MULTISCALE MODEL. SIMUL.
Vol. 5, No. 2, pp. 615–645

AN OPTIMIZATION-BASED MULTILEVEL ALGORITHM FOR TOTAL VARIATION IMAGE DENOISING[*]

TONY F. CHAN[†] AND KE CHEN[‡]

Abstract. This paper proposes a fast multilevel method using primal relaxations for the total variation image denoising and analyzes its convergence. The basic primal relaxation is known to get stuck at a nonstationary point (nearly a local minimum) of the minimization, whose solution is known to be "nonsmooth" in the space of functions with bounded variation. Our idea is to use coarse level corrections, overcoming the deadlock in a basic primal relaxation scheme and achieving much improvement over relaxation. Moreover, to reach a global minimizer, further refinement of the multilevel method is needed, and we propose a nonregular coarse level based on a patch-detection idea (relating to hemivariateness) to correct and improve the standard multilevel method. Both algorithmic and analytical results together with numerical experiments on both one- and two-dimensional images are presented.

Key words. image restoration, regularization, total variation, nondifferentiability, hemivariateness, primal relaxation, multilevel solvers, optimization, global convergence

AMS subject classifications. 68U10, 65F10, 65K10

DOI. 10.1137/050644999

1. Introduction.

The variational formulation has become a well-established technique for modeling a class of image processing problems [2, 7, 23, 56, 1, 70]. This paper is mainly concerned with fast solution issues for the most basic of these formulations—the regularization formulation based on the total variation (TV) minimization for image restoration due to Rudin, Osher, and Fatemi (see [56, 17]). Much of the previous work on developing fast solvers attempts to solve the associated nonlinear partial differential equation (PDE) from the Euler–Lagrange solution. This PDE has a nonlinear and highly nonsmooth coefficient which leads to convergence problems of many iterative solvers (even after extra smoothing is added [40, 54]).

As far as solving the nonlinear PDE is concerned, the existing work on the topic falls into three categories: (i) Fixed point iteration [1, 68, 71, 72, 69, 70, 55]. Once the coefficients are fixed, various iterative solver techniques have been considered [71, 72, 19, 20, 18, 46, 12, 36, 47, 60]. Further improvements are still useful. (ii) Explicit time marching scheme [56, 48] that turns the nonlinear PDE into a parabolic equation before using an explicit Euler method to march in time to (slow) convergence. (iii) Primal-dual method [23, 24, 7] that solves for both the primal and dual variables together in order to achieve faster convergence with the Newton method (and a constrained optimization with the dual variable). One observes that all these methods require a small positive parameter β to avoid singularities in the PDE coefficient associated with flat regions of the solution and consequently are sensitive to the parameter. In a recent work, recognizing that primal relaxation alone does not

[*]Received by the editors November 11, 2005; accepted for publication (in revised form) April 10, 2006; published electronically July 31, 2006. This work was supported in part by the Office of Naval Research ONR N00014-03-1-0888 and ONR N00014-06-1-0345, the National Institutes of Health NIH U54-RR021813, and the Leverhulme Trust RF/9/RFG/2005/0482.

http://www.siam.org/journals/mms/5-2/64499.html

[†]Department of Mathematics, University of California, Los Angeles, CA 90095-1555 (TonyC@college.ucla.edu, http://www.math.ucla.edu/~chan).

[‡]Corresponding author. Department of Mathematical Sciences, University of Liverpool, Peach Street, Liverpool L69 7ZL, UK (k.chen@liverpool.ac.uk, http://www.liv.ac.uk/~cmchenke).

work, Carter [13] tried the dual formulation that does not involve this parameter and achieved some initial success with the so-called barrier (interior point) dual relaxation method. We also note that some fundamental work [15, 34] has been done recently for the dual formulation that deserves further study. Other recent work that solves the same TV model on a single level includes the active set methods [39, 14, 40], the tube method [35], and the second-order cone programming method [31]. The interior point method is also studied in [30] for related models. This paper will focus on multilevel methods.

We propose an alternative to the PDE approach to solve the image minimization problem *directly*. The problems to overcome in this study include the treatment of the nondifferentiable functional in the minimization problem by using local minimization (and hence dimension reduction) and the stagnation problem with the primal relaxation by using coarse levels in a multilevel scheme aided by a patch-detection idea.

Below we introduce the image problem and the associated nonlinear equations before discussing some solution methods. In section 2, we present the primal relaxation method and the algorithmic development details in a multilevel setting. In section 3, we give a simple convergence result for the proposed algorithms and in section 4 consider an extension of the theory to a non-TV minimization. In section 5, we show a complexity analysis. In section 6, we present some supporting numerical results for the denoising case. Throughout the paper, we denote by MGM either a multigrid method or a multilevel method.

The total variation model and its solution methods. Denote by $u = u(x, y)$ the true image and $z = z(x, y)$ the observed image, both defined in the bounded and rectangular domain $\Omega = [0, 1] \times [0, 1] \subset \mathbb{R}^2$. In practice, only z is available in a discrete (matrix) form [3, 32, 57]. The observed image z has been contaminated in the data collection stage. The purpose of image restoration is to recover u as much as we can using a degradation model

$$Ku - z = \eta, \tag{1}$$

where η is a Gaussian white noise (unknown) and K is a known linear degradation operator; for deblurring problems K is often a convolution operator and for denoising problems $K = I$.

Image restoration is thus an inverse problem that may not have a unique solution. Some regularity condition has to be imposed on the solution space in order to turn the underlying ill-posed problem to a well posed one [70]. We shall use the well-known TV regularization to ensure that sharp features of an image are preserved [56]. However, we note that there are many other regularization functionals (some beyond the variational framework) that might be used; see [2, 45, 22, 10, 51, 46, 11] and the references therein.

Below we briefly review the common methodology to set the context for our algorithm in the next sections. Following early work [23], we choose the Tikhonov direct regularization technique to solve the inverse problem (1)

$$\min_u J(u), \qquad J(u) = \overline{\alpha} R(u) + \frac{1}{2}\|Ku - z\|_2^2, \tag{2}$$

where the regularization functional $R(u)$ is selected as the TV norm [56, 23]

$$R(u) = \|u\|_{TV} = \int_\Omega |\nabla u| dxdy = \int_\Omega \sqrt{u_x^2 + u_y^2} dxdy. \tag{3}$$

Here the parameter $\overline{\alpha}$ represents a trade-off between the quality of the solution and the fit to the observed data. Thus the overall image restoration problem is the following:

$$\min_u \overline{\alpha} \, \|u\|_{TV} + \frac{1}{2} \|Ku - z\|_2^2 \, . \tag{4}$$

The theoretical solution to problem (4) is given by the Euler–Lagrange equation

$$\overline{\alpha} \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) - K^* K u = -K^* z \tag{5}$$

with homogeneous Neumann boundary conditions, where K^* is the adjoint operator of K. Notice that the nonlinear coefficient may have a zero denominator, so the equation is not defined at such points (corresponding to flat regions of the solution). In one dimension, one uses the notation $\nabla u = \frac{du}{dx}$.

A commonly adopted idea to deal with $|\nabla u| = 0$ was to introduce (yet) another parameter β to (4) and (5), so the new Euler–Lagrange equation becomes

$$\overline{\alpha} \nabla \cdot \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}} \right) - K^* K u = -K^* z, \tag{6}$$

where corresponds to minimizing, instead of (4),

$$\min_u J_\beta(u), \qquad J_\beta(u) = \int_\Omega \Big[\overline{\alpha} \sqrt{u_x^2 + u_y^2 + \beta} + \frac{1}{2} (Ku - z)^2 \Big] dxdy \tag{7}$$

and in theory $u = u_\beta(x, y)$ differs from u in (5). Observe that when $\beta = 0$, (6) reduces to (5); moreover, $u_\beta \to u$ as $\beta \to 0$, as shown in [1].

The existing solution methods for solving (6) differ in how to deal with nonlinearities:

- **Fixed point iteration** [1, 68, 71, 72, 69, 70].
  Solve a lagged diffusion problem until $u^{k+1} - u^k$ is small:

  $$\overline{\alpha} \nabla \cdot \left(\frac{\nabla u^{k+1}}{\sqrt{|\nabla u^k|^2 + \beta}} \right) - K^* K u^{k+1} = -K^* z. \tag{8}$$

  There exists a large literature on this topic, mainly due to wide interest in developing fast iterative solvers for the above linear equations (once discretized). When K is a convolution operator, the challenge is to solve the resulting linear system without forming the discretized matrix of $K^* K$ (mimicking the capability of the fast multipole method) [71, 72, 19, 20, 18, 46]. Further improvements on robustness of these solvers are still needed.
- **Explicit time marching scheme** [56, 48].
  The original idea in [56] was refined in [48] as solving the following parabolic PDE until a steady state has been reached:

  $$u_t = |\nabla u| \left[\overline{\alpha} \nabla \cdot \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}} \right) - K^* K u + K^* z \right]. \tag{9}$$

  As remarked in [53], for linear problems, this type of idea represents a kind of relaxation scheme. The drawback is that the artificial time step Δt must be small due to the stability requirement.

- **Primal-dual method** [23, 24, 7].
  As discussed in [7], the Newton method for (6) leads to very slow or no convergence because z is often not a sufficiently close initial guess for u. Introducing the dual variable (vector) $\omega = \nabla u / \sqrt{|\nabla u|^2 + \beta}$ appears to have made the combined system
  $$\begin{cases} \overline{\alpha} \nabla \cdot \omega - K^* K u = -K^* z, \\ \omega \sqrt{|\nabla u|^2 + \beta} - \nabla u = 0 \end{cases}$$
  in two variables (u, ω) more amenable to Newton iterations, as the new system is nearly "linear" in the two variables (not as linear as a single variable after elimination). Note that ω is constrained in each iteration step, so the overall algorithm needs some careful implementation. However, up to now, an efficient multilevel implementation of this method remains to be developed. The same is true for the alternative primal-dual formulation [35].

In addition, there exist the powerful dual formulations [15, 34] that replace the primal variable u by its dual variable $\mathbf{p} = (p_1, p_2)$. The work of [17, 41, 54] modifies the TV formulation so that the new equations become more amenable to numerical implementation.

REMARK 1. *Of the three types of methods, the recommended approach seems to be the Chan–Golub–Mulet (CGM) method* [23]*, from our experience, although there is scope to realize the implementation more efficiently and to provide a complete theory for it. A simple observation of* (5) *reveals that there is nothing wrong with it in the sense that the equation is still "nonsingular" as $\omega = \nabla u / |\nabla u|$ is always bounded if $|\nabla u| \neq 0$, although it tends towards being undefined as $|\nabla u|$ gets smaller. This suggests that it may not be an appropriate action to take in introducing the parameter β dependent on the size of $|\nabla u|$ which is a free gradient otherwise. For fixed point iterations, this point is more pronounced since the "linearized" PDE* (8) *has large coefficients near points where the solution is flat (or less interesting!), creating somewhat "unnecessary" jumps in coefficients.*

In what follows we shall attempt to solve the original primal and optimization formulation (4) by a multilevel algorithm and concentrate on the denoising problem with $K = I$.

2. A multilevel scheme with piecewise constant relaxation.

Our objective is to solve (4) using a multilevel method. Such a task is more challenging than solving the regularized equation (6), as previously done in [18, 68, 71, 69, 70, 25, 58, 29], where the merit functional is differentiable.

For a minimization problem, the important issue in designing a multilevel method is how to make use of an approximate solution $\widetilde{u}$ to improve it further or how to measure the "distance" from the true minimizer so that this information can be passed onto the coarse levels somehow. (We note that for differentiable functions, first-order conditions can be used to define a residual of $\widetilde{u}$, and then they are passed onto the modification of a coarse level minimization functional; see [62, 8, 49].) That is to say, here there is no obvious way to define a residual correction functional (as done for an operator equation [26]).

We shall first discuss standard multilevel schemes to solve (4), and it turns out that such approaches, although always improving on the coordinate descent method, is not sufficient to reach the global minimizer due to nondifferentiability. We then present our new multilevel algorithm before addressing convergence.

For a given image $z \in \mathbb{R}^{n\times n}$, assume $n = 2^L$, and let the standard coarsening be used giving rise to $L+1$ levels $k = 1(\text{finest}), 2, \ldots, L, L+1(\text{coarsest})$. Denote the dimension of level k by $\tau_k \times \tau_k$ with $\tau_k = n/2^{k-1}$. (For $z \in \mathbb{R}^n$, the same assumption is used in coarsening.)

2.1. Local minimization by coordinate descent methods. We shall use the coordinate descent method [6, 33, 50, 67, 38] as a smoothing method to find a solution near the true minimizer (not to find the minimizer itself). In the image context, Carter [13] appears to be the first who reported on using this approach.

Below we illustrate the well-known problem with the converged solutions to a nonsmooth minimization stuck at a local nonstationary point. Then in explaining how to unstick the relaxation, we are naturally motivated to consider the multilevel approach and give the early hint as to why patch relaxation may be needed. More rigorous explanation is given in the next section.

To this end, consider the discretized form of (4), respectively, in one and two dimensions:

$$\min_{u\in\mathbb{R}^n} J(u), \qquad J(u) = \alpha\sum_{j=1}^{n-1}|u_j - u_{j+1}| + \frac{1}{2}\sum_{j=1}^{n}(u_j - z_j)^2 \tag{10}$$

and

$$\begin{aligned}\min_{u\in\mathbb{R}^{n\times n}} J(u), \qquad J(u) = {}& \alpha\sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\sqrt{(u_{i,j}-u_{i,j+1})^2 + (u_{i,j}-u_{i+1,j})^2}\\ &+ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(u_{i,j}-z_{i,j})^2\end{aligned} \tag{11}$$

with $\alpha = \overline{\alpha}/h$ and $h = 1/(n-1)$. For the purpose of illustration and motivation of ideas, we shall mainly use the one-dimensional (1D) problem (10).

The coordinate descent method may be used to solve (10) as follows:

$$\begin{aligned}&\text{Given} \quad u^{(0)} = [u_1^{(0)}, \ldots, u_n^{(0)}] \text{ with } k = 0,\\ &\text{Solve} \quad u_j^{(k)} = \operatorname*{argmin}_{u_j\in\mathbb{R}} J^{\text{loc}}(u_j) \text{ for } j = 1, 2, \ldots, n,\\ &\qquad\text{Set } u^{(k+1)} = [u_1^{(k)}, \ldots, u_n^{(k)}] \text{ and repeat the above step with } k = k+1\\ &\qquad\qquad\text{until a prescribed stopping step on } k,\end{aligned} \tag{12}$$

where $J^{\text{loc}}(u_j) = \alpha|u_{j-1}^{(k)} - u_j| + \alpha|u_j - u_{j+1}^{(k)}| + \frac{1}{2}(u_j - z_j)^2$ for $1 < j < n$ with $J^{\text{loc}}(u_1) = \alpha|u_1 - u_2^{(k)}| + \frac{1}{2}(u_1 - z_1)^2$ and $J^{\text{loc}}(u_n) = \alpha|u_{n-1}^{(k)} - u_n| + \frac{1}{2}(u_n - z_n)^2$ due to Neumann's condition for the continuous variable u. Here problem (12) can be solved analytically because the local solution u_j is taken from the set $\{z_j - 2\alpha, z_j + 2\alpha, u_{j-1}, u_{j+1}, z_j\}$. Similar results on exact 1D solutions can be found in [9, 59, 73]. Figure 1 shows two examples from using the coordinate descent method where it works fine for the first and not for the second, as previously observed in [13] and known to some readers.

2.2. Standard multilevel methods. To develop a multilevel method to improve on the coordinate descent method, we may interpret solving (12) as looking for the best scalar constant $u_j^{(k)}$ that minimizes the local merit functional $J^{\text{loc}}(u_j)$ or, equally, as finding the best correction constant c that makes $u_j^{(new)} = u_j^{(old)} + c$

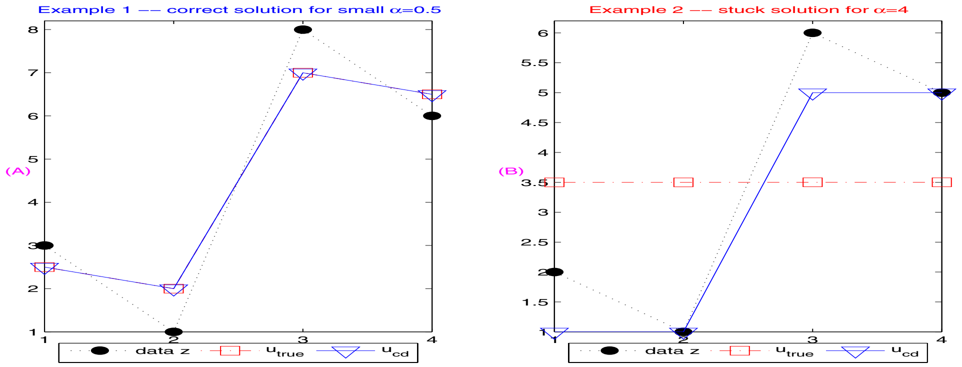

FIG. 1. *Two examples to illustrate the coordinate descent (cd) method: the first* ($\alpha = 1/2$) *solution is correct, while the second* ($\alpha = 4$) *solution gets stuck at the wrong position. Here the true solution* u_{true} *of* (10) *is denoted by* □ *and the cd solution* u_{cd} *by* ▽, *while the image data* z *is denoted by* •.

the minimizer of $J^{\text{loc}}(u_j)$. These local correction constants may be associated with a piecewise constant function for the finest level. Similarly, we may search for the best piecewise constant corrections on all coarser levels.

Suppose $\widetilde{u}$ is our current approximation for the minimizer u of (10) on the finest level 1. Denote a level k constant vector (for $k = 1, 2, \ldots, L+1$) by

$$c^{(k)} = [c_1 \ c_2 \ \cdots \ c_{\tau_k}]^T. \tag{13}$$

Then define implicitly the interpolation matrix $P_k : \mathbb{R}^{\tau_k} \to \mathbb{R}^n$ from level k to level 1 from

$$c = P_k c^{(k)} = [\underbrace{c_1 \ \cdots \ c_1}_{\text{block } 1} \ \underbrace{c_2 \ \cdots \ c_2}_{\text{block } 2} \ \cdots \ \cdots \ \underbrace{c_{\tau_k} \ \cdots \ c_{\tau_k}}_{\text{block } \tau_k}]^T, \tag{14}$$

where each block is of size $b = n/\tau_k = 2^{k-1}$. Thus on each level $k = 1, \ldots, L+1$, we propose to solve the correction problem

$$\hat{c}^{(k)} = \operatorname*{argmin}_{c^{(k)}} J(\widetilde{u} + P_k c^{(k)}), \qquad \hat{c} = P_k \hat{c}^{(k)} \in \mathbb{R}^n, \quad c^{(k)} \in \mathbb{R}^{\tau_k}, \tag{15}$$

by a coordinate descent method and then set $\widetilde{u} = \widetilde{u} + \hat{c}$. This will define a multilevel method which we shall call the standard multilevel method. Note that there are only k unknowns in problem (15). Here on the coarsest level with $\tau_{L+1} = 1$, $\hat{c}$ is just a single constant, while on the finest level with $\tau_1 = n$, $\hat{c}$ is a vector containing n unknowns.

To solve (15) efficiently, it is of interest to simplify its functional. By direct calculation,

$$\begin{aligned}
&J(\widetilde{u} + c) \\
&= \alpha \Big[|\widetilde{u}_1 + c_1 - (\widetilde{u}_2 + c_1)| + \cdots + |\widetilde{u}_b + c_1 - (\widetilde{u}_{b+1} + c_2)| \\
&\quad + |\widetilde{u}_{b+1} + c_2 - (\widetilde{u}_{b+2} + c_2)| + \cdots + |\widetilde{u}_{2b} + c_2 - (\widetilde{u}_{2b+1} + c_3)| \\
&\quad + \cdots + |\widetilde{u}_{n-b+1} + c_{\tau_k} - (\widetilde{u}_{n-b+2} + c_{\tau_k})| + \cdots + |\widetilde{u}_{n-1} + c_{\tau_k} - (\widetilde{u}_n + c_{\tau_k})| \Big]
\end{aligned}$$

$$
\begin{aligned}
&+ \frac{1}{2}\sum_{\ell=1}^{\tau_k}\sum_{j=(\ell-1)b+1}^{\ell b}(\widetilde{u}_j + c_j - z_j)^2 \\
&= \alpha \sum_{j=1}^{\tau_k-1}\Big|\widetilde{u}_{jb} - \widetilde{u}_{jb+1} + c_j - c_{j+1}\Big| + \frac{1}{2}\sum_{\ell=1}^{\tau_k}\sum_{j=(\ell-1)b+1}^{\ell b}(\widetilde{u}_j + c_\ell - z_j)^2 + J_0(\widetilde{u}) \\
&= \alpha \sum_{j=1}^{\tau_k-1}\Big|\overline{c}_j - \overline{c}_{j+1}\Big| + \frac{b}{2}\sum_{\ell=1}^{\tau_k}(c_\ell - \widetilde{w}_\ell)^2 + J_0(\widetilde{u}) + J_1(\widetilde{u}) \qquad (16) \\
&= b\Big\{ \underbrace{\frac{\alpha}{b}\sum_{j=1}^{\tau_k-1}\Big|\overline{c}_j - \overline{c}_{j+1}\Big| + \frac{1}{2}\sum_{j=1}^{\tau_k}(\overline{c}_j - \overline{z}_j)^2}_{\text{coarse level equation with a different “}\alpha\text{”}} + J_0(\widetilde{u})/b + J_1(\widetilde{u})/b\Big\}, \qquad (17)
\end{aligned}
$$

where we separated the terms associated with c from the other $J_0(\widetilde{u})$ and $J_1(\widetilde{u})$ terms not involving c. In particular, $\widetilde{d}_{j+1} = \widetilde{u}_{jb} - \widetilde{u}_{jb+1}$ (with $\widetilde{d}_1 = 0$) denotes the jth jump between two adjacent blocks and

$$
d_j = \sum_{\ell=1}^{j}\widetilde{d}_\ell, \qquad \overline{c}_j = c_j - d_j, \qquad \overline{z}_j = \widetilde{w}_j - d_j \quad \text{for } j = 1, 2, \ldots, \tau_k. \tag{18}
$$

Here with $\widetilde{z}_j = z_j - \widetilde{u}_j$ for $j = 1, 2, \ldots, n$, the mean “z” value for each block is

$$
\widetilde{w}_\ell = \sum_{j=(\ell-1)b+1}^{\ell b}\widetilde{z}_j/b, \tag{19}
$$

and (16) is obtained from using the simple equality of the type

$$
\sum_{j=1}^{b}(v - z_j)^2 = bv^2 - 2\left(\sum_{j=1}^{b} z_j\right)v + \sum_{j=1}^{b} z_j^2 = b\Big(v - \overline{z}\Big)^2 + \sum_{j=1}^{b} z_j^2 - b\overline{z}^2,
$$

where $\overline{z} = \sum_{j=1}^{b} z_j/b$.

Thus from (17), we see that minimizing the finest level correction equation associated with level k is equivalent to solving the following level k correction equation:

$$
\min_{\overline{c}\in\mathbb{R}^{\tau_k}} \frac{\alpha}{b}\sum_{j=1}^{\tau_k-1}\Big|\overline{c}_j - \overline{c}_{j+1}\Big| + \frac{1}{2}\sum_{j=1}^{\tau_k}(\overline{c}_j - \overline{z}_j)^2, \tag{20}
$$

whose solution defines the correction vector $c = \overline{c} + d$. As with most multilevel methods, we shall not solve (20) exactly. Instead we apply the smoothing method of section 2.1 and then apply the idea repeatedly with coarser levels until the coarsest level where the solution is simply $\overline{c}_1 = \overline{z}_1$ as $\tau_1 = 1$.

We now summarize the overall multilevel algorithm as follows.

ALGORITHM 1A. *Given z and an initial guess $\widetilde{u} = z$ with $L + 1$ levels, do the following:*

(1) *Let $u_0 = \widetilde{u}$.*

(2) *Smooth the approximation on the finest level 1; i.e., solve (12) for $j = 1, 2, \ldots, n$.*

```
(3) On coarse levels $k = 2, 3, \ldots, L+1$, do the following:
    — compute $\widetilde{z} = z - \widetilde{u}$ via (19);
    — compute the local mean $\widetilde{w}_\ell = mean\big(\widetilde{z}\big((\ell-1)b+1 : \ell b\big)\big)$ via (19);
    — compute $\widetilde{d}_j, d_j, \overline{c}_j, \overline{z}_j$ via (18);
    — solve (20) by solving local minimization as in (12) if $k \le L$ or
    — on the coarsest level $k = L+1$, the correction constant is simply
              $\overline{c} = mean(\overline{z}) = c = mean(z - \widetilde{u})$;
    — add the correction $\widetilde{u} = \widetilde{u} + P_k c^{(k)}$ via (15).
(4) If $\|\widetilde{u} - u_0\|_2$ is small enough, stop or return to step (1).
```

Algorithm 1A has been applied to solve a number of problems successfully; for example, see Figure 2(A), where we show how the coarse level problem is solved correctly by the coordinate descent method.

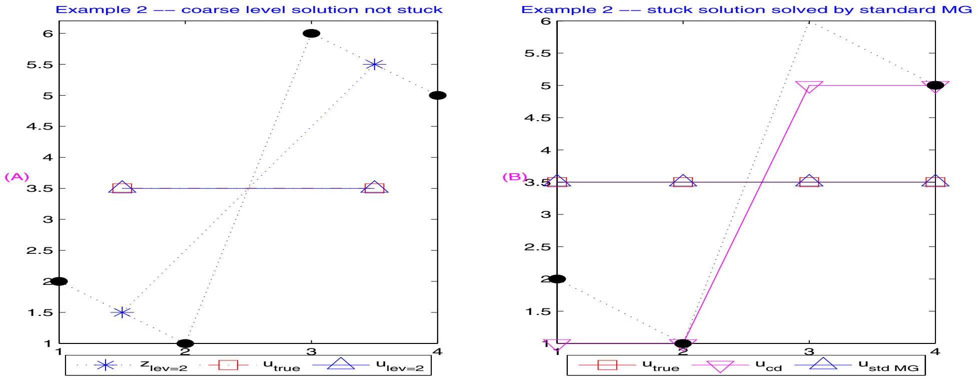

FIG. 2. *Example to illustrate the success of the standard MGM (Algorithm* 1A*): the second* ($\alpha = 4$) *example from Figure* 1 *can be solved by the MGM to obtain the correct minimizer* $\triangle$*. Here again the true solution* u_{true} *of* (10) *is denoted by* $\square$ *and the cd solution* u_{cd} *by* $\bigtriangledown$*, while the image data* z *is denoted by* $\bullet$.

In the literature, there exists related work to our piecewise constant-based MGM as in Algorithm 1A. For convex minimization, the work of [63, 66, 65, 64, 27] suggests the use of locally spanned spaces of piecewise polynomials to construct local minimizations. For variational inequalities, the monotone MGM work of [43, 44] suggests using piecewise linear spaces to construct local minimizations. However, the assumption of the differentiability of the merit functional is required. For image restoration, the authors of [14] established that any function from the space of bounded variation (BV) functions can be uniformly approximated by piecewise constant functions.

However, Algorithm 1A as a standard multilevel method can still get stuck just as the coordinate descent method does for some problems, as illustrated in Figure 3. In attempting to find a remedy for Algorithm 1A, we observe that the incorrect multilevel solution in Figure 3 is "geometrically" correct; i.e., the wrong solution is associated with flat patches, and these flat patches, are of the correct sizes, but the values (patch heights) are incorrect. So if we add an extra coarse grid based on these patches we find that the modified MGM will converge correctly, as shown in Figure 4.

From Figures 1–4, we may summarize three points on solving (10) by a coordinate descent method and the multilevel method (Algorithm 1A):

1. For small α when the true solution has no patches, the coordinate descent solution (and also the MGM) can be quite good. See Figure 1(A).

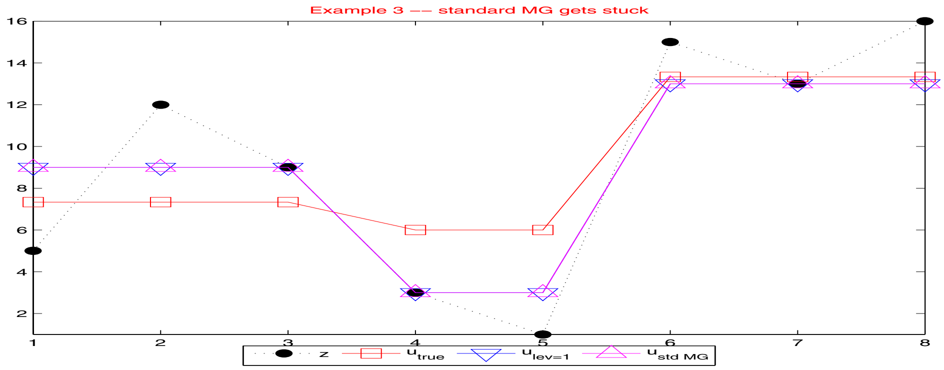

FIG. 3. *Example to illustrate the failure of the standard MGM (Algorithm* 1A*): Here* $n = 8$*,* $\alpha = 4$*, and* z *is shown as* ● *and the true solution as* □*. Note that the solution* u_{stdMG} *(as* △*) does not improve much from the initial "stuck" solution* ▽*.*

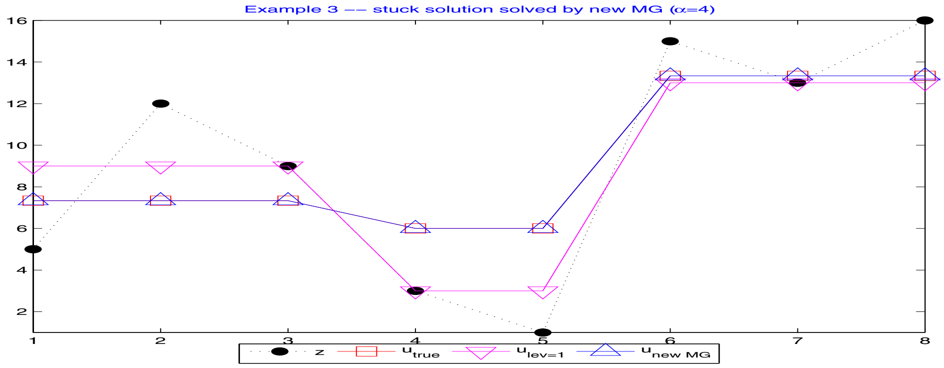

FIG. 4. *The example showing that the new algorithm produces* u_{newMG} *converging to the true solution (as* □*), i.e., improving on the initial "stuck" solution* ▽*.*

2. For large α, the true solution normally has patches, so both the coordinate descent method and the multilevel method can get stuck easily. See Figure 1(B). This is quite reasonable for each local minimization at a point, because the local object functional will increase if the wrong solution is set to be the correct solution at this point. To be more specific, consider node 3 in Figure 1(B): as $z = [2\ 1\ 6\ 5]$ and $\widetilde{u} = [1\ 1\ 5\ 5]$, the local problem is $\min_{u_3} 4|u_3 - 1| + 4|u_3 - 5| + (u_3 - 6)^2/2$, which has the local minimizer $u_3 = u_{lev=1}(3) = 5$ at which $J^{\rm loc}(u_3) = 16.5$, while $J^{\rm loc}(u_3^*) = 19.124$ at the true global minimizer $u_3^* = 3.5004$. Note that the TV term takes the same value of 16 at either u_3 or u_3^*. However, if nodes $1, 2$ and nodes $3, 4$ are viewed as two separate constant patches (on level 2), the new minimizing problem is

$$\begin{aligned}&\min_{u_1,u_2} 4|u_1-u_2| + \frac{(u_1-2)^2}{2} + \frac{(u_1-1)^2}{2} + \frac{(u_2-6)^2}{2} + \frac{(u_2-5)^2}{2}\\ &\quad = \underbrace{2\min_{u_1,u_2}\left[2|u_1-u_2| + \frac{1}{2}(u_1-1.5)^2 + \frac{1}{2}(u_2-5.5)^2\right]}_{\text{coarse problem at level 2 with “}\alpha = 2\text{”}} + \frac{1}{2},\end{aligned} \tag{21}$$

which has the local minimizer $u_1 = u_{lev=2}(1) = u_{lev=2}(2) = 3.4996$ and $u_2 = u_{lev=2}(3) = u_{lev=2}(4) = 3.5004$, obtainable by local relaxation, and coincides with the global minimizer!

Thus a natural way to introduce "global interaction" is to let the patches (group of points) achieve minimization together—this is exactly the idea of using coarse grids in a multilevel strategy as done in Figure 2(A), where two groups move together on a coarse level to drive the solution to the correct value.

3. For large α, the patch sizes are not likely to conform to the multigrid coarsening all the time. So a standard multilevel method may not be sufficient, as shown in Figure 3, where the standard multilevel solution is stuck in the same way as a coordinate descent solution. We may need a special coarse grid to accommodate the varying sizes; i.e., three patches move on a coarse level to drive the solution to the correct value in Figure 4—this is the early indication of why our new multilevel idea will be successful.

It turns out that the above observations provided all the insights needed to overcome a stuck solution. They not only motivate our algorithm next but also can be justified theoretically (section 3).

A new multilevel algorithm. We now consider how to implement the patch idea. The flat patch-based coarsening idea is to supplement the use of a standard coarsening. Our refinement consists of two parts based on solving (25), i.e.,

$$\min_c J(\widetilde{u} + c).$$

First, assume the current approximation $\widetilde{u}$ has a union of m flat patches of respective lengths $b_1, b_2, \ldots, b_m$ (with $b_j \geq 1$). Second, we take an extra coarse level with m piecewise constants, as in (14) and (15), and seek to optimize these constants. Let

$$c^{\langle m \rangle} = [c_1 \ c_2 \ \cdots \ c_m]^T$$

and

$$c = P_m c^{\langle m \rangle} = [\underbrace{c_1 \ \cdots \ c_1}_{\text{block } b_1} \ \underbrace{c_2 \ \cdots \ c_2}_{\text{block } b_2} \ \cdots \ \cdots \ \underbrace{c_m \ \cdots \ c_m}_{\text{block } b_m}]^T. \tag{22}$$

To solve $\min_c J(\widetilde{u} + c)$, one can derive an equation for c similarly to before with the only change being computing the local mean using b_j instead of b. More precisely, similar to (17), we obtain

$$J(\widetilde{u} + c) = \alpha \sum_{j=1}^{m-1} \left| \overline{c}_j - \overline{c}_{j+1} \right| + \frac{1}{2} \sum_{j=1}^{m} b_j (\overline{c}_j - \overline{z}_j)^2 + M(\widetilde{u}), \tag{23}$$

where $M(\widetilde{u})$ is a generic term not depending on c or $\overline{c}$. Clearly the solution of (23) proceeds just as with (20); in particular the local minimization problem at a typical node j takes the form

$$\min_{\overline{c}_j} J^{\text{loc}}(\overline{c}_j), \qquad J^{\text{loc}}(\overline{c}_j) = \overline{\alpha} |\overline{c}_j - \overline{c}_{j-1}^{(k)}| + \overline{\alpha} |\overline{c}_j - \overline{c}_{j+1}^{(k)}| + \frac{1}{2} (\overline{c}_j - \overline{z}_j)^2, \tag{24}$$

where $\overline{\alpha} = \alpha / b_j$, which can be solved exactly as with (12).

In summary, our new piecewise constant-based multilevel method is modified from Algorithm 1A to the following.

ALGORITHM 1. *Given z and an initial guess $\widetilde{u} = z$ with $L+1$ levels, do the following:*

(1) *Let $u_0 = \widetilde{u}$.*
(2) *Smooth the approximation on the finest level 1; i.e., solve* (12) *for $j = 1, 2, \dots, n$.*
(3) *On coarse levels $k = 2, 3, \dots, L+1$, do the following:*
— *compute $\widetilde{z} = z - \widetilde{u}$ via* (19)*;*
— *compute the local mean $\widetilde{w}_\ell = mean\big(\widetilde{z}\big((\ell-1)b+1 : \ell b\big)\big)$ via* (19)*;*
— *compute $\widetilde{d}_j, d_j, \overline{c}_j, \overline{z}_j$ via* (18)*;*
— *solve* (20) *by solving local minimization as in* (12) *if $k \le L$ or*
— *on the coarsest level $k = L+1$, the correction constant is simply*

$$\overline{c} = mean(\overline{z}) = c = mean(z - \widetilde{u});$$

— *add the correction $\widetilde{u} = \widetilde{u} + P_k c^{(k)}$ via* (15)*.*
(4) *Detect flat patches in the approximation $\widetilde{u}$. Solve the patch level minimization problem* (23) *via* (24)*.*
Add the patch correction $\widetilde{u} = \widetilde{u} + P_k c^{\langle m \rangle}$ via (22)*.*
(5) *If $\|\widetilde{u} - u_0\|_2$ is small enough, stop or return to step* (1)*.*

As shown in Figure 4, Algorithm 1 has much improved on Algorithm 1A in practice. In fact, there is a fundamental reason why the new Algorithm 1 is better than the standard multigrid method; see Lemmas 3.2–3.3.

2.3. The two-dimensional (2D) algorithm. Assume that $\widetilde{u} \in \mathbb{R}^{n \times n}$ is our current approximation to (11). We wish to find the best piecewise constant function $C \in \mathbb{R}^{n\times n}$ so that it is the solution of the following:

$$\min_{c} J(\widetilde{u} + c). \tag{25}$$

Such a problem is equivalent to the original problem (11). Instead, we propose to solve it on coarse levels:

$$\hat{c} = \operatorname*{argmin}_{\mathbf{c} \in \mathbb{R}^{\tau_k \times \tau_k}} J(\widetilde{u} + P_k \mathbf{c}), \qquad C = P_k \hat{c}, \tag{26}$$

where $P_k : \mathbb{R}^{\tau_k \times \tau_k} \to \mathbb{R}^{n \times n}$ is the interpolation operator, so $C \in \mathbb{R}^{n\times n}$. That is to say, we seek a sequence of corrections of the form $C_k = P_k \hat{c}$ (with $\hat{c}$ a constant matrix of c's):

$$C_k = \left[\begin{array}{cc|ccc|cc} 0 & 0 & \cdots & \cdots & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cline{3-5} 0 & \cdots & c & \cdots & c & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & \cdots & c & \cdots & c & \cdots & 0 \\ \cline{3-5} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \cdots & \cdots & 0 & 0 \end{array}\right] \text{ to approximate}$$

$$\left[\begin{array}{cc|ccc|cc} c_{1,1} & \cdots & \cdots & \cdots & \cdots & \cdots & c_{1,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cline{3-5} c_{i,1} & \cdots & c_{i,i} & \cdots & c_{i,j} & \cdots & c_{i,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ c_{j,1} & \cdots & c_{j,i} & \cdots & c_{j,j} & \cdots & c_{j,n} \\ \cline{3-5} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ c_{n,1} & \cdots & \cdots & \cdots & \cdots & \cdots & c_{n,n} \end{array}\right].$$

Below we discuss how to solve (26). As our methodology does not depend on the dimension, describing our multilevel algorithm in two dimensions is straightforward. However, for efficient implementation, we show some details of simplifying the main formulae.

2D local minimization. Consider the minimization of (11) by the coordinate direction method as in (12):

$$\begin{aligned} &\text{Given} \quad u^{(0)} = (u_{i,j}^{(0)}) \text{ with } k=0,\\ (27)\qquad &\text{Solve} \quad u_{i,j}^{(k)} = \operatorname*{argmin}_{u_{i,j}\in\mathbb{R}} J^{\text{loc}}(u_{i,j}) \text{ for } i,j=1,2,\dots,n,\\ &\text{Set } u^{(k+1)} = (u_{i,j}^{(k)}) \text{ and repeat the above step with } k=k+1\\ &\qquad\text{until a prescribed stopping step on } k, \end{aligned}$$

where

$$\begin{aligned} J^{\text{loc}}(u_{i,j}) = \alpha\Big[&\sqrt{(u_{i,j}-u_{i+1,j}^{(k)})^2 + (u_{i,j}-u_{i,j+1}^{(k)})^2}\\ &+\sqrt{(u_{i,j}-u_{i-1,j}^{(k)})^2 + (u_{i-1,j}^{(k)}-u_{i-1,j+1}^{(k)})^2}\\ &+\sqrt{(u_{i,j}-u_{i,j-1}^{(k)})^2 + (u_{i,j-1}^{(k)}-u_{i+1,j-1}^{(k)})^2}\ \Big]\\ &+\frac{1}{2}(u_{i,j}-z_{i,j})^2, \end{aligned}$$

where, due to Neumann's condition for the continuous variable u, all difference terms involving indices in subscripts larger than n are set to zero (representing first derivatives in the first sum at image boundaries); e.g., when $i=2$, $j=n$, the first square root term in $J^{\text{loc}}(u_{2,n})$ becomes

$$\sqrt{(u_{2,n}-u_{3,n}^{(k)})^2 + (u_{2,n}-u_{2,n+1}^{(k)})^2} = |u_{2,n}-u_{3,n}^{(k)}|.$$

Although problem (27) does not have an analytical solution, our new method is not sensitive to the choice of elementary iterative schemes for solving (27); two simple methods are shown in Appendix 1.

It should be remarked that earlier work on formulation and experiments of (12) and (27) can be found in Carter [13], where a coordinate descent method was extensively used. After a few sweeps, the resulting solution u is smooth in some sense, but, unfortunately, the method will converge (quickly) to the wrong solution as the number of steps increases. This is known as the local minimizer getting "stuck"; see [13] and refer also to [38] for a coordinate descent method applied to a different image restoration model. It is in fact well known [67] that *for a general nondifferentiable function, even if it is convex, the coordinate descent method may get stuck at a nonstationary point.* We also refer readers to other studies on nonlinear Gauss–Seidel-type methods [6, 33, 67, 27, 21, 4].

The local minimization on a general level k. On level k, set $b=2^{k-1}$, $k_1=(i-1)b+1$, $k_2=ib$, $\ell_1=(j-1)b+1$, $\ell_2=jb$. Let $c=(c_{i,j})$. Then the (i,j)th computational block (stencil) involving $c_{i,j}$ on level k can be depicted in terms of pixel updates on level 1 as follows:

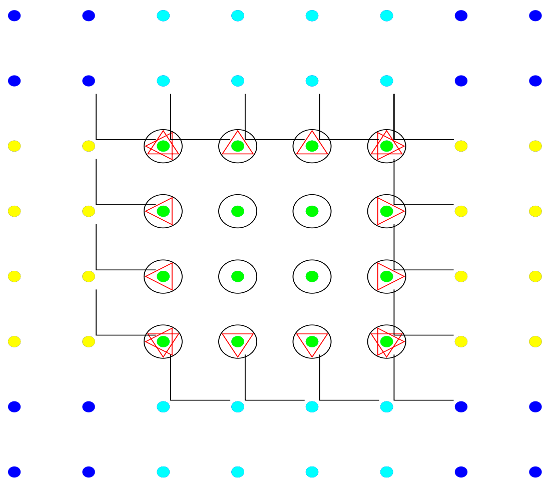

FIG. 5. *Illustration of a level $k = 3$, 4×4, box $\odot$ interacting with neighboring pixels $\bullet$. The interior 2×2 pixels do not contribute to a constant update.*

(28)

$$\begin{array}{c|ccc|c}
\vdots & \vdots & \cdots & \vdots & \vdots \\
\widetilde{u}_{k_1-1,\ell_2+1}+c_{i-1,j+1} & \widetilde{u}_{k_1,\ell_2+1}+c_{i,j+1} & \cdots & \widetilde{u}_{k_2,\ell_2+1}+c_{i,j+1} & \widetilde{u}_{k_2+1,\ell_2+1}+c_{i+1,j+1} \\ \hline
\widetilde{u}_{k_1-1,\ell_2}+c_{i-1,j} & \widetilde{u}_{k_1,\ell_2}+c_{i,j} & \cdots & \widetilde{u}_{k_2,\ell_2}+c_{i,j} & \widetilde{u}_{k_2+1,\ell_2}+c_{i+1,j} \\
\cdots & \vdots & \cdots & \vdots & \cdots \\
\widetilde{u}_{k_1-1,\ell_1}+c_{i-1,j} & \widetilde{u}_{k_1,\ell_1}+c_{i,j} & \cdots & \widetilde{u}_{k_2,\ell_1}+c_{i,j} & \widetilde{u}_{k_2+1,\ell_1}+c_{i+1,j} \\ \hline
\widetilde{u}_{k_1-1,\ell_1-1}+c_{i-1,j-1} & \widetilde{u}_{k_1,\ell_1-1}+c_{i,j-1} & \cdots & \widetilde{u}_{k_2,\ell_1-1}+c_{i,j-1} & \widetilde{u}_{k_2+1,\ell_1-1}+c_{i+1,j-1} \\
\vdots & \vdots & \cdots & \vdots & \vdots
\end{array}$$

The middle block as a stencil is now illustrated for the case of $k = 3$ (i.e., $b = 2^{k-1} = 4$) in Figure 5, where only the boundary pixels denoted by symbols $\triangleleft, \triangle, \triangleright, \triangle$ involve $c_{i,j}$ in their TV terms. Here the "interior" pixels do not involve $c_{i,j}$ because the finite differences are not changed at interior pixels by the same constant update as in

$$\begin{aligned}
&\sqrt{[(c_{i,j}+\widetilde{u}_{k,\ell})-(c_{i,j}+\widetilde{u}_{k+1,\ell})]^2+[(c_{i,j}+\widetilde{u}_{k,\ell})-(c_{i,j}+\widetilde{u}_{k,\ell+1})]^2} \\
&\qquad = \sqrt{[\widetilde{u}_{k,\ell}-\widetilde{u}_{k+1,\ell}]^2+[\widetilde{u}_{k,\ell}-\widetilde{u}_{k,\ell+1}]^2}.
\end{aligned}$$

This observation can help us simplify the formulation. (In one dimension, similar simplification to avoid interior nodes reduces (14) to a coarse problem involving interactions between jumps of 1D blocks.)

Then, as a local minimization for $c_{i,j}$, the problem is equivalent to minimizing the following (resulting from the four sides of the above stencil; refer to (11) and Figure 5):

(29)

$$F(c_{i,j}) = J^{\mathrm{loc}}(\widetilde{u}_{i,j}+c_{i,j}) = \alpha \sum_{\ell=\ell_1}^{\ell_2} \sqrt{[c_{i,j}-(\widetilde{u}_{k_1-1,\ell}-\widetilde{u}_{k_1,\ell})]^2+(\widetilde{u}_{k_1-1,\ell}-\widetilde{u}_{k_1-1,\ell+1})^2}$$

$$
\begin{aligned}
&+ \alpha \sum_{k=k_1}^{k_2-1} \sqrt{[c_{i,j} - (\widetilde{u}_{k,\ell_2+1} - \widetilde{u}_{k,\ell_2})]^2 + (\widetilde{u}_{k,\ell_2} - \widetilde{u}_{k+1,\ell_2})^2} \\
&+ \alpha \sqrt{[c_{i,j} - (\widetilde{u}_{k_2,\ell_2+1} - \widetilde{u}_{k_2,\ell_2})]^2 + [c_{i,j} - (\widetilde{u}_{k_2+1,\ell_2} - \widetilde{u}_{k_2,\ell_2})]^2} \\
&+ \alpha \sum_{\ell=\ell_1}^{\ell_2-1} \sqrt{[c_{i,j} - (\widetilde{u}_{k_2+1,\ell} - \widetilde{u}_{k_2,\ell})]^2 + (\widetilde{u}_{k_2,\ell} - \widetilde{u}_{k_2,\ell+1})^2} \\
&+ \alpha \sum_{k=k_1}^{k_2} \sqrt{[c_{i,j} - (\widetilde{u}_{k,\ell_1-1} - \widetilde{u}_{k,\ell_1})]^2 + (\widetilde{u}_{k,\ell_1-1} - \widetilde{u}_{k+1,\ell_1-1})^2} \\
&+ \frac{1}{2} \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big[c_{k,\ell} - (z_{k,\ell} - \widetilde{u}_{k,\ell}) \Big]^2 .
\end{aligned}
$$

To simplify the formulation, we define

$$
\begin{aligned}
&\widetilde{z}_{k,\ell} = z_{k,\ell} - \widetilde{u}_{k,\ell}, \quad \widetilde{w}_{i,j} = mean\Big(\widetilde{z}(k_1 : k_2, \ell_1 : \ell_2) \Big) = \frac{1}{b^2} \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \widetilde{z}(k,\ell), \\
&\widetilde{v}_{k,\ell} = \widetilde{u}_{k,\ell+1} - \widetilde{u}_{k,\ell}, \quad \widetilde{h}_{k,\ell} = \widetilde{u}_{k+1,\ell} - \widetilde{u}_{k,\ell}.
\end{aligned} \tag{30}
$$

To also simplify the third term in (29), we may use the simple equality

$$
\sqrt{(c-a)^2 + (c-b)^2} = \sqrt{2} \sqrt{ \Big(c - \frac{a+b}{2} \Big)^2 + \Big(\frac{a-b}{2} \Big)^2 } .
$$

Therefore we can rewrite (29) as

$$
\begin{aligned}
F(c_{i,j}) &= \alpha \sum_{\ell=\ell_1}^{\ell_2} \sqrt{(c_{i,j} - h_{k_1-1,\ell})^2 + v_{k_1-1,\ell}^2} \; + \; \alpha \sum_{k=k_1}^{k_2-1} \sqrt{(c_{i,j} - v_{k,\ell_2})^2 + h_{k,\ell_2}^2} \\
&\quad + \; \alpha \sum_{\ell=\ell_1}^{\ell_2-1} \sqrt{(c_{i,j} - h_{k_2,\ell})^2 + v_{k_2,\ell}^2} \; + \; \alpha \sum_{k=k_1}^{k_2} \sqrt{(c_{i,j} - v_{k,\ell_1-1})^2 + v_{k,\ell_1-1}^2} \\
&\qquad + \; \alpha\sqrt{2} \sqrt{(c_{i,j} - \overline{v}_{k_2,\ell_2})^2 + \overline{h}_{k_2,\ell_2}^2} \; + \; \frac{1}{2} \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big(c_{k,\ell} - \widetilde{z}_{k,\ell} \Big)^2 \\
&= \alpha \sum_{\ell=\ell_1}^{\ell_2} \sqrt{(c_{i,j} - h_{k_1-1,\ell})^2 + v_{k_1-1,\ell}^2} \; + \; \alpha \sum_{k=k_1}^{k_2-1} \sqrt{(c_{i,j} - v_{k,\ell_2})^2 + h_{k,\ell_2}^2} \\
&\quad + \; \alpha \sum_{\ell=\ell_1}^{\ell_2-1} \sqrt{(c_{i,j} - h_{k_2,\ell})^2 + v_{k_2,\ell}^2} \; + \; \alpha \sum_{k=k_1}^{k_2} \sqrt{(c_{i,j} - v_{k,\ell_1-1})^2 + v_{k,\ell_1-1}^2} \\
&\qquad + \; \alpha\sqrt{2} \sqrt{(c_{i,j} - \overline{v}_{k_2,\ell_2})^2 + \overline{h}_{k_2,\ell_2}^2} \; + \; \frac{b^2}{2} (c_{i,j} - \widetilde{w}_{i,j})^2 \; + \; F_0(\widetilde{u}),
\end{aligned} \tag{31}
$$

where F_0 is not dependent on $c_{i,j}$ and

$$
\overline{v}_{k_2,\ell_2} = \frac{v_{k_2,\ell_2} + h_{k_2,\ell_2}}{2}, \qquad \overline{h}_{k_2,\ell_2} = \frac{v_{k_2,\ell_2} - h_{k_2,\ell_2}}{2}. \tag{32}
$$

Further we conclude that the local minimization problem for block (i, j) on level k amounts to minimizing the functional

$$\begin{aligned}\overline{F}(c_{i,j}) = \alpha \Bigg[&\sum_{\ell=\ell_1}^{\ell_2} \sqrt{(c_{i,j} - h_{k_1-1,\ell})^2 + v_{k_1-1,\ell}^2} \;+\; \sum_{k=k_1}^{k_2-1} \sqrt{(c_{i,j} - v_{k,\ell_2})^2 + h_{k,\ell_2}^2} \\ &+ \sum_{\ell=\ell_1}^{\ell_2-1} \sqrt{(c_{i,j} - h_{k_2,\ell})^2 + v_{k_2,\ell}^2} \;+\; \sum_{k=k_1}^{k_2} \sqrt{(c_{i,j} - v_{k,\ell_1-1})^2 + v_{k,\ell_1-1}^2} \\ &+ \sqrt{2}\sqrt{(c_{i,j} - \overline{v}_{k_2,\ell_2})^2 + \overline{h}_{k_2,\ell_2}^2}\,\Bigg] \;+\; \frac{b^2}{2}(c_{i,j} - \widetilde{w}_{i,j})^2. \end{aligned} \tag{33}$$

The solution of minimizing (33) can be sought using any iterative methods for univariable optimization; the Richardson and Newton iterations are reviewed in Appendix 1.

Formulation associated with a patch coarse level. As seen in Figure 1, Gauss–Seidel-type iterations from all levels of a multilevel method converge to an approximate solution with patches emerging in the iterate. The detection of a local patch (constant) can be done by checking the relative differences of a current approximation at a pixel point with its neighboring points. Let the general box (with general indices $k_1, \dots, k_2; \ell_1, \dots, \ell_2$) be as depicted in (28) with the only difference that $k_1 \neq (i-1)b + 1$, $k_2 \neq ib$, $\ell_1 \neq (j-1)b + 1$, $\ell_2 \neq jb$ in general for any i, j, k, as these general indices are explicitly detected. Let $b_1 = k_2 - k_1 + 1$ and $b_2 = \ell_2 - \ell_1 + 1$.

Then the local (varying) $b_1 \times b_2$ patch minimization proceeds similarly to (31) and (33) with the only essential change being replacing b^2 (the old patch size) by $b_1 b_2$ (the new patch size).

Solution on the coarsest level. On the coarse level $k = L + 1$, we look for a single constant update for the current approximation, i.e., $\min_c J(\widetilde{u} + c)$. Clearly the TV term remains the same for any c, so the functional can be written as

$$J(\widetilde{u} + c) = \alpha TV(\widetilde{u}) + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\widetilde{u}_{i,j} + c - z_{i,j})^2,$$

where $TV(\widetilde{u})$ denotes the first term in (11). Therefore the exact solution to the minimization with respect to c is

$$c = \text{mean}(z - \widetilde{u}), \qquad u_{\text{new}} = \widetilde{u} + \text{mean}(z - \widetilde{u}). \tag{34}$$

The 2D multilevel algorithm. We are ready to state our proposed piecewise constant-based multilevel method for solving (11).

Algorithm 2. *Given z and an initial guess $\widetilde{u} = z$ with $L + 1$ levels, do the following:*

(1) *Let $u_0 = \widetilde{u}$.*

(2) *Smooth the approximation on the finest level 1; i.e., solve (27) for $j, \ell = 1, 2, \dots, n$.*

(3) *On coarse levels $k = 2, 3, \dots, L + 1$, do the following:*
- — *compute $\widetilde{z} = z - \widetilde{u}$ via (30);*
- — *compute the local mean $\widetilde{w}_{i,j} = mean\big(\widetilde{z}\big((k-1)b + 1 : kb, (\ell - 1)b + 1 : \ell b\big)\big)$ via (30);*
- — *compute $\widetilde{v}_{k,\ell}$, $\widetilde{h}_{k,\ell}$ via (32);*
- — *solve (33) by solving local minimization as in (38) if $k \leq L$ or*

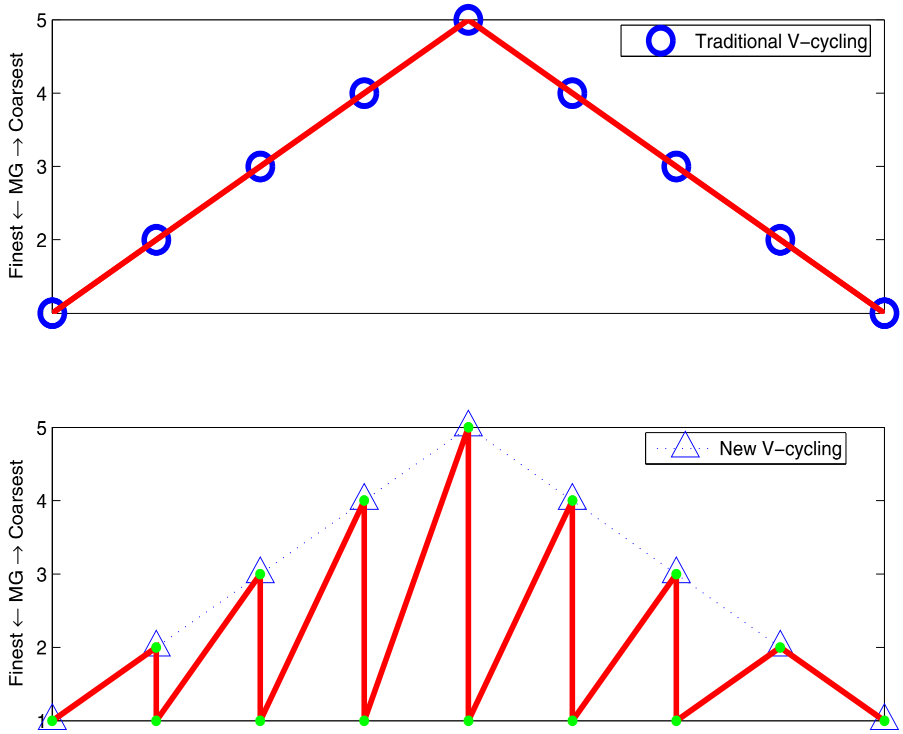

FIG. 6. *V-cycling comparison. Top: the normal MGM V-cycling. Bottom: the new V-cycling. Clearly one can see two differences:* (1) *the new V-cycling has a simple interpolation scheme;* (2) *each level of the new V-cycling interacts with the finest level directly.*

— *on the coarsest level $k = L + 1$, the correction constant is simply*
$$c = mean(\widetilde{w}) = mean(z - \widetilde{u});$$
— *add the correction $\widetilde{u} = \widetilde{u} + P_k c$ via* (26).

(4) *On level $k = 1$, check the possible patch size for each position (i, j): This is done by comparing the neighboring consecutive pixels (i_ℓ, j_ℓ) of pixel (i, j); i.e.,*

$$patch = \{(i_\ell, j_\ell) \mid |u_{i_\ell,j_\ell} - u_{i,j}| < \varepsilon\}$$

for some small ε (usually $\varepsilon = 10^{-3}$). Implement the piecewise constant update as with step (3).

(5) *If $\|\widetilde{u} - u_0\|_2$ is small enough, stop or return to step* (1).

Although Algorithm 2 (and Algorithm 1) uses the V-cycling-like pattern in a multilevel setting, the precise manner of information transfers is different, as illustrated by Figure 6. It is also feasible to develop other cycling patterns, e.g., the W-cycling, as described in [26]. In fact, some (though small) improvements have been observed with W-cycling.

We remark that there exists related work on solving a global minimization (or a nonlinear system) by block Gauss–Seidel alternating optimization methods [6, 33, 50, 67]. Such applications are similar to the use of a fixed coarse level to some extent. However, although our methods are not the standard block Gauss–Seidel alternating optimization methods, they may be viewed as adaptively partitioned alternating optimization methods.

3. A convergence analysis. To prove the convergence of Algorithm 2 (and its 1D version Algorithm 1), we have to show some monotone decay of the optimization object functional and address the issue of global minimization. It should be remarked that if one solves the regularized formulation (7) with $\beta > 0$, then the functional is differentiable (and in C^∞), so the analysis done in [43, 44, 63, 64, 65, 66] may be adapted for such a purpose. However, we aim to solve the unregularized formulation (4), which has a nondifferentiable functional, so we shall develop a direct approach of analysis. In fact, as our experiments have shown, the standard MGM does not work for (4) anyway.

For readers' benefit, we recall some well-known convergence results first.

THEOREM 3.1 (known results). *Consider the problem* $\min_{u\in\mathbb{R}^n} g(u)$*, where* $g : D \subset \mathbb{R}^n \to \mathbb{R}$ *solved by the descent method* $x^{k+1} = x^k + \alpha_k p^k$ *using a sequence of search directions* $\{p^j\}$*'s or if* $p^j = e_j$*,*

$$x^{k+1} = \arg\min_{\xi_j} g(x_1^k, x_2^k, \ldots, \xi_j, \ldots, x_n^k).$$

Let $D_o \subset D$ *be a compact set.*

1. *If* g *is continuous and hemivariate on* D_o*, then every strongly downward sequence* $\{x^k\}$ *(similar to* (35) *but stronger) satisfies that* $\lim_{k\to\infty}(x^k - x^{k+1}) = 0$*; i.e., no such sequences will oscillate or do not converge. See* [50, *Chap.* 14]*. (Note that the converged point may not be a local or global minimizer.)*
2. *If* D_o *is convex and* g *is continuously differentiable on* D *and uniformly convex in* D_o*, then the sequence* $\{x^k\}$ *converges to the global minimizer of* g *in either of these cases:*
   (a) *if all iterates* $\{x^k\}$ *lie in* D_o*, the sequence* $\{p^j\}$ *is free-steering (i.e., whose span covers* R^n*) and satisfy* $g(x^{k+1}) \le g(x^k)$ *(see* [50, *Chap.* 14]*);*
   (b) *if* g *is hemivariate in* D_o *(see* [5, 38, 67]*).*

It turns out that for our problems, differentiability amounts to checking and overcoming nonhemivariateness associated with our TV term, which is in turn linked with our patch-detection step of Algorithms 1 and 2. Without differentiability or hemivariateness, a converged solution might be a nonstationary point, which is essentially observed in Figure 1.

REMARK 2. *Let* u_{old} *be the current approximation before the* k*th iteration associated with a local block with index* I_i *and* u_{new} *be the new approximation after the* k*th iteration on* I_i*. Then we see that the minimizations from Algorithms* 1 *and* 2 *satisfy*

$$J(u_{new}) \le J(u_{old}). \tag{35}$$

Note that for continuously differentiable functionals, this decaying property (35) *is sufficient to guarantee a global minimization* [50]*, as our search directions (the coordinate unit vectors) form a free-steering sequence of nonzero vectors, but our* J *is nondifferentiable.*

Now we consider the convergence of Algorithms 1 and 2. We first consider the ideal case in one dimension where no constant patches are present in the solution.

LEMMA 3.2. *Assume the true minimizer* u *to the problem* (10) *is hemivariate; i.e.,* $u_i \ne u_{i+1}$ *for any* i*. Then the solution from Algorithm* 1*, even with* $L = 0$*, will be the global minimizer.*

Proof. We shall use the elementary decomposition approach. Since $u_i \neq u_{i+1}$ for any i, we may assume that $|u_i - u_{i+1}| = s_i u_i - s_i u_{i+1}$, where the scalar (sign) quantity $s_i = 1$ or -1 (but not 0) and s_i does not depend on values of u. Therefore we can rewrite the functional in (10) as

$$J(u) = \alpha \sum_{j=1}^{n-1} |u_j - u_{j+1}| + \frac{1}{2} \sum_{j=1}^{n} (u_j - z_j)^2 = \sum_{j=1}^{n} \phi_j(u_j),$$

where $\phi_j(u_j) = \alpha(s_j - s_{j-1})u_j + \frac{1}{2}(u_j - z_j)^2$ is *independent* of other ϕ's with $s_0 = 0$. Furthermore we can connect the local minimization to the global minimization directly:

$$\min_u J(u) = \min_u \sum_{j=1}^{n} \phi_j(u_j) = \sum_{j=1}^{n} \min \phi_j(u_j).$$

Here each local minimization is done locally in Algorithm 1; this proves the lemma. □

If we relax the assumption of nonexistence of local patches, we can refine the above result.

LEMMA 3.3. *Let the true minimizer u to problem* (10) *possess ℓ flat patches ($\ell < n$) with the kth patch I_k of the size τ_k such that $\sum \tau_k = \ell$. Assume that all such patches are correctly identified (but do not necessarily have the correct pixel values). Then the solution from Algorithm* 1 *will be the global minimizer.*

Proof. The proof is similar to that of the previous lemma. Let the true solution be denoted by

$$u = [u_1 \ \ldots \ u_n] = [u_{\tau_1}, \ldots, u_{\tau_1};\ u_{\tau_2}, \ldots, u_{\tau_2};\ \cdots;\ u_{\tau_\ell}, \ldots, u_{\tau_\ell}].$$

Then clearly $u_{\tau_i} \neq u_{\tau_{i+1}}$ for any i. As the functional can be rewritten as

$$\begin{aligned} J(u) &= \alpha \sum_{j=1}^{n-1} |u_j - u_{j+1}| + \frac{1}{2} \sum_{j=1}^{n} (u_j - z_j)^2 \\ &= \alpha \sum_{j=1}^{\ell-1} |u_{\tau_j} - u_{\tau_{j+1}}| + \frac{1}{2} \sum_{j=1}^{\ell} \tau_j (u_{\tau_j} - \bar{z}_j)^2 + \frac{1}{2} \sum_{j=1}^{\ell} \left[\tilde{z}_j^2 - \tau_j \bar{z}_j^2\right], \end{aligned}$$

where $\bar{z}_j = 1/\tau_j \sum_{t \in I_j} z_t$ and $\tilde{z}_j^2 = \sum_{t \in I_j} z_t^2$. Then we can follow the decomposition idea used previously to show that

$$\min_u J(u) = \sum_{j=1}^{\ell} \min \phi_j(u_{\tau_j})$$

with ϕ_j defined similarly, as done in the patch step of Algorithm 1. □

Next we analyze Algorithm 2. The difficulty here is that we cannot find a decomposition of the underlying functional to derive a simple theory. However, we can use the established theory from minimization [50] under suitable conditions.

LEMMA 3.4. *Algorithm* 2 *yields a global minimizer to* (11) *under any one of the following conditions:*

1. *The solution u to problem* (11) *is hemivariate; i.e., $u_{i,j} \neq u_{i,j+1}$, $u_{i,j} \neq u_{i+1,j}$ for any (i,j), and all iterates (in vector form) of Algorithm* 2 *lie in a compact and convex set $D_o \subset \mathbb{R}^{n^2}$ that contains the solution.*
2. *The solution u to problem* (11) *has ℓ patches ($\ell < n^2$) that can be identified, and all iterates (in vector form) from Algorithm* 2 *lie in a compact and convex set $D_o \subset \mathbb{R}^{\ell}$ that contains the solution.*

Proof. Here hemivariateness of u is associated with Algorithm 2 detecting no constant patches anywhere. With this assumption in part 1, the functional $J(u)$ is continuously differentiable in a neighborhood of the true solution. Note that the search directions of local minimization are unit coordinate vectors, so they form a free-steering sequence. Together with the decaying property (35), convergence to the global minimizer is immediate; refer to Theorem 3.1.

The proof of part 2 follows from part 1 in a way similar to Lemma 3.3 following Lemma 3.2; our Algorithm 2 essentially merges variables on a patch into one. □

It is of interest to point out that detecting the patches in our algorithms is not difficult. As shown by Lemma 3.2, indeed, Algorithm 1 can find the local minimizers (also global) accurately (even within one sweep) if there are no patches nearby. Therefore fine level local minimizations can reveal the patches quickly, as illustrated in Figure 1.

4. An application to a related restoration model.

A related and interesting result from Lemma 3.2 is stated in the following. Consider the following nonrotationally invariant model [46, 51]:

$$\min_u J(u), \qquad J(u) = \int_\Omega \left[\overline{\alpha}\big(|u_x| + |u_y|\big) + \frac{1}{2}(Ku - z)^2 \right] dxdy, \tag{36}$$

whose discrete counterpart is

$$\min_{u\in\mathbb{R}^{n\times n}} J(u) = \min_{u\in\mathbb{R}^{n}} \alpha \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} |u_{i,j} - u_{i,j+1}| + |u_{i,j} - u_{i+1,j}| + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(u_{i,j} - z_{i,j})^2. \tag{37}$$

This model is known to exhibit interesting characteristics in solution quality quite opposite to the TV model [51], depending on the image.

LEMMA 4.1. *Algorithm* 2 *applying to* (37) *converges to the global minimizer. Moreover, if there are no patches in u, Algorithm* 2 *can find the global minimizer in one sweep over the finest level.*

We also remark that several authors [37, 28, 16, 42] have studied very efficient solution methods of models similar to (36) via developing methods for solving Markov random field (MRF) problems. It will be of interest to consider multilevel methods based on MRF methods [16]. See also [30] for use of interior point approaches.

Readers who are familiar with optimization techniques may wonder how our low-dimensional subproblems (relaxations) are related to line search methods. Below we give such details in Appendix 2.

5. Complexity analysis.

The effectiveness of any iterative solver can be measured by the total number of iterative steps (hopefully not many) it takes to yield a solution up to an acceptable accuracy and the cost associated with a single iterative step. Our analysis below shows that the cost per step is $O(LN)$, where $L + 1$ is the number of levels employed and N is the dimension of our problems; i.e., $N = n$ in one dimension (Algorithm 1) and $N = n^2$ in two dimensions (Algorithm 2). We remark that the complexity of the robust algorithm in [31] has the complexity $O(N\sqrt{N})$.

First, we analyze Algorithm 1. Let $N = n = 2^{L_m}$ and $L \le L_m$. Denote as before $b_k = 2^{k-1}$ and $\tau_k = N/b_k$. We can work out the number of floating point operations (flops) as follows:

	Quantities	Flop counts
Level k	$d_j,\ \widetilde{d}_j,\ \bar{c}_j,\ \bar{z}_j$	$4\tau_k$
	$\widetilde{z},\ \widetilde{\omega}$	$2N$
	s smoothing steps	$4s\tau_k$
All levels	$W_1 = \sum_{k=1}^{L+1}\Big(2N + 4\tau_k + 4s\tau_k\Big)$	

Therefore we can bound the total number of flops by

$$
\begin{aligned}
W_1 &= \sum_{k=1}^{L+1}\left(2N + \frac{4N}{b_k} + \frac{4sN}{b_k}\right) = 2(L+1)N + 4N(s+1)\sum_{k=0}^{L}\frac{1}{2^k}\\
&< 2(L+1)N + 8N(s+1) \approx O(LN).
\end{aligned}
$$

Second, for Algorithm 2, the notation is changed only slightly; here $N = n^2$ and set $\tau_k = N/b_k^2$ with $b_k = 2^{k-1}$. Then the flop counts are as follows:

	Quantities	Flop counts
Level k	$h_{i,j},\ v_{i,j}$	$4b_k\tau_k$
	$\widetilde{z}_{i,j},\ \widetilde{\omega}_{i,j}$	$2N$
	s smoothing steps	$(24b_k + 1)s\tau_k$
All levels	$W_2 = \sum_{k=1}^{L+1}\Big(2N + 4b_k\tau_k + (24b_k+1)s\tau_k\Big)$	

Therefore the upper bound is the following:

$$
\begin{aligned}
W_2 &= 2(L+1)N + \sum_{k=1}^{L+1}\left(\frac{4(6s+1)}{b_k} + \frac{s}{b_k^2}\right) = 2(L+1)N + \sum_{k=0}^{L}\left(\frac{4(6s+1)}{2^k} + \frac{s}{4^k}\right)\\
&< 2(L+1)N + \left(2 + \frac{4s}{3}\right)N \approx O(LN).
\end{aligned}
$$

Note that $\max(L) = \log n = O(\log N)$. This "extra" complexity, although practically insignificant, arises from the cycling pattern, as shown in the bottom plot of Figure 6; i.e., each coarse level interacts with the finest level directly and brings $O(\log N)$ work per interaction. One way to get rid of this $O(\log N)$ is to modify the cycling pattern from $k = 1, 2, \ldots, L+1$ to $k = 1, L+1, L, \ldots, 2$ and to compute the quantity $\widetilde{\omega}_{i,j}$ without forming $\widetilde{z}_{i,j}$ explicitly (as $\widetilde{\omega}_{i,j}$ is changed only by a constant on each block at a new level k in the new cycling order).

6. Numerical experiments.

In this section, we shall test several aspects of the proposed multilevel method. First, we demonstrate the convergence of the method for several test cases and show that the proposed method gives fast and quality solutions in both one and two dimensions. Second, we engage in the challenging task of

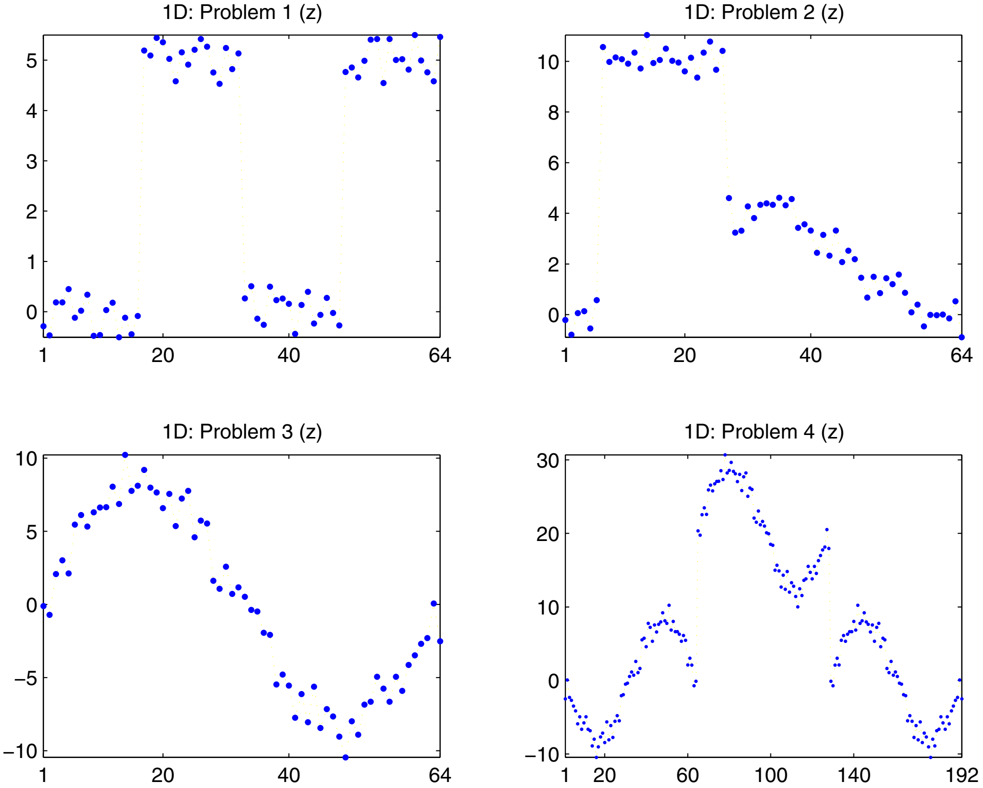

FIG. 7. *Four 1D test examples.*

comparing it to the well-known CGM [23] to show some advantages of our MGM approaches. The latter is particularly pleasing, as our MGM can give sharper solutions in addition to delivering a solution at a faster speed.

Fast convergence. We first consider four 1D denoising problems with the signal-to-noise ratio (SNR; see [23]) of 10, as shown in Figure 7. The processed solutions by our multilevel algorithm are shown in Figure 8, where one observes that the method is extremely fast (as expected of a multilevel method), converging to the tolerance of $tol = 10^{-4}$ in, respectively, 4, 4, 4, 4 multilevel steps; here α takes the values of 7.68×10^{-5}, 19.2, 160, 25.6, respectively, for the four examples. We then test further four examples of 2D denoising problems, as shown in Figure 9, where SNR = 5. The computed solutions by our multilevel algorithm are shown in Figure 10. Again one observes that the method is quick to obtain solutions which are comparable to other methods that we tested. We next consider details of one comparison.

Comparison with CGM. As remarked, the method CGM [23] has been known to produce the most reliable solutions (in terms of speed and reduction of merit functional) when compared to the time-marching and fixed iteration methods. We shall compare with it only for suitable parameter $\beta > 0$; in particular if $\beta = 0$ (or nearly 0), CGM does not converge. We remark that there exist other methods [36, 15, 31] that have not been compared; a future evaluation of all such methods will be valuable.

As an exact solution is not available in general, any comparison must be based on

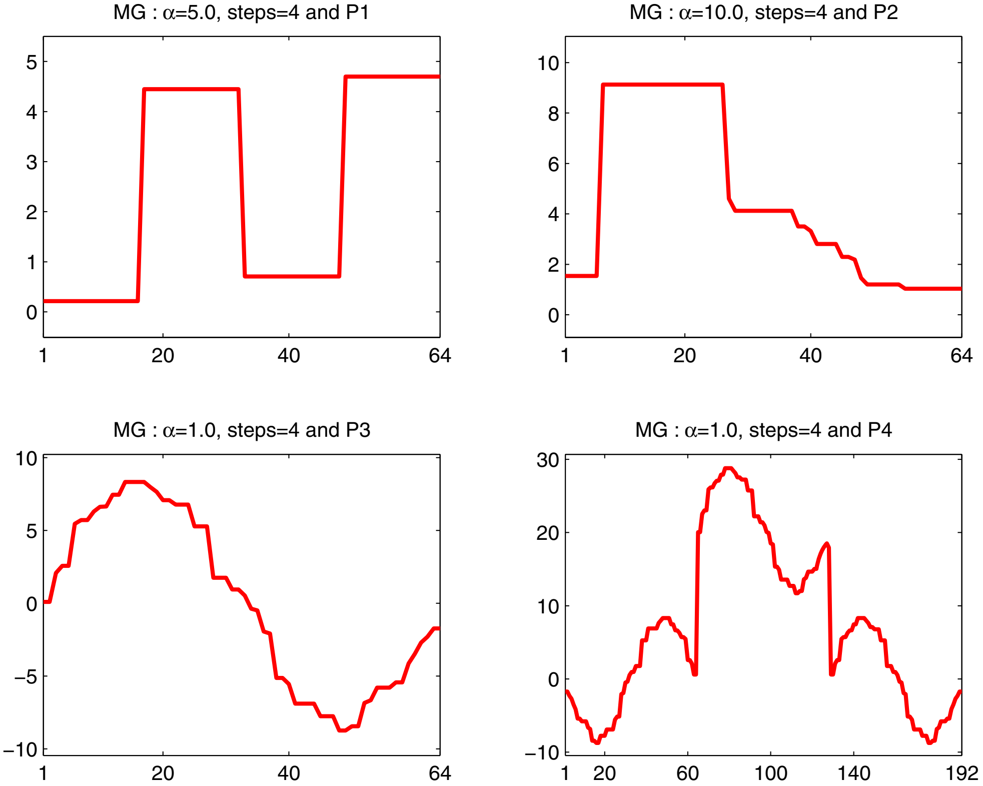

FIG. 8. *New MGM for the four 1D test examples.*

other measures. What we propose to compare is the value of the minimizing function $J(u)$ as in (10) for one dimension. From Table 1, one can see that our method is quite competitive with CGM in terms of reduction of $J(u)$, and CGM depends heavily on the choice of β. For the 2D examples, we take the parameter $\gamma = \beta$, as we intend to compare our new multilevel solutions with the CGM. For the image size 128×128 (for all the above four problems), we observe the following results in Table 2. Clearly the MGM is much less sensitive to the parameter γ than the CGM, and of course it may be possible to replace the 2D local minimization by some new methods, e.g., [52], and then we expect that the multilevel approach improves further.

Next we illustrate what the improvements over CGM solutions look like in terms of image sharpness. We shall take problem 2 (1D) as an example. With $\beta = 10^{-10}$, we have seen that $J(u_{CGM}) = 185.413$, while $J_{MGM} = 185.407$ with the data size $n = 64$ (refer to the top right plot of Figure 7). However, if we zoom in on two locations of the solutions, we can see an interesting (but expected) difference between CGM and MGM; i.e., the multilevel solution is sharper (more like a staircase function) than the CGM. This is visible from Figure 11 near node 28 and Figure 12 near node 55. Clearly the CGM solutions have smoother solutions (though almost piecewise constants).

Finally, we show some results in Table 3 comparing the speed of our method with CGM [23] and in Table 4 illustrating the mild dependence of our method on (especially large) α where solution patches dominate. Clearly as the problem size increases, our MGM performs faster, and as expected for larger α, our MGM takes

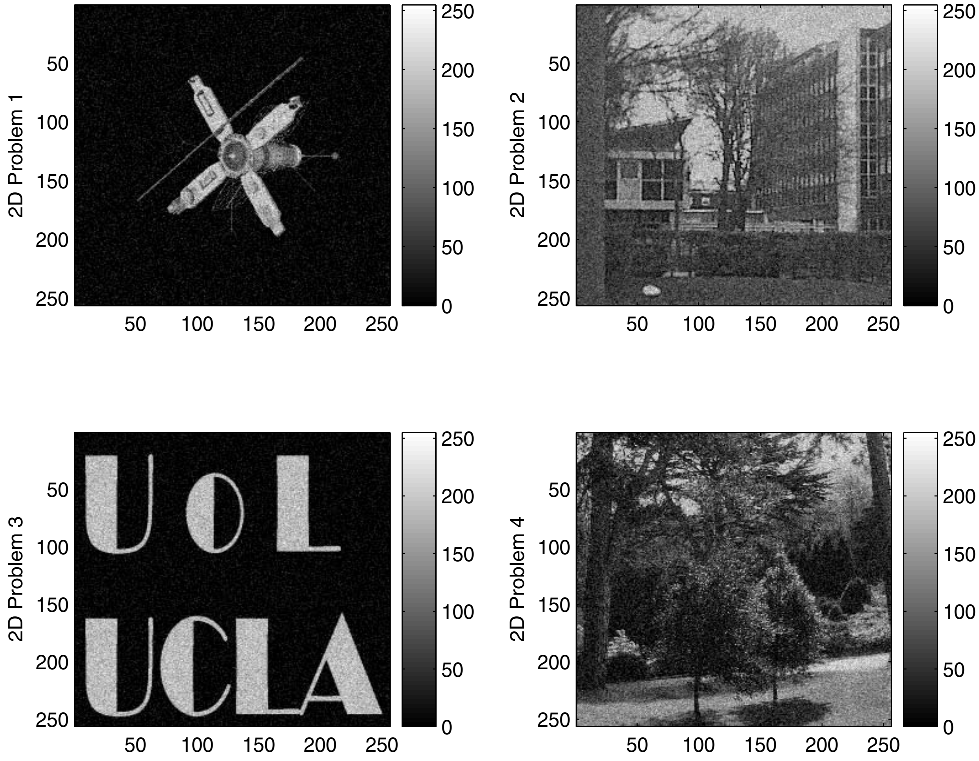

FIG. 9. *Four 2D test examples.*

more MG cycles to detect the underlying patches (recall that when $\alpha \to \infty$, all TV solutions are global constants).

7. Conclusions. This paper proposed a new and effective multilevel method based on piecewise constant refinements of the minimization problem modeling the image denoising problem in the TV norm. We have demonstrated that standard multilevel methods do not work all the time. Our new method is different from those multigrid methods applied to the PDEs and also from others which assume the minimizing functional is differentiable. The apparently nonsmooth solutions from primal relaxations of local minimization are corrected using multilevels and varying size coarse level elements.

As our method is less sensitive to parameters, as a by-product (when compared to the well-known CGM [23] method), there is strong evidence to suggest that sharper solutions are obtained in many cases in the true spirit of TV norms. On the other hand, our method also works for variational models with a smooth object functional.

Appendix 1—Local minimization formulation. We first use the Richardson iteration to solve (27) from repeating $\widetilde{u} \to u \to \widetilde{u}$:

$$\alpha \Bigg[\frac{2u_{i,j} - \widetilde{u}_{i+1,j} - \widetilde{u}_{i,j+1}}{\sqrt{(u_{i,j} - \widetilde{u}_{i+1,j})^2 + (u_{i,j} - \widetilde{u}_{i,j+1})^2 + \gamma}}$$
$$+ \frac{u_{i,j} - \widetilde{u}_{i-1,j}}{\sqrt{(u_{i,j} - \widetilde{u}_{i-1,j})^2 + (\widetilde{u}_{i-1,j} - \widetilde{u}_{i-1,j+1})^2 + \gamma}}$$

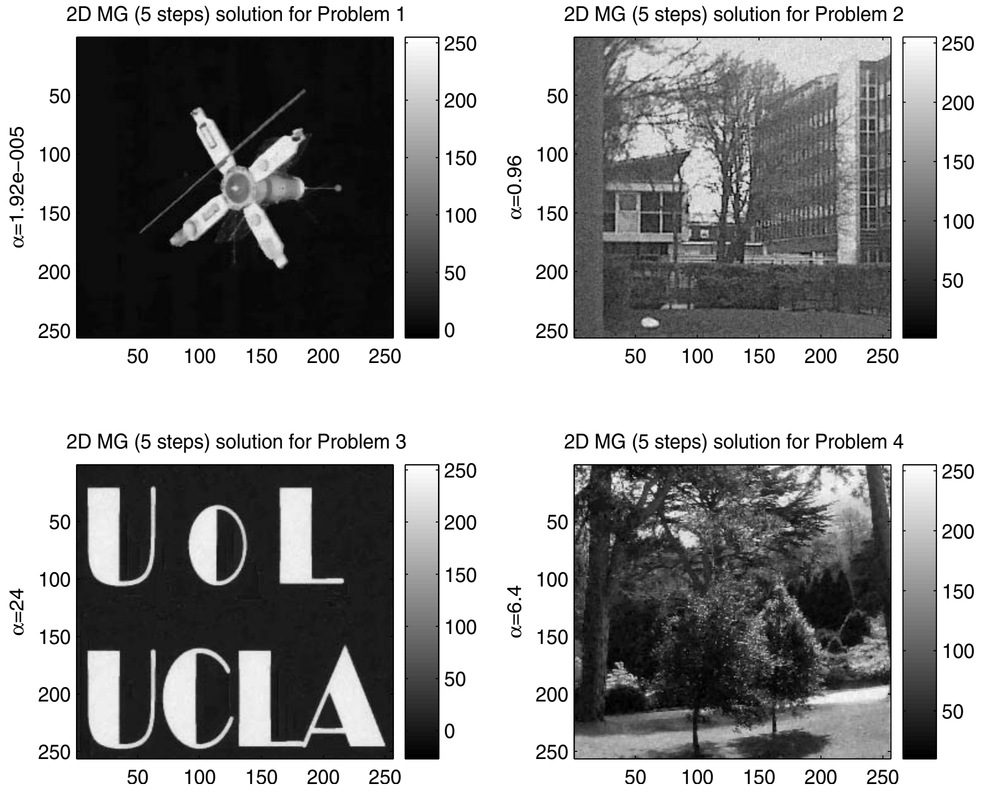
2D MG (5 steps) solution for Problem 1
α=1.92e−005
50
100
150
200
250
0
2D MG (5 steps) solution for Problem 2
α=0.96
2D MG (5 steps) solution for Problem 3
α=24
2D MG (5 steps) solution for Problem 4
α=6.4

FIG. 10. *New multilevel solutions of the* 2D *test examples.*

$$+\frac{u_{i,j}-\widetilde{u}_{i,j-1}}{\sqrt{(u_{i,j}-\widetilde{u}_{i,j-1})^2+(\widetilde{u}_{i,j-1}-\widetilde{u}_{i+1,j-1})^2+\gamma}}\Bigg]+(u_{i,j}-z_{i,j})=0,$$

where $\gamma > 0$ is a regularizing parameter which is much taken to be smaller than β above (e.g., $\gamma = 10^{-30}$, which is far too small for β in (6)). Then our Richardson iterations for solving (27) take the form

$$u_{i,j}^{new} = R^{old}\Big/L^{old}, \tag{38}$$

where

$$\begin{aligned} L^{old} &= 2\alpha\Big/\sqrt{(u_{i,j}^{old}-\widetilde{u}_{i+1,j})^2+(u_{i,j}^{old}-\widetilde{u}_{i,j+1})^2+\gamma} \\ &\quad+\alpha\Big/\sqrt{(u_{i,j}^{old}-\widetilde{u}_{i-1,j})^2+(\widetilde{u}_{i-1,j}-\widetilde{u}_{i-1,j+1})^2+\gamma} \\ &\quad+\alpha\Big/\sqrt{(u_{i,j}^{old}-\widetilde{u}_{i,j-1})^2+(\widetilde{u}_{i,j-1}-\widetilde{u}_{i+1,j-1})^2+\gamma}\ +1, \\ R^{old} &= \alpha(\widetilde{u}_{i+1,j}+\widetilde{u}_{i,j+1})\Big/\sqrt{(u_{i,j}^{old}-\widetilde{u}_{i+1,j})^2+(u_{i,j}^{old}-\widetilde{u}_{i,j+1})^2+\gamma} \\ &\quad+\alpha\widetilde{u}_{i-1,j}\Big/\sqrt{(u_{i,j}^{old}-\widetilde{u}_{i-1,j})^2+(\widetilde{u}_{i-1,j}-\widetilde{u}_{i-1,j+1})^2+\gamma} \\ &\quad+\alpha\widetilde{u}_{i,j-1}\Big/\sqrt{(u_{i,j}^{old}-\widetilde{u}_{i,j-1})^2+(\widetilde{u}_{i,j-1}-\widetilde{u}_{i+1,j-1})^2+\gamma}\ +z_{i,j}. \end{aligned}$$

Table 1
Comparison of our new MGM with CGM [23] *in* 1*D.*

Method	Problem	Convergence steps	J(u)
CGM $\beta = 10^{-1}$	1	8	82.5160
	2	7	207.648
	3	6	34.1475
	4	6	135.143
CGM $\beta = 10^{-5}$	1	11	70.6723
	2	17	185.745
	3	11	32.8829
	4	11	131.433
CGM $\beta = 10^{-10}$	1	15	70.4687
	2	22	185.413
	3	13	32.8668
	4	11	131.389
CGM $\beta = 10^{-14}$	1	16	70.4680
	2	22	185.409
	3	13	32.8668
	4	11	131.390
MG (new)	1	2	70.4680
	2	4	185.407
	3	4	32.8664
	4	4	131.385

Table 2
Comparison of our new MGM with CGM [23] *in* 2*D.*

Method	Problem	Convergence steps	J(u)
CGM $\beta = 10^{-1}$	1	2	1.23384E-5
	2	7	6.59000E5
	3	10	2.55551E6
	4	10	4.31471E6
MGM (new)	1	1	4.60572E-6
	2	3	6.59045E5
	3	5	2.67872E6
	4	5	4.31899E6
CGM $\beta = 10^{-5}$	1	3	1.17532E-5
	2	10	6.58996E5
	3	22	2.55007E6
	4	16	4.31415E6
MGM (new) $\beta = 10^{-5}$	1	1	4.58596E-6
	2	5	6.59045E5
	3	11	2.70140E6
	4	8	4.31898E6
CGM $\beta = 10^{-10}$	1	7	4.50816E-6
	2	15	6.58996E5
	3	40	2.55002E6
	4	21	4.31414E6
MGM (new) $\beta = 10^{-10}$	1	3	4.24560E-6
	2	7	6.59045E5
	3	20	2.68725E6
	4	10	4.31898E6

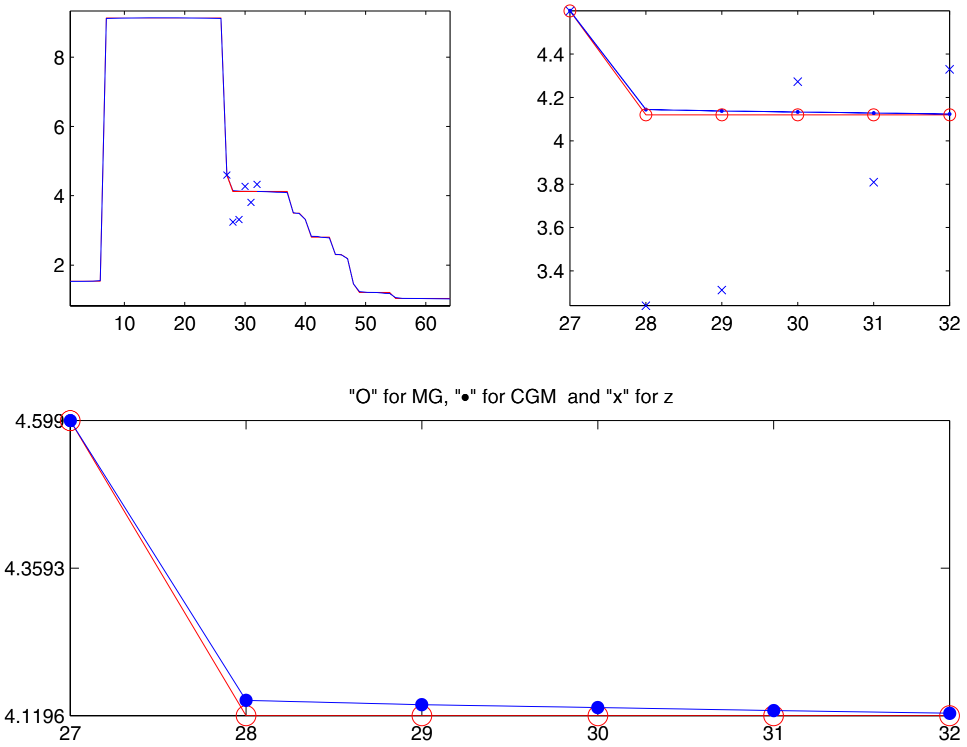

FIG. 11. *Detailed comparison of CGM and MGM solutions near node* 27 *(Problem* 2*).*

An alternative to the Richardson iterations is the Newton method, which we found to perform similarly, as shown below:

$$u_{i,j}^{new} = u_{i,j}^{old} - T^{old}/B^{old},$$

where

$$\begin{aligned} T^{old} = {} & \alpha \frac{2u_{i,j}^{old} - \widetilde{u}_{i+1,j} - \widetilde{u}_{i,j+1}}{\sqrt{(u_{i,j}^{old} - \widetilde{u}_{i+1,j})^2 + (u_{i,j}^{old} - \widetilde{u}_{i,j+1})^2 + \gamma}} \\ & + \alpha \frac{u_{i,j}^{old} - \widetilde{u}_{i-1,j}}{\sqrt{(u_{i,j}^{old} - \widetilde{u}_{i-1,j})^2 + (\widetilde{u}_{i-1,j} - \widetilde{u}_{i-1,j+1})^2 + \gamma}} \\ & + \alpha \frac{u_{i,j}^{old} - \widetilde{u}_{i,j-1}}{\sqrt{(u_{i,j}^{old} - \widetilde{u}_{i,j-1})^2 + (\widetilde{u}_{i,j-1} - \widetilde{u}_{i+1,j-1})^2 + \gamma}} + u_{i,j}^{old} - z_{i,j}, \end{aligned}$$

$$\begin{aligned} B^{old} = {} & \alpha \frac{2}{\sqrt{(u_{i,j}^{old} - \widetilde{u}_{i+1,j})^2 + (u_{i,j}^{old} - \widetilde{u}_{i,j+1})^2 + \gamma}} \\ & - \alpha \frac{(2u_{i,j}^{old} - \widetilde{u}_{i+1,j} - \widetilde{u}_{i,j+1})^2}{\left((u_{i,j}^{old} - \widetilde{u}_{i+1,j})^2 + (u_{i,j}^{old} - \widetilde{u}_{i,j+1})^2 + \gamma\right)^{3/2}} \end{aligned}$$

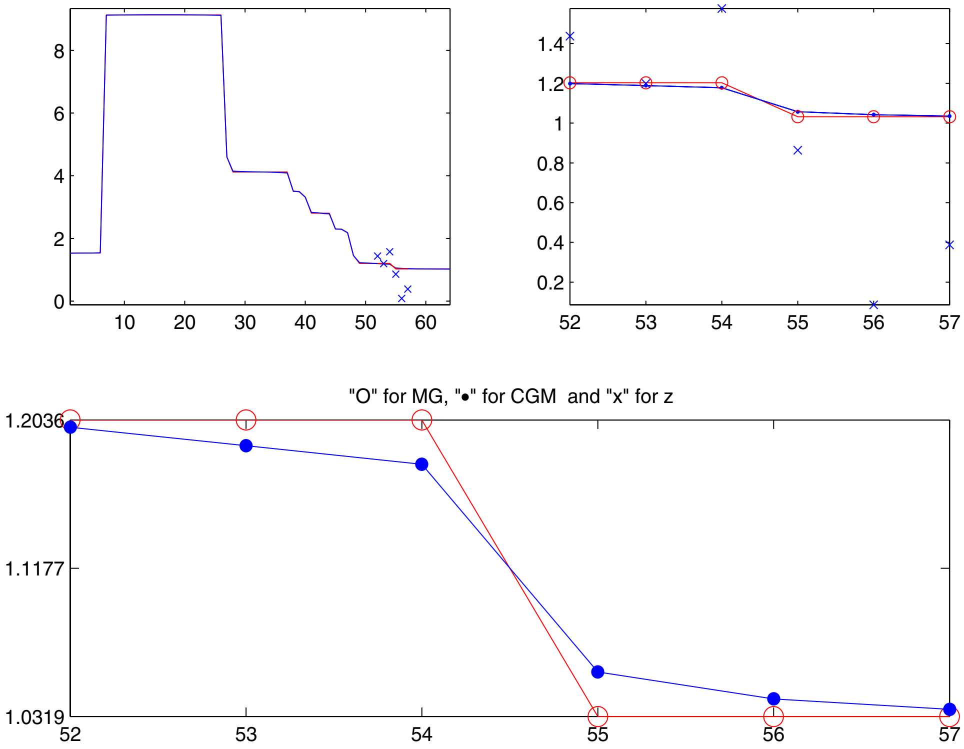

FIG. 12. *Detailed comparison of CGM and MGM solutions near node* 55 *(Problem* 2*).*

$$
\begin{aligned}
&+\alpha\frac{1}{\sqrt{(u_{i,j}^{old}-\widetilde{u}_{i-1,j})^2+(\widetilde{u}_{i-1,j}-\widetilde{u}_{i-1,j+1})^2+\gamma}}\\
&-\alpha\frac{(u_{i,j}^{old}-\widetilde{u}_{i-1,j})^2}{\Big((u_{i,j}^{old}-\widetilde{u}_{i-1,j})^2+(\widetilde{u}_{i-1,j}-\widetilde{u}_{i-1,j+1})^2+\gamma\Big)^{3/2}}\\
&+\alpha\frac{1}{\sqrt{(u_{i,j}^{old}-\widetilde{u}_{i,j-1})^2+(\widetilde{u}_{i,j-1}-\widetilde{u}_{i+1,j-1})^2+\gamma}}\\
&-\alpha\frac{(u_{i,j}^{old}-\widetilde{u}_{i,j-1})^2}{\Big((u_{i,j}^{old}-\widetilde{u}_{i,j-1})^2+(\widetilde{u}_{i,j-1}-\widetilde{u}_{i+1,j-1})^2+\gamma\Big)^{3/2}}\\
&+1.
\end{aligned}
$$

The full Newton method is known to converge slowly [7, 13, 70, 26] when α is relatively large and γ is small. Here the pointwise Newton appears to be adequate, but other optimization methods (see, e.g., [52]) may be considered to solve the above local minimization.

Appendix 2—Interpretation as a line search method. We give an alternative interpretation of the proposed method which may be of interest to readers in the optimization community. First, for Algorithm 1, the multigrid idea for solving

$$\min_{\lambda} J(u_k+\lambda\vec{s}), \qquad u^{(k)}=u_k+\lambda\vec{s},$$

TABLE 3
CPU comparison of our new MGM with CGM [23] *with* $\beta = 10^{-12}$ *and tol* $= 10^{-6}$.

Problem	Size n (for $n \times n$)	CPU by CGM	CPU by MGM
2	128	6.14	33.19
	256	117.95	133.38
	512	2192.59	548.19
3	128	343.78	33.06
	256	1418.30	133.94
	512	6149.64	567.88

TABLE 4
Dependence of our new MGM on α *for* 2*D Problems* 1–2 *of size* 256×256. *Here tol* $= 10^{-4}$.

Problem	α used	MGM cycles	Problem	α used	MGM cycles
1	3.84×10^{-6}	1	2	9.6×10^{-2}	2
	1.92×10^{-5}	1		1.92×10^{-1}	2
	1	2		9.60×10^{-1}	2
	10	4		9.6	4
	15	8		20	12

modifies the simple "alternating variables methods"

$$\vec{s} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \text{etc.}$$

to take a more systematic set of search directions

$$\vec{s} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{[level 0]}, \quad \vec{s} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{[level 1]}, \quad \vec{s} = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ \vdots \end{bmatrix} \text{[level 2]}, \quad \text{etc.}$$

Likewise, a patch (of first three nodes) may be denoted by the search direction

$$\vec{s} = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix}, \quad \text{etc.}$$

Second, for two dimensions, we may specify

$$\vec{s} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{[level 0]}, \quad \vec{s} = \frac{1}{2} \begin{bmatrix} \vec{s}_1 \\ \vec{s}_1 \end{bmatrix}, \quad \vec{s}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{[level 1]},$$

$$\vec{s} = \frac{1}{4}\begin{bmatrix}\vec{s}_2\\ \vec{s}_2\\ \vec{s}_2\\ \vec{s}_2\end{bmatrix}, \quad \vec{s}_2 = \begin{bmatrix}1\\ 1\\ 1\\ 1\\ 0\\ \vdots\\ 0\end{bmatrix}\text{[level 2]}, \quad \text{etc.}$$

Acknowledgment. The authors thank all three anonymous referees for constructive criticism and helpful suggestions.

REFERENCES

[1] R. Acar and C. R. Vogel, *Analysis of total variation penalty method for ill-posed problems*, Inverse Problems, 10 (1994), pp. 1217–1229.

[2] L. Alvarez, P.-L. Lions, and J.-M. Morel, *Image selective smoothing and edge detection by nonlinear diffusion.* II, SIAM J. Numer. Anal., 29 (1992), pp. 845–866.

[3] H. C. Andrews and B. R. Hunt, *Digital Image Restoration*, Prentice–Hall, Englewood Cliffs, NJ, 1977.

[4] L. Badea, X.-C. Tai, and J. Wang, *Convergence rate analysis of a multiplicative Schwarz method for variational inequalities*, SIAM J. Numer. Anal., 41 (2003), pp. 1052–1073.

[5] D. P. Bertsekas, *Nonlinear Programming*, Athena, Belmont, MA, 1995.

[6] J. C. Bezdek and R. J. Hathaway, *Convergence of alternating optimization*, Neural Parallel Sci. Comput., 11 (2003), pp. 351–368.

[7] P. Blomgren, T. F. Chan, P. Mulet, L. Vese, and W. L. Wan, *Variational PDE models and methods for image processing*, in Numerical Analysis 1999, Chapman Hall/CRC Res. Notes Math. 420, Chapman & Hall/CRC, Boca Raton, FL, 2000, pp. 43–67.

[8] A. Brandt, *Multigrid solvers and multilevel optimization strategies*, in Multiscale Optimization and VLSICAD, J. Cong and J. R. Shinnerl, eds., Kluwer Academic Publishers, Boston, 2002, pp. 1–69.

[9] T. Brox, M. Welk, G. Steidl, and J. Weickert, *Equivalence results for TV diffusion and TV regularisation*, in Scale-Space Methods in Computer Vision, Lecture Notes in Comput. Sci. 2695, L. D. Griffin, ed., Springer, Berlin, 2003, pp. 86–100.

[10] A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr, *A Multigrid Platform for Real-Time Motion Computation with Discontinuity-Preserving Variational Methods*, Technical Report 136, Saarland University, Saarbrücken, Germany, 2005.

[11] M. Burger, S. Osher, J. Xu, and G. Gilboa, *Nonlinear Inverse Scale Space Methods for Image Restoration*, CAM Report 05-34, University of California, Los Angeles, CA, 2005.

[12] D. Calvetti, B. Lewis, and L. Reichel, *Krylov subspace iterative methods for nonsymmetric discrete ill-posed problems in image restoration*, in Advanced Signal Processing Algorithms, Architectures, and Implementations X, Proc. SPIE 4116, F. T. Luk, ed., The International Society for Optical Engineering, Bellingham, WA, 2001, to appear.

[13] J. L. Carter, *Dual Method for Total Variation-Based Image Restoration*, CAM Report 02-13, University of California, Los Angeles, CA, 2002, http://www.math.ucla.edu/applied/cam/index.html.

[14] E. Casas, K. Kunisch, and C. Pola, *Regularization of functions of bounded variation and applications to image enhancement*, Appl. Math. Optim., 40 (1999), pp. 229–257.

[15] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision, 20 (2004), pp. 89–97.

[16] A. Chambolle, *Total variation minimization and a class of binary MRF models*, in Energy Minimization Methods in Computer Vision and Pattern Recognition, Lecture Notes in Comput. Sci. 3757, Springer, Berlin, 2005, pp. 136–152.

[17] A. Chambolle and P. L. Lions, *Image recovery via total variation minimization and related problems*, Numer. Math., 76 (1997), pp. 167–188.

[18] R. H. Chan, T. F. Chan, and W. L. Wan, *Multigrid for differential convolution problems arising from image processing*, in Scientific Computing, R. Chan, T. F. Chan, and G. H. Golub, eds., Springer, Singapore, 1997.

[19] R. H. Chan, T. F. Chan, and C. K. Wong, *Cosine transform based preconditioners for total variation minimization problems in image processing*, IEEE Trans. Image Process.,

8 (1999), pp. 1472–1478.

[20] R. H. Chan, Q.-S. Chang, and H.-W. Sun, *Multigrid method for ill-conditioned symmetric Toeplitz systems*, SIAM J. Sci. Comput., 19 (1998), pp. 516–529.

[21] T. F. Chan and K. Chen, *On a nonlinear multigrid algorithm with primal relaxation for the image total variation minimisation*, Numer. Alg., 41 (2006), pp. 387–411.

[22] T. F. Chan and S. Esedoḡlu, *Aspects of total variation regularized L^1 function approximation*, SIAM J. Appl. Math., 65 (2005), pp. 1817–1837.

[23] T. F. Chan, G. H. Golub, and P. Mulet, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM J. Sci. Comput., 20 (1999), pp. 1964–1977.

[24] T. F. Chan and P. Mulet, *Iterative Methods for Total Variation Restoration*, CAM Report 96-38, University of California, Los Angeles, CA, 1996, http://www.math.ucla.edu/applied/cam/index.html.

[25] Q. Chang and I.-L. Chern, *Acceleration methods for total variation-based image denoising*, SIAM J. Sci. Comput., 25 (2003), pp. 982–994.

[26] K. Chen, *Matrix Preconditioning Techniques and Applications*, Cambridge Monogr. Appl. Comput. Math. 19, Cambridge University Press, Cambridge, UK, 2005.

[27] K. Chen and X. C. Tai, *A Nonlinear Multigrid Method for Total Variation Minimization from Image Restoration*, UCLA CAM Report CAM05-26, University of California, Los Angeles, CA, 2005.

[28] J. Darbon and M. Sigelle, *A fast and exact algorithm for total variation minimization*, in Proceedings of the 2nd Iberian Conference on Pattern Recognition and Image Analysis, Lecture Notes in Comput. Sci. 3522, J. S. Marques, N. Pérez de la Blanca, and P. Pina, eds., Springer, Berlin, 2005, pp. 351–359.

[29] C. Frohn-Schauf, S. Henn, and K. Witsch, *Nonlinear multigrid methods for total variation image denoising*, Comput. Visual Sci., 7 (2004), pp. 199–206.

[30] H. Fu, M. K. Ng, M. Nikolova, and J. L. Barlow, *Efficient minimization methods of mixed $\ell 2$-$\ell 1$ and $\ell 1$-$\ell 1$ norms for image restoration*, SIAM J. Sci. Comput., 27 (2006), pp. 1881–1902.

[31] D. Goldfarb and W. Yin, *Second-order cone programming methods for total variation-based image restoration*, SIAM J. Sci. Comput., 27 (2005), pp. 622–645.

[32] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison–Wesley, Reading, MA, 1993.

[33] M. Heinkenschloss, M. B. Hribar, and M. Kokkolaras, *Acceleration of multidisciplinary analysis solvers by inexact subsystem simulations*, in Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, 1998, paper 98-4712.

[34] M. Hintermüller and K. Kunisch, *Total bounded variation regularization as a bilaterally constrained optimization problem*, SIAM J. Appl. Math., 64 (2004), pp. 1311–1333.

[35] M. Hintermüller and G. Stadler, *An infeasible primal-dual algorithm for total variation–based inf-convolution-type image restoration*, SIAM J. Sci. Comput., 28 (2006), pp. 1–23.

[36] W. Hinterberger, M. Hintermüller, K. Kunisch, M. von Oehsen, and O. Scherzer, *Tube methods for BV regularization*, J. Math. Imaging Vision, 19 (2003), pp. 219–235.

[37] D. Hochbaum, *An efficient algorithm for image segmentation, Markov random fields and related problems*, J. ACM, 48 (2001), pp. 686–701.

[38] J. Idier, *Convex half-quadratic criteria and interacting auxiliary variables for image restoration*, IEEE Trans. Image Process., 10 (2001), pp. 1001–1009.

[39] K. Ito and K. Kunisch, *An active set strategy based on the augmented Lagrangian formulation for image restoration*, M2AN Math. Model. Numer. Anal., 33 (1999), pp. 1–21.

[40] T. Kärkkäinen and K. Majava, *Nonmonotone and monotone active set methods for image restoration* II. *Numerical results*, J. Optim. Theory Appl., 106 (2000), pp. 81–105.

[41] T. Kärkkäinen, K. Majava, and M. M. Mäkelä, *Comparison of Formulations and Solution Methods for Image Restoration Problems*, Series B Report B 14/2000, University of Jyväskylä, Jyväskylä, Finland, 2000.

[42] V. Kolmogorov, *Primal-Dual Algorithm for Convex Markov Random Fields*, Microsoft Technical Report MSR-TR-2005-117, Microsoft Corporation, Redmond, WA, 2005.

[43] R. Kornhuber, *Monotone multigrid methods for variational inequalities* II, Numer. Math., 72 (1996), pp. 481–499.

[44] R. Kornhuber, *On constrained Newton linearization and multigrid for variational inequalities*, Numer. Math., 91 (2002), pp. 699–721.

[45] S. H. Lee and J. K. Seo, *Noise removal with Gauss curvature driven diffusion*, IEEE Trans. Image Process., 14 (2005), pp. 904–909.

[46] Y. Y. Li and F. Santosa, *A computational algorithm for minimizing total variation in image*

restoration, IEEE Trans. Image Process., 5 (1996), pp. 987–995.

[47] E. Mammen and S. van de Geer, *Locally adaptive regression splines*, Ann. Statist., 25 (1997), pp. 387–413.

[48] 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., 22 (2000), pp. 387–405.

[49] S. Nash, *A multigrid approach to discretized optimisation problems*, J. Opt. Methods Softw., 14 (2000), pp. 99–116.

[50] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, London, 1970; reprinted, SIAM, Philadelphia, 2000.

[51] S. Osher and S. Esedoglu, *Decomposition of images by the anisotropic Rudin-Osher-Fatemi model*, Comm. Pure Appl. Math., 57 (2004), pp. 1609–1626.

[52] M. J. D. Powell, *The NEWUOA software for unconstrained optimization without derivatives*, in Large-Scale Nonlinear Optimization, Springer, New York, 2006, pp. 255–297.

[53] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, UK, 1992.

[54] E. Radmoser, O. Scherzer, and J. Schöberl, *A Cascadic Algorithm for Bounded Variation Regularization*, SFB Report 00-23, Johannes Kepler University, Linz, Austria, 2000.

[55] K. L. Riley, *Two-Level Preconditioners for Regularized Ill-Posed Problems*, Ph.D. thesis, Montana State University, Bozeman, MT, 1999.

[56] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.

[57] G. Sapiro, *Geometrical Differential Equations and Image Analysis*, Cambridge University Press, Cambridge, UK, 2001.

[58] J. Savage and K. Chen, *An improved and accelerated nonlinear multigrid method for total-variation denoising*, Int. J. Comput. Math., 82 (2005), pp. 1001–1015.

[59] G. Steidl, J. Weickert, T. Brox, P. Mrázek, and M. Welk, *On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and SIDEs*, SIAM J. Numer. Anal., 42 (2004), pp. 686–713.

[60] G. Steidl, S. Didas, and J. Neumann, *Relations between higher order TV regularization and support vector regression*, in Scale-Space 2005, Lecture Notes in Comput. Sci. 3459, R. Kimmel, N. Sochen, and J. Weickert, eds., Springer, Berlin, 2005, pp. 515–527.

[61] K. Stuben, *An introduction to algebraic multigrid*, in Multigrid, U. Trottenberg, C. Oosterlee, and A. Schuller, Academic Press, London, 2001, pp. 413–532.

[62] S. Ta'asan, *Lecture Note* 4 *of Von-Karman Institute Lectures*, http://www.math.cmu.edu/~shlomo/VKI-Lectures/lecture4 (1997).

[63] X. C. Tai, *Rate of convergence for some constraint decomposition methods for nonlinear variational inequalities*, Numer. Math., 93 (2000), pp. 755–786.

[64] X.-C. Tai and M. Espedal, *Rate of convergence of some space decomposition methods for linear and nonlinear problems*, SIAM J. Numer. Anal., 35 (1998), pp. 1558–1570.

[65] X. C. Tai and P. Tseng, *Convergence rate analysis of an asynchronous space decomposition method for convex minimization*, Math. Comp., 71 (2001), pp. 1105–1135.

[66] X. C. Tai and J. C. Xu, *Global and uniform convergence of subspace correction methods for some convex optimization problems*, Math. Comp., 71 (2001), pp. 105–124.

[67] P. Tseng, *Convergence of a block coordinate descent method for nondifferentiable minimization*, J. Optim. Theory Appl., 109 (2001), pp. 475–494.

[68] C. R. Vogel, *A multigrid method for total variation-based image denoising*, in Computation and Control IV, Progr. Systems Control Theory 20, K. Bowers and J. Lund, eds., Birkhäuser, Boston, 1995.

[69] C. R. Vogel, *Negative results for multilevel precondtioners in image deblurring*, in Scale-Space Theories in Computer Vision, M. Nielson, P. Johansen, O. F. Olsen, and J. Weickert, eds., Springer, Berlin, 1999, pp. 292–304.

[70] C. R. Vogel, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.

[71] C. R. Vogel and M. E. Oman, *Iterative methods for total variation denoising*, SIAM J. Sci. Comput., 17 (1996), pp. 227–238.

[72] C. R. Vogel and M. E. Oman, *Fast, robust total variation-based reconstruction of noisy, blurred images*, IEEE Trans. Image Process., 7 (1998), pp. 813–824.

[73] A. M. Yip and F. Park, *Solution Dynamics, Causality, and Critical Behavior of the Regularization Parameter in Total Variation Denoising Problems*, CAM Report 03-59, University of California, Los Angeles, CA, 2003.