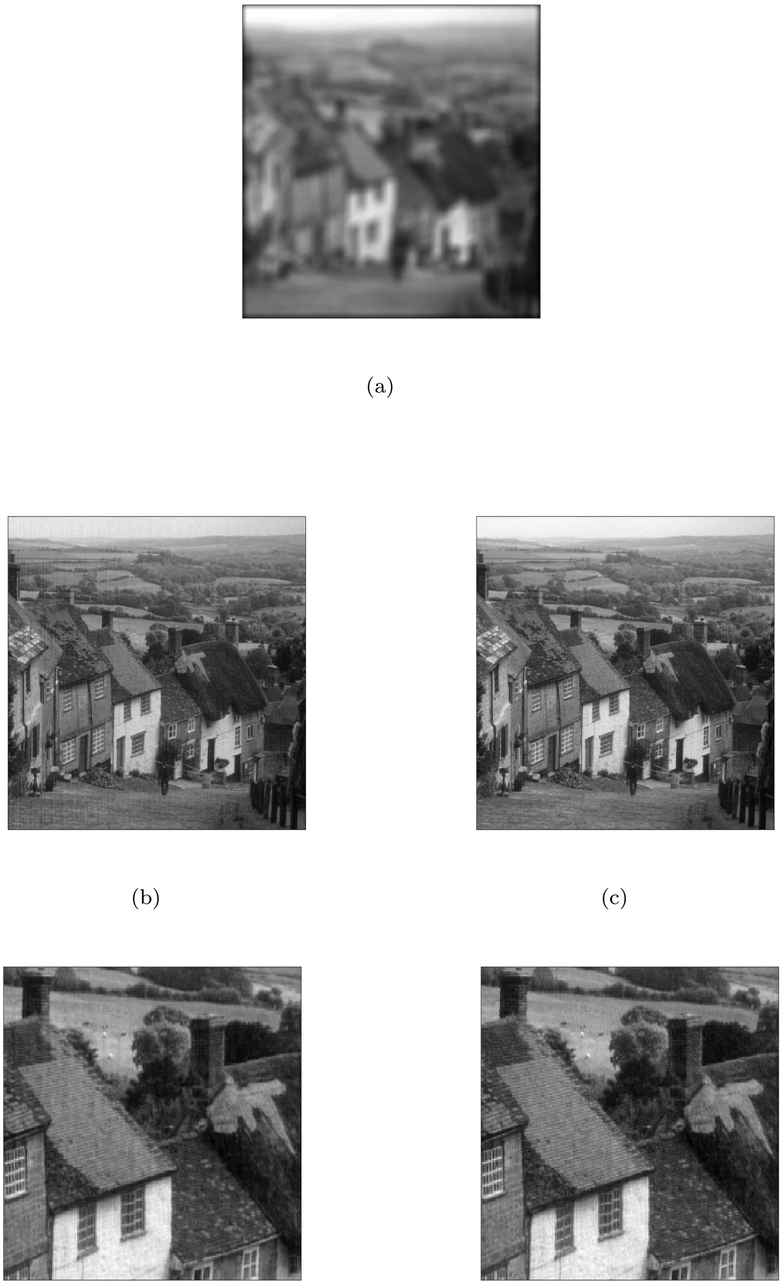
Example 1

This example show the application of Two-Level method for image deblurring problem. Here, we have compared the results of Two-Level method with One-Level method. In this experiment Goldhill image is used. The different aspects of Goldhill image have shown in Figure 1. The size of each sub-figure is 512×512. These are (a) Blurry image (b) Deblurred image by One-Level method (c) Deblurred image by Two-Level method (d) Local region deblurred by One-Level method and (e) Local region deblurred by Two-Level method. For numerical calculations, we have used the $ke\_gen(N, 300, 10)$ kernel. The parameters $\beta = 0.1$ and $\alpha_{MC} = 1e-8$.

In Two-Level method, at Level-II, we have solved the problem (26) with CG (Conjugate Gradient) method. In (26), $\widetilde{L}_h$ is a symmetric positive semidefinite [18] matrix arise from the discretization of J_{TV} (TV regularization functional) and $\widetilde{\alpha} = \alpha_{TV}$. To measure the quality of the restored images, we have used PSNR (Peak Signal to Noise Ratio)[18]. The higher value of PSNR indicate better quality. In this experiment, we have taken three values of N_x. These are $128, 256$ and 512. The corresponding blurry images PSNR are $22.9784, 22.2335$ and 21.4633 respectively. For convergence rate we have used the following formula;

$$rate = log_{10}(\frac{e_{2N_x}}{e_{N_x}})/log_{10}(\frac{2N_x}{N_x}),$$

where e_{N_x} and e_{2N_x} denote the errors of u at N_x and $2N_x$ respectively, in discrete norm. In Table 1, we have summarized all the information of this experiment.



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

Figure 1: Goldhill Image: (a) Blurry image (b) Deblurred image by One-Level method (c) Deblurred image by Two-Level method (d) Local region deblurred by One-Level method and (e) Local region deblurred by Two-Level method.

		One-Level Method			Two-Level Method			
N_x	h	Deblurred PSNR	CPU-Time	*rate*	$\widetilde{\alpha}$	Deblurred PSNR	CPU-Time	*rate*
128	$\frac{1}{128}$	40.2928	10.4733		0.0134	40.1985	4.2049	
				1.0775				1.0255
256	$\frac{1}{256}$	37.7598	58.6002		0.0366	37.5116	21.3622	
				1.1691				1.0001
512	$\frac{1}{512}$	35.0239	415.0476		0.0845	34.8626	136.3840	

Table 1: One-Level Method vs Two-Level Method

Remarks

1. The Table 1 shows that the CPU-Time by Two-Level method is less than the CPU-Time by One-Level method for all values of N_x. For $N_x = 128$ and $N_x = 256$ we save more than 60%. For $N_x = 512$ we save more than 70%. For $N_x = 512$, in One-Level method we have to solve a nonlinear MC-based system of size 512^2 equations. While for the same $N_x = 512$, in Two-Level method, we first solve a nonlinear MC-based system of size 256^2 equations at coarse mesh and then solve a linear TV-based system of size 512^2 equations at fine mesh. This is why, we are saving time in Two-Level method. We expect the increase in savings as the mesh size decreases.

2. From Figure 1(b) and Figure 1(c), one can notice the quality of deblurred images produce by both methods. Both images are almost similar and most of the blurry has been removed. This can also be seen in the local images (Figure 1(d) and Figure 1(e)) deblurred by both methods. So deblurred image quality by the Two-Level method is the same as by the One-Level method.

3. The Table 1 shows that for all values of N_x the PSNR for both methods are almost the same. This means that Two-Level method generates same quality in less CPU-Time.

4. From the Table 1, it is clear that the convergence rate of Two-Level method is almost same like One-Level method. For both methods convergence rate is approximately 1.

Example 2

This example also show the application of Two-Level method for image deblurring problem. For this experiment the Cameraman image is used. The different aspects of Cameraman image can be shown in Figure 2, each one is having size 512×512. These are (a) Blurry image (b) Deblurred image by One-Level method (c) Deblurred image by Two-Level method (d) Local region deblurred by One-Level method and (e) Local region deblurred by Two-Level method. This example is different from Example 1. Because here in Two-Level method at Level-II, we have used Tikhonov regularization functional instead of TV regularization functional. So here $\widetilde{\alpha} = \alpha_{Tik}$. All other parameters are same like the Example 1. The $21.4242, 21.2064$ and 21.1871 are the blurry PSNR against the mesh size $\frac{1}{128}, \frac{1}{256}$ and $\frac{1}{512}$ respectively. In Table 2, we have summarized all the information of this experiment.

Remark

The Table 2 clearly shows that the Two-Level method is significantly reducing the cost of time for all value of mesh size. The Two-Level method also producing the almost same quality in the deblurred images (see Figure 2). So we have no doubt in saying that in comparison the Two-Level method is faster than One-Level method for MC-based image deblurring Problem.

		One-Level Method		Two-Level Method		
N_x	h	Deblurred PSNR	CPU-Time	$\widetilde{\alpha}$	Deblurred PSNR	CPU-Time
128	$\frac{1}{128}$	41.7343	10.5387	6.9020e-05	41.3578	6.5197
256	$\frac{1}{256}$	42.6843	52.5101	3.3754e-05	42.5413	20.7124
512	$\frac{1}{512}$	43.2529	328.1486	9.5601e-05	43.3116	122.1332

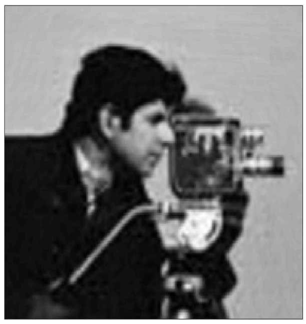
Table 2: One-Level Method vs Two-Level Method

(a)

(b)

(c)

(d)



(e)

Figure 2: Cameraman Image: (a) Blurry image (b) Deblurred image by One-Level method (c) Deblurred image by Two-Level method (d) Local region deblurred by One-Level method and (e) Local region deblurred by Two-Level method.

Example 3

In this example we have used Two-Level method for image denoising problem. Here we have presented comparison between our proposed Two-Level method and Multigrid method by C. Brito-Loeza and K. Chen [2]. For this we have used Brain image. The different aspects of Brain image have shown in Figure 3. The size of each sub-figure is 512×512. These are (a) Noisy image (b) Denoised image by Multigrid method (c) Denoised image by Two-Level method (d) Local region denoised by Multigrid method and (e) Local region denoised by Two-Level method.

In Multigrid method we have used MC regularization functional with optimum parameters [2] $\alpha_{MC} = \frac{1}{200}$ and $\beta = 1e-2$. In Two-Level method, at Level-I, we have solved the problem with Algorithm 1 with optimum parameters $\beta = 0.2$ and α_{MC} varies from $1e-1$ to $1e-3$. At Level-II, we have solved the problem (26) with CG (Conjugate Gradient) method. In (26), $\widetilde{L}_h$ is a matrix arise from the discretization of TV regularization functional J_{TV} and $\widetilde{\alpha} = \alpha_{TV}$. For comparison we have taken three values of N_x. These are $128, 256$ and 512. In this experiment, we have used Gaussian noise. The added Gaussian noise is large so that all of the noisy images have SNR $= 3.5$ (Signal to Noise Ratio) [2]. In Table 3, we have summarized all the information of this experiment.

Remarks

1. The Table 3 shows that the CPU-Time by Two-Level method is less than the CPU-Time by Multigrid method for all values of N_x. For $N_x = 128$ we save more than 30% of CPU-Time. For $N_x = 256$ we save almost 50% of CPU-Time. For $N_x = 512$ we save more than 70% of CPU-Time.

2. From Figure 3(b) and Figure 3(c), one can notice the quality of denoised images produce by both methods. Both images are almost similar and most of the noise has been removed. This can also be seen in the local images (Figure 3(d) and Figure 3(e)) denoised by both methods. This means that Two-Level method generates same quality in less CPU-Time. So in comparison the Two-Level method is faster than Multigrid method for MC-based image denoising Problem.

		Multigrid Method		Two-Level Method
N_x	h	CPU-Time	$\widetilde{\alpha}$	CPU-Time
128	$\frac{1}{128}$	10.9399	30.0108	7.5182
256	$\frac{1}{256}$	21.9365	10.0103	11.1275
512	$\frac{1}{512}$	81.7316	1.0090	23.4819

Table 3: Multigrid Method vs Two-Level Method

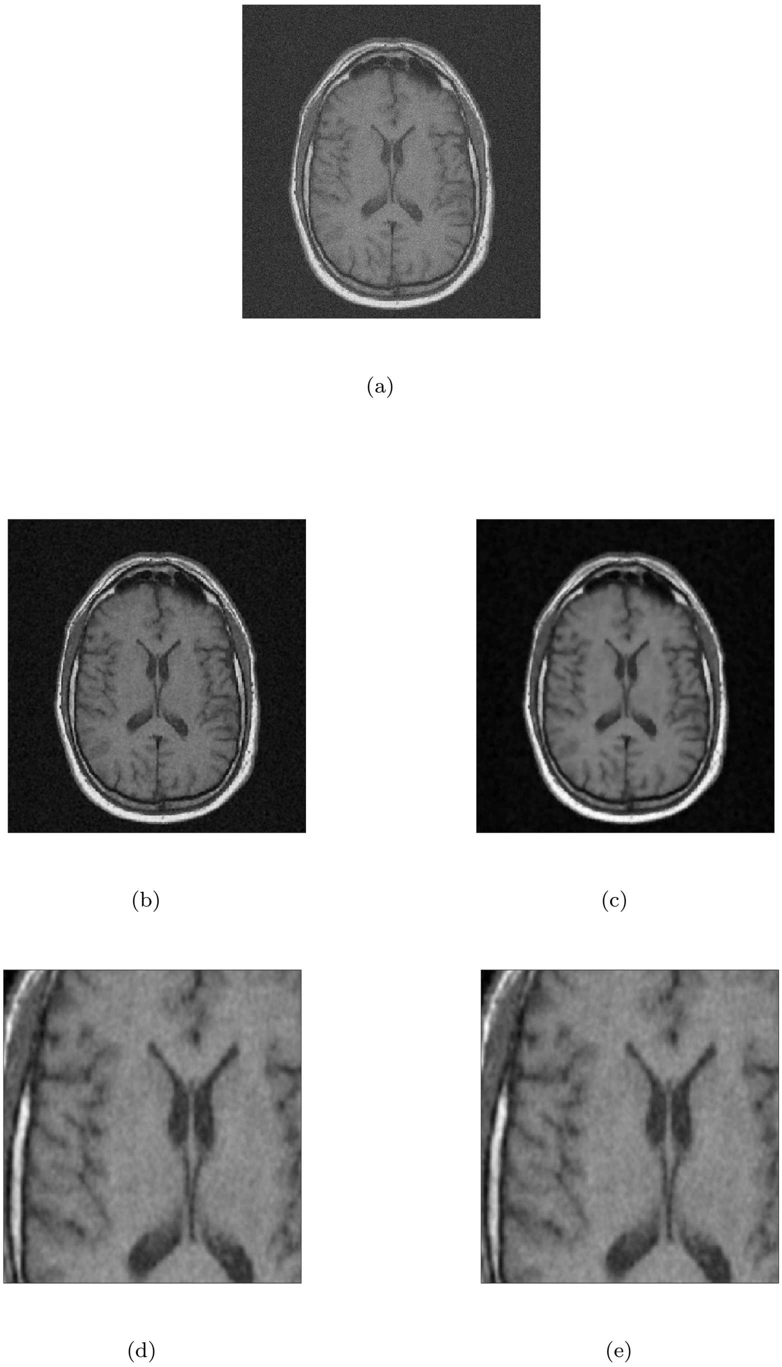


Figure 3: Brain Image: (a) Noisy image (b) Denoised image by Multigrid Method (c) Denoised image by Two-Level method (d) Local region denoised by Multigrid Method and (e) Local region denoised by Two-Level method.

Example 4

In this example we have consider different kinds of images. These are Peppers, Kids and Moon images. The Peppers image is a nontexture image. The Kids image is a complicated image, as it contains a large scale cartoon part (face) and also a small scale texture part (shirt). The Moon image is real and synthetic image. The different aspects of each image has shown in Figure 4. The size of each sub-figure is 512×512. (a), (d) and (g) are blurry images. (b), (e) and (h) are deblurred images by One-Level method. (c), (f) and (i) are deblurred images by Two-Level method. This example also show the application of Two-Level method for image deblurring problem. Here, we have also compared the results of Two-Level method with One-Level method. For numerical calculations, we have used the $ke\_gen(N, 300, 10)$ kernel. The parameters β varies from $1e-2$ to 1 and α_{MC} varies from $1e-6$ to $1e-12$.

In Two-Level method, at Level-II, we have solved the problem (26) with CG (Conjugate Gradient) method. In (26), $\widetilde{L}_h$ is a symmetric positive semidefinite [18] matrix arise from the discretization of J_{TV} (TV regularization functional) and $\widetilde{\alpha} = \alpha_{TV}$. To measure the quality of the restored images, we have used PSNR and SSIM (Structural Similarity Index Measure). The SSIM value close to 1 indicates that compared images have the almost same quality. In SSIM calculation, we have used exact image as a reference image. The blurry PSNR of Peppers, Kids and Moon images are $20.3400, 20.2170$ and 26.6641 respectively. All the information of this experiment is summarized in Table 4.

Remark

The Table 4 clearly shows that the Two-Level method is significantly reducing the cost of time for all images. The Two-Level method is generating the almost same PSNR and same SSIM in less CPU-Time as compared with One-Level method. The Two-Level method also producing the almost same quality in the deblurred images (see Figure 4). So Two-Level method is more efficient algorithm for solving the mean curvature model.

	One-Level Method			Two-Level Method		
Image	Deblurred PSNR	SSIM	CPU-Time	Deblurred PSNR	SSIM	CPU-Time
Peppers	42.1928	0.9789	210.0256	40.7792	0.9717	129.3050
Kids	43.7508	0.8348	409.3115	43.1272	0.8270	317.6124
Moon	51.9372	0.9069	279.1825	51.6423	0.9024	223.8510

Table 4: One-Level Method vs Two-Level Method

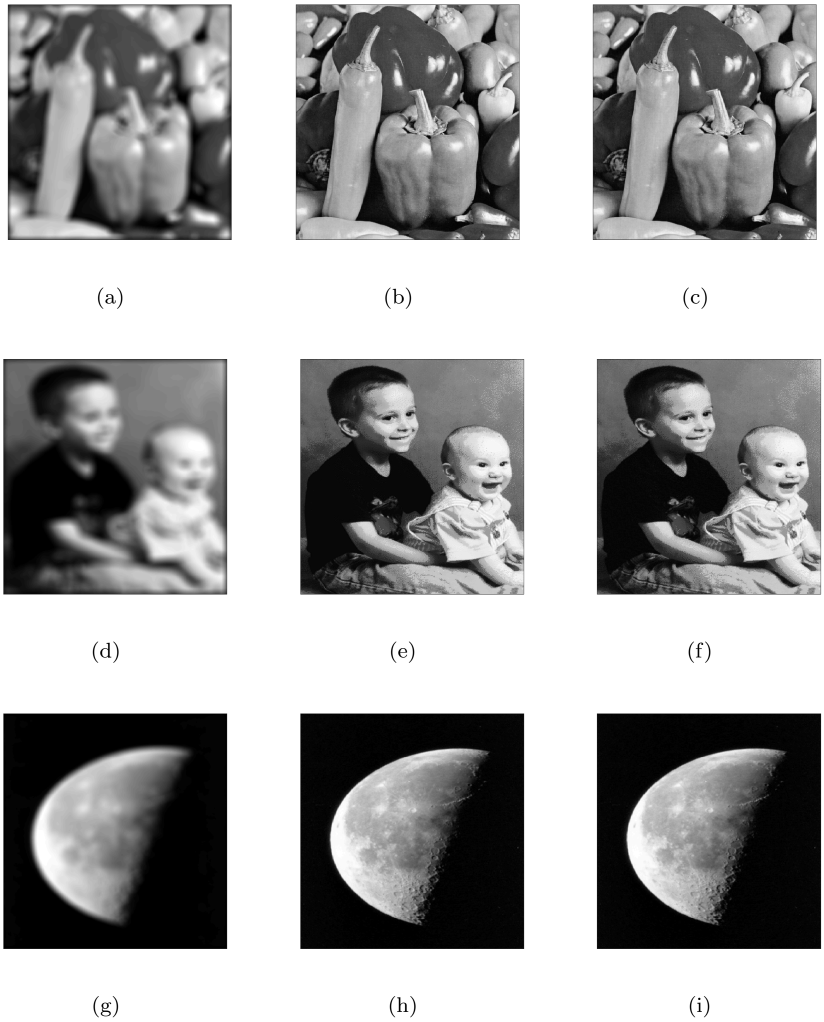


Figure 4: Peppers Image: (a) Blurry image (b) Deblurred image by One-Level method and (c) Deblurred image by Two-Level method. Kids Image: (d) Blurry image (e) Deblurred image by One-Level method and (f) Deblurred image by Two-Level method. Moon Image: (g) Blurry image (h) Deblurred image by One-Level method and (i) Deblurred image by Two-Level method.

Example 5

In this example we have used Two-Level method for image denoising problem. Here we have used different levels of Gaussian noise and have presented a comparison between our proposed Two-Level method and One-Level method. For this we have used Peppers, Cameraman and Rice images. The different aspects of the images have shown in Figure 5. The size of each sub-figure is 512×512. (a), (d) and (g) are noisy images. (b), (e) and (h) are denoised images by One-Level method. (c), (f) and (i) are denoised images by Two-Level method. The parameters $\beta = 0.9$ and α_{MC} varies from $1e-1$ to $5e+1$.

In Two-Level method, at Level-II, we have solved the problem (26) with CG (Conjugate Gradient) method. The $\widetilde{L}_h$ is a matrix arise from the discretization of TV regularization functional J_{TV} and $\widetilde{\alpha} = \alpha_{TV}$. To measure the quality of the restored images, we have used SSIM (Structural Similarity Index Measure). We have done SSIM calculation, between the restored images by both methods. In Table 5, we have summarized all the information of this experiment.

Remarks

1. The Table 5 shows that the CPU-Time by Two-Level method is less than the CPU-Time by One-Level method for all values of N_x. From Figure 5, one can notice the quality of denoised images produce by both methods. Both images are almost similar and most of the noise has been removed. This can also be seen with SSIM calculations in Table 5. The SSIM value is almost close to one for all images. This means that Two-Level method generates same quality in less CPU-Time. So in comparison the Two-Level method is faster than One-Level method for MC-based image denoising Problem.

Image	Gaussian Noise Mean (μ), Variance (σ^2)	One-Level Method CPU-Time	Two-Level Method CPU-Time	SSIM
Peppers	$\mu = 0.1$, $\sigma^2 = 0.015$	36.2887	21.1417	0.9715
Cameraman	$\mu = 0.1$, $\sigma^2 = 0.1$	39.3454	25.0652	0.9935
Rice	$\mu = 0.2$, $\sigma^2 = 0.01$	33.1571	20.5771	0.9402

Table 5: One-Level Method vs Two-Level Method

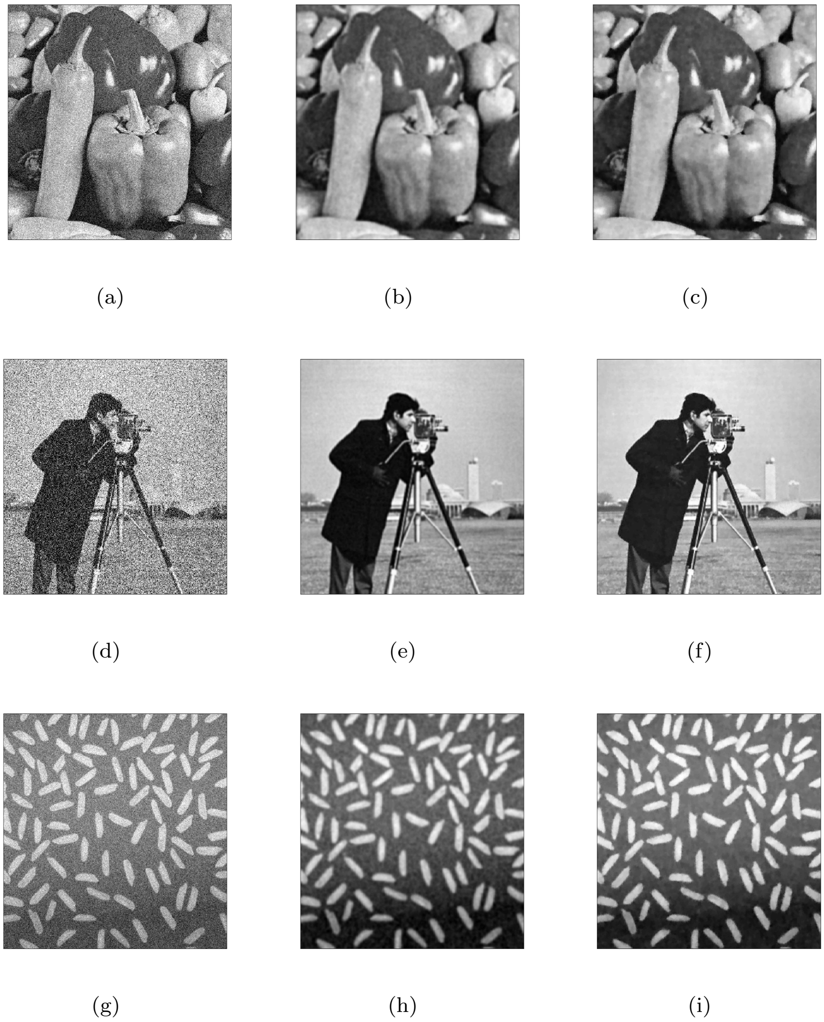


Figure 5: Peppers Image: (a) Noisy image (b) Denoised image by One-Level method and (c) Denoised image by Two-Level method. Cameraman Image: (d) Noisy image (e) Denoised image by One-Level method and (f) Denoised image by Two-Level method. Rice Image: (g) Noisy image (h) Denoised image by One-Level method and (i) Denoised image by Two-Level method.

Example 6

In this example we have used different types of noises (Poisson, Random and Speckle) and have presented a comparison between our proposed Two-Level method and One-Level method. Here we have also applied Two-Level method to image denoising problem. For this we have used Kids, Brain and Moon images. The different aspects of images have shown in Figure 6. The size of each sub-figure is 512×512. (a), (d) and (g) are noisy images. (b), (e) and (h) are denoised images by One-Level method. (c), (f) and (i) are denoised images by Two-Level method. The parameters $\beta = 0.9$ and α_{MC} varies from $1e-1$ to

$5e + 1$.

In Two-Level method, at Level-II, we have solved the problem (26) with CG (Conjugate Gradient) method. The $\widetilde{L}_h$ is a matrix arise from the discretization of TV regularization functional J_{TV} and $\widetilde{\alpha} = \alpha_{TV}$. To measure the quality of the restored images, we have used SSIM (Structural Similarity Index Measure). We have done SSIM calculation, between the restored images by both methods. In Table 6, we have summarized all the information of this experiment.

Remark

The Table 6 clearly shows that the Two-Level method is significantly reducing the cost of time for all images. The Two-Level method is generating the same image quality (see Figure 6) in less CPU-Time as compared with One-Level method. This can also be seen with SSIM calculations in Table 6. So Two-Level method is robust and more efficient algorithm for solving the mean curvature model.

Noise	Image	One-Level Method CPU-Time	Two-Level Method CPU-Time	SSIM
Poisson	Kids	41.3401	24.8889	0.9459
Random	Brain	30.0051	19.6381	0.9875
Speckle	Moon	35.7097	23.6052	0.9699

Table 6: One-Level Method vs Two-Level Method

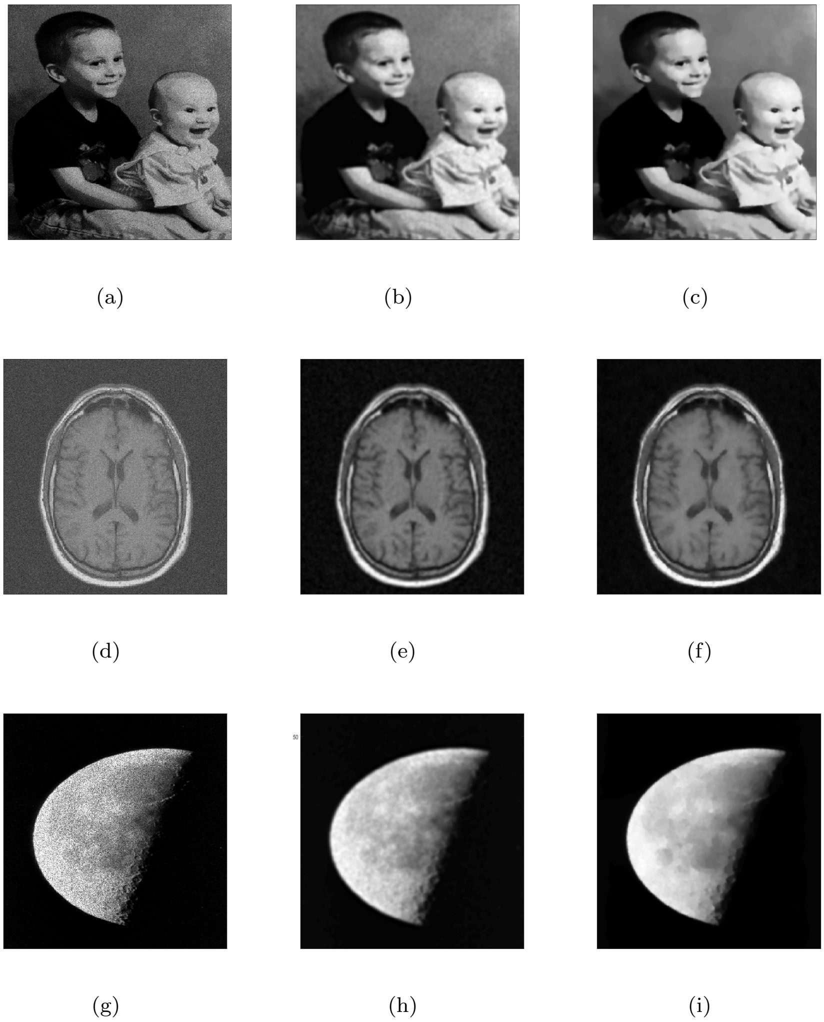


Figure 6: Kids Image: (a) Noisy image (b) Denoised image by One-Level method and (c) Denoised image by Two-Level method. Brain Image: (d) Noisy image(e) Denoised image by One-Level method and (f) Denoised image by Two-Level method. Moon Image: (g) Noisy image (h) Denoised image by One-Level method and (i) Denoised image by Two-Level method.

7 Conclusion

A Two-Level method for mean curvature-based image denoising and image deblurring problems is discussed. Six examples are tested using our technique. In Example 1 and Example 2, we have applied Two-Level method to image deblurring problems. In both examples, we have compared the results with One-Level method by using different kinds of images. In Example 3, we have applied Two-Level method to image denoising problem and we have compared our results with Multigrid method. In Example 4, we have applied Two-Level method to image deblurring problem and we have compared our results with One-Level method by using different kinds of images. In Example 5, we have applied Two-Level method to image denoising problem and we have compared our results with One-Level method by using different kinds of images and different levels of Gaussian noise. In Example 6, we have also applied Two-Level method to image denoising problem and we have compared our results with One-Level method by using different kinds of images and different kinds of noises (Poisson, Random and Speckle). All examples designate that the Two-Level method is faster and more efficient for MC-based image denoising and image deblurring problems. In this paper, we have developed a more efficient algorithm for solving the mean curvature model. The proposed Two-Level algorithm is not intended for enhancing the quality of reconstruction of the model but for finding a fast solution. This technique can also be extended to other image reconstruction problems.

Acknowledgments

The first and the last author would like to acknowledge the support provided by the Deanship of Scientific Research at KFUPM for funding this work through small business project (SB181013).

References

[1] Robert Acar and Curtis R Vogel. Analysis of bounded variation penalty methods for ill-posed problems. *Inverse problems*, 10(6):1217–1229, 1994.

[2] Carlos Brito-Loeza and Ke Chen. Multigrid algorithm for high order denoising. *SIAM Journal on Imaging Sciences*, 3(3):363–389, 2010.

[3] Carlos Brito-Loeza, Ke Chen, and Victor Uc-Cetina. Image denoising using the gaussian curvature of the image surface. *Numerical Methods for Partial Differential Equations*, 32(3):1066–1089, 2016.

[4] Ke Chen. Introduction to variational image-processing models and applications, 2013.

[5] Ren Chun-feng and Ma Yi-chen. Residual a posteriori error estimate two-grid methods for the steady navier-stokes equation with stream function form. *Applied Mathematics and Mechanics*, 25(5):546–559, 2004.

[6] Faisal Fairag. A two-level finite-element discretization of the stream function form of the Navier-Stokes equations. *Computers & Mathematics with Applications*, 36(2):117–127, 1998.

[7] Faisal Fairag. Two level finite element technique for pressure recovery from stream function formulation of the Navier-Stokes equations. In Timothy J. Barth, Tony Chan, and Robert Haimes, editors, *Multiscale and Multiresolution Methods*, pages 297–306. Springer, 2002.

[8] Erich L Foster, Traian Iliescu, and Zhu Wang. A finite element discretization of the streamfunction formulation of the stationary quasi-geostrophic equations of the ocean. *Computer Methods in Applied Mechanics and Engineering*, 261:105–117, 2013.

[9] Vivette Girault and Jacques-Louis Lions. Two-grid finite-element schemes for the steady navier-stokes problem in polyhedra. *Portugaliae Mathematica*, 58(1):25–58, 2001.

[10] W Layton. A two-level discretization method for the Navier-Stokes equations. *Computers & Mathematics with Applications*, 26(2):33–38, 1993.

[11] W Layton and W Lenferink. Two-level picard and modified picard methods for the Navier-Stokes equations. *Applied Mathematics and Computation*, 69(2-3):263–274, 1995.

[12] W Layton and Lutz Tobiska. A two-level method with backtracking for the Navier-Stokes equations. *SIAM Journal on Numerical Analysis*, 35(5):2035–2054, 1998.

[13] Mo Mu and Jinchao Xu. A two-grid method of a mixed stokes–darcy model for coupling fluid flow with porous media flow. *SIAM journal on numerical analysis*, 45(5):1801–1813, 2007.

[14] Maxim A Olshanskii. Two-level method and some a priori estimates in unsteady navier-stokes calculations. *Journal of Computational and Applied Mathematics*, 104(2):173–191, 1999.

[15] Fuquan Ren, Tianshuang Qiu, and Hui Liu. Mean curvature regularization-based poisson image restoration. *Journal of Electronic Imaging*, 24(3):033025, 2015.

[16] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[17] Andrei Nikolaevich Tikhonov. Regularization of incorrectly posed problems. In *Soviet Math. Dokl*, volume 4, pages 1624–1627, 1963.

[18] Curtis R Vogel and Mary E Oman. Fast, robust total variation-based reconstruction of noisy, blurred images. *Image Processing, IEEE Transactions on*, 7(6):813–824, 1998.

[19] Jinchao Xu. A novel two-grid method for semilinear elliptic equations. *SIAM J. Sci. Comput.*, 15:231–237, 1994.

[20] Jinchao Xu. Two-grid discretization techniques for linear and nonlinear PDEs. *SIAM J. Numer. Anal.*, 33:1759–1777, 1996.

[21] Fenlin Yang, Ke Chen, Bo Yu, and Donghui Fang. A relaxed fixed point method for a mean curvature-based denoising model. *Optimization Methods and Software*, 29(2):274–285, 2014.

[22] Wei Zhu and Tony Chan. Image denoising using mean curvature of image surface. *SIAM Journal on Imaging Sciences*, 5(1):1–32, 2012.

[23] Wei Zhu, Xue-Cheng Tai, and Tony Chan. Augmented lagrangian method for a mean curvature based image denoising model. *Inverse Probl. Imaging*, 7(4):1409–1432, 2013.