SIAM J. Imaging Sciences
Vol. 1, No. 1, pp. 51–74

Cascadic Multiresolution Methods for Image Deblurring[*]

Serena Morigi[†], Lothar Reichel[‡], Fiorella Sgallari[†], and Andriy Shyshkov[‡]

Abstract. This paper investigates the use of cascadic multiresolution methods for image deblurring. Iterations with a conjugate gradient-type method are carried out on each level, and terminated by a stopping rule based on the discrepancy principle. Prolongation is carried out by nonlinear edge-preserving operators, which are defined via PDEs associated with Perona–Malik or total variation-type models. Computed examples demonstrate the effectiveness of the methods proposed.

Key words. ill-posed problems, deblurring, multiresolution, regularizing iterative methods, edge-preserving operators

AMS subject classifications. 65R32, 65F10, 65N06

DOI. 10.1137/070694065

1. Introduction.

Image deblurring is an important task with many applications. The blurring of images may be caused by object motion, calibration errors of imaging devices, or random fluctuations of the medium, e.g., the atmosphere. We are interested in restoring images that have been contaminated by both blur and noise; in particular, we would like to be able to recover edges accurately.

Gray-scale two-dimensional images often are represented by real-valued functions defined on a rectangular region $\Omega \subset \mathbb{R}^2$ or by the discretization of such functions. Let the function f^δ represent the available observed blur- and noise-contaminated image and let the function $\hat{u}$ represent the associated (unavailable) blur- and noise-free image that we would like to recover. We assume that f^δ and $\hat{u}$ are related by the degradation model

$$f^\delta(x) = \int_\Omega h(x,y)\hat{u}(y)dy + \eta^\delta(x), \qquad x \in \Omega, \tag{1}$$

where η^δ represents additive noise in the data f^δ and h is the point spread function. In our applications, h is smooth, and, hence, the integral operator is compact. Moreover, h does not depend on x and y individually but only on $x - y$. Thus, with a slight abuse of notation, we have $h(x,y) = h(x-y)$. Then (1) can be expressed as

$$f^\delta = h * \hat{u} + \eta^\delta, \tag{2}$$

[*] Received by the editors June 8, 2007; accepted for publication (in revised form) December 3, 2007; published electronically March 20, 2008. This work has been supported by a MIUR-Cofin 2006 project grant, by the University of Bologna "Funds for selected research topics," and by an OBR Challenge grant.
http://www.siam.org/journals/siims/1-1/69406.html

[†] Department of Mathematics-CIRAM, University of Bologna, Via Sarogozza 8, 40123 Bologna, Italy (morigi@dm.unibo.it, sgallari@dm.unibo.it).

[‡] Department of Mathematical Sciences, Kent State University, Kent, OH 44242 (reichel@math.kent.edu, ashyshko@math.kent.edu).

where $*$ denotes two-dimensional convolution. In addition, we assume $h(s)$ to be symmetric; i.e., $h(-s) = h(s)$. Nonsymmetric kernels will be considered in future work.

Our task is to recover $\hat{u}$ given h and the observed image f^δ. The noise η is not known, but a fairly accurate bound for the norm of η^δ,

$$\left(\int_\Omega (\eta^\delta(x))^2 dx\right)^{1/2} \leq \delta, \tag{3}$$

is assumed to be available.

Straightforward solution of the linear model

$$f^\delta = h * u \tag{4}$$

for u does not provide a meaningful approximation of the desired noise- and blur-free image $\hat{u}$. The reason for this is that (4) ignores the noise η^δ in the right-hand side of (2) and that the inverse of the integral operator in (4) is unbounded if it exists. The latter follows from the compactness of the integral operator. The task of solving (4) therefore is an ill-posed problem; see, e.g., Engl, Hanke, and Neubauer [7] for discussions on the solution of this kind of problem.

A meaningful approximation of $\hat{u}$ can be determined by first replacing (4) with a nearby problem whose solution is less sensitive to perturbations in the data f^δ. Tikhonov regularization is possibly the best understood replacement approach. For image restoration problems, the following form of Tikhonov regularization has proved to be successful:

$$\min_u \left\{ \int_\Omega \frac{1}{2}(h * u - f^\delta)^2 + \alpha\, R(u) dx \right\}, \tag{5}$$

where $R(u)$ is a regularization operator and $\alpha > 0$ a regularization parameter. For instance, we may choose R to be of the form

$$R(u) = \psi(|\nabla u|^2), \tag{6}$$

where ψ is a monotonically increasing function and ∇u denotes the gradient of u; see, e.g., Welk et al. [23] for a discussion on this kind of regularization operator. For example, $\psi(t) = t^{1/2}$ yields the total variation (TV) operator

$$R(u) = |\nabla u|; \tag{7}$$

see, e.g., Rudin, Osher, and Fatemi [19].

The Euler–Lagrange equation associated with (5), supplied with a gradient descent, which gives a minimizer of (5) as $t \to \infty$, is of the form

$$\frac{\partial u}{\partial t} = -h * (h * u - f^\delta) + \alpha\, D(u), \qquad u^0 = f^\delta, \tag{8}$$

where u^0 denotes the initial function. The derivation of (8) uses the symmetry of h and the Dirichlet boundary condition $u = 0$; however, we note that other boundary conditions, such as Neumann conditions, may yield a more accurate restoration in some situations; see, e.g., [15, 23]. We also refer to D as a regularization operator. For instance, $\psi(t) = \frac{1}{2}t$ gives

$$D(u) = \Delta u, \tag{9}$$

where Δ denotes the Laplacian. For more general monotonically increasing differentiable functions ψ, we obtain

$$D(u) = \mathrm{div}(g(|\nabla u|^2)\nabla u) \tag{10}$$

with diffusivity $g(t) = d\psi(t)/dt$.

Instead of specifying ψ, one may choose the diffusivity g. A common choice is the Perona–Malik diffusivity

$$g(s) = 1/(1 + s/\rho), \tag{11}$$

where $\rho > 0$ is a small positive constant; see [16]. The TV operator (7) yields

$$D(u) = \mathrm{div}\left(\frac{\nabla u}{|\nabla u|}\right). \tag{12}$$

The choice of regularization operator in (8) is important for the success of the deblurring process. The operator (9) typically yields oversmoothed restored images, while the operators (10) and (12) generally provide restored images of higher quality and with sharper edges than (9).

This paper considers two nonlinear regularization operators. The first is a weighted TV operator,

$$D_1(u) = |\nabla u|_\epsilon^q \,\mathrm{div}\left(\frac{\nabla u}{|\nabla u|_\epsilon^q}\right), \tag{13}$$

where $|\nabla u|_\epsilon^q = \left(|\nabla u|^2 + \epsilon\right)^{q/2}$ with q a constant, such that $1 \le q \le 2$. This operator is a modification of (12) and, hence, is related to TV-norm regularization (7); see, e.g., [13].

The other operator we consider is given by (10) with g defined by (11). We refer to this operator as D_2, i.e.,

$$D_2(u) = \mathrm{div}(g(|\nabla u|^2)\nabla u), \qquad g(|\nabla u|^2) = 1/(1 + |\nabla u|^2/\rho). \tag{14}$$

We turn to the discretization of linear and nonlinear deblurring methods based on (4) and (8), respectively, and discuss the computational effort required by these methods. Discretization of (4) yields a linear system of equations

$$b^\delta = Au, \qquad A \in \mathbb{R}^{n\times n}, \qquad u, b^\delta \in \mathbb{R}^n, \tag{15}$$

where A represents the blurring operator and b^δ the available noise- and blur-contaminated image. The special form of (4) makes it possible to choose A as a real symmetric block Toeplitz matrix with Toeplitz blocks. The singular values of the matrix A "cluster" at the origin. Therefore, A is of ill-determined rank. In particular, A is severely ill-conditioned and may be singular. The matrix A may be indefinite.

Let $\hat{u}$ also denote a discrete approximation of the desired continuous solution, $\hat{u}$, of (2). Specifically, we let $\hat{u}$ be the minimal-norm solution of the linear system of equations

$$\hat{b} = A\hat{u}, \tag{16}$$

which is assumed to be consistent. The left-hand side, $\hat{b}$, represents a blurred, but noise-free, image. Let $\mathcal{R}(A)$ and $\mathcal{N}(A)$ denote the range and null space of A, respectively. Then $\hat{u} \perp \mathcal{N}(A)$ or, equivalently, $\hat{u} \in \mathcal{R}(A)$.

Iterative methods for the solution of (15) provide an attractive alternative to Tikhonov regularization for large-scale problems; see, e.g., [7, 9]. The iteration number can be thought of as a discrete regularization parameter. A regularized solution of (15) is obtained by terminating the iterations after suitably few steps. Truncated iteration methods of this kind often give reasonable results when applied to image deblurring; however, due to cut-off of high frequencies, these methods may introduce artifacts, such as "ringing," and fail to recover edges accurately.

The Euler–Lagrange equation (8) is discretized in space using finite differences; the computational grid for the spatial finite difference discretization corresponds to the pixel grid. Semidiscretization of (8) yields

$$\frac{du}{dt} = (\alpha\, L(u) - A^2)u + Ab^\delta, \tag{17}$$

where $L(u)u$ is a discretization of the regularizing operator $D(u)$ in the right-hand side of (8). Thus $L(u)$ is a discrete nonlinear diffusion operator.

Explicit time-stepping schemes give fully discretized versions of (17) with the discretized solution easily computable for every time-step τ. The Courant–Friedrich–Levy (CFL) condition determines an upper bound for the time-step τ that guarantees stability of the evolution. This bound is very restrictive and typically requires that a very large number of time-steps be carried out; see, e.g., [23] for a discussion. Explicit methods therefore are computationally expensive.

Semi-implicit time-discretizations of (17) allow larger time-steps τ. These discretizations linearize (17) by using the u-value from the previous time-step in the evaluation of $L(u)$. Let u^i denote the computed solution at time τi. Semi-implicit integration methods require the solution of a linear system of equations at every time-step; i.e., u^i is determined by solving

$$[I - \tau(\alpha\, L(u^{i-1}) - A^2)]u^i = u^{i-1} + \tau A b^\delta \tag{18}$$

(see, e.g., [8] and [21] for further details on these kinds of schemes). Although the matrix $L(u^{i-1})$ is sparse and A is a block Toeplitz matrix with Toeplitz blocks, the matrix in the left-hand side of (18) cannot be easily factored or inverted. Moreover, the term A^2 in (18) demands a restriction in the size of the time-step in order to guarantee stability. This restriction combined with the difficulty of solving (18) for u^i makes solution methods based on semi-implicit integration expensive. Thus, while nonlinear models based on (17) provide denoising of high quality and deblurring of acceptable quality, their integration by explicit or semi-implicit methods is computationally expensive. This is illustrated with a computed example in section 4.

The present paper proposes new cascadic multiresolution methods that share the computational efficiency of truncated iteration for the solution of linear systems of (15) with the edge-preserving property of nonlinear models (8). The multiresolution methods typically require fewer matrix-vector product evaluations than standard 1-level truncated iterative methods for the solution of (15) and often determine restored images of as high quality as the much more expensive methods for the solution of nonlinear models of the form (8). Our multiresolution methods are based on regularization by truncated iteration on each level and use a stopping criterion for the iterations on each level based on the discrepancy principle. Nonlinear edge-preserving prolongation operators enable the accurate restoration of edges. These operators are inspired by the nonlinear regularization operators (13) and (14) used in (8).

This paper is organized as follows. Section 2 introduces the multiresolution framework and our stopping criterion for the iterations. Section 3 discusses the edge-preserving prolongation operators used. Numerical examples in section 4 illustrate the performance of the methods, and section 5 contains concluding remarks.

Other multilevel methods for ill-posed problems recently have been described in [6, 18]. The numerical examples presented in these references show the promise of multilevel methods. In [18] cascadic multilevel methods that use linear prolongation operators are discussed. Computed examples in section 4 show the method of the present paper to yield restorations of higher quality. Further references to multilevel methods for ill-posed problems can be found in [18].

2. Multiresolution and the discrepancy principle.

We first discuss the application of a 1-level conjugate gradient-type iterative method, commonly referred to as MR-II, to the solution of (15), and describe the discrepancy principle for the termination of the iterations. This is followed by a presentation of multiresolution methods and some of their properties.

Introduce for $v = [v^{(1)}, v^{(2)}, \ldots, v^{(n)}]^T \in \mathbb{R}^n$ the weighted least-squares norm

$$\|v\| = \left(\frac{1}{n} \sum_{i=1}^{n} \left(v^{(i)} \right)^2 \right)^{1/2}. \tag{19}$$

Assume that a bound for the error $e = b^\delta - \hat{b}$ in the right-hand side b^δ of (15) is available; i.e., analogously to (3) we have

$$\|e\| \le \delta. \tag{20}$$

This bound allows us to use the discrepancy principle to determine a suitable number of iterations with the MR-II method applied to the (approximate) solution of (15). Let the initial approximate solution be $u_0 = 0$. Then, typically, the first few iterates furnish improved approximations of the desired solution $\hat{u}$ of the error-free system (16). However, after an optimal number of iterations, which generally is not explicitly known, subsequently computed iterates commonly suffer from severe error-contamination due to the error e in b^δ. It is therefore important to terminate the iterations sufficiently early. The discrepancy principle is an aid for determining how many iterations to carry out.

Definition (discrepancy principle). *Let $\gamma > 1$ be a fixed constant and let $e = b^\delta - \hat{b}$ satisfy* (20). *The vector u is said to satisfy the discrepancy principle if $\|b^\delta - Au\| \leq \gamma\delta$.*

We will apply the discrepancy principle for fixed γ and different values of δ. MR-II is a conjugate gradient-type method, which is well suited for the solution of linear systems of equations with a symmetric, possibly indefinite, matrix. Each iteration requires the evaluation of one matrix-vector product with the matrix A. The MR-II method has been studied by Hanke [9, Chapter 6]; an alternate implementation is described in [3].

With initial iterate $u_0 = 0$, the kth iterate determined by MR-II, u_k, belongs to the Krylov subspace

$$\mathcal{K}_k(A, Ab^\delta) = \text{span}\{Ab^\delta, A^2b^\delta, \ldots, A^kb^\delta\}.$$

MR-II is a minimal residual method; i.e., u_k satisfies

$$\|b^\delta - Au_k\| = \min_{u \in \mathcal{K}_k(A, Ab^\delta)} \|b^\delta - Au\|,$$

and it follows that

$$\|b^\delta - Au_{k+1}\| \leq \|b^\delta - Au_k\| \qquad \forall k. \tag{21}$$

Thus, all iterates generated by the method live in $\mathcal{R}(A)$. Therefore, the MR-II method is referred to as a range restricted minimal residual (RRMR) method in [3]. Since $\mathcal{R}(A) = \mathcal{N}(A)^\perp$, the iterates u_k are orthogonal to $\mathcal{N}(A)$. The property (21) makes it natural to terminate the iterations using the following stopping rule based on the discrepancy principle.

Stopping Rule 2.1. *Let δ and γ be the same as in the discrepancy principle. Terminate the iterations when for the first time*

$$\|b^\delta - Au_k\| \leq \gamma\delta. \tag{22}$$

Denote the resulting stopping index by $k(\delta)$.

Note that typically $k(\delta)$ increases monotonically as δ decreases to zero with γ kept fixed. This depends on the fact that the right-hand side bound in (22) gets smaller when δ decreases. Bounds for the growth of $k(\delta)$ as δ decreases are provided by [9, Corollary 6.18].

An iterative method equipped with this stopping rule is said to be a *regularization method* if the computed iterates $u_{k(\delta)}$ satisfy

$$\lim_{\delta \searrow 0} \sup_{\|e\| \leq \delta} \|u_{k(\delta)} - \hat{u}\| = 0, \tag{23}$$

where $\hat{u}$ is the minimal-norm solution of (16). Hanke [9, Theorem 6.15] shows that the iterates determined by MR-II satisfy (23) in a Hilbert space setting. Regularization is achieved because the computed approximate solution $u_{k(\delta)}$ lives in the Krylov subspace $\mathcal{K}_{k(\delta)}(A, Ab^\delta)$ of typically fairly small dimension $k(\delta)$. The parameter $\gamma > 1$ in (22) is chosen to be close to unity if an accurate estimate of the norm of the noise $\|e\|$ is available; otherwise a larger value of γ is used.

There are also quite a few heuristic stopping rules that do not use information about the norm of the noise in the right-hand side b^δ; see, e.g., [12, 14, 17] for recent discussions and references. Many of these rules work well for a large number of problems, but they may fail for others. It would appear possible to use some of these rules in the context of multiresolution methods. This requires further investigation.

We use the MR-II method because it is a regularization method and the iterates satisfy a short recurrence relation. The conjugate gradient method applied to the normal equations associated with (15) also satisfies these properties, but generally requires more computational work, since each iteration demands the evaluation of two matrix-vector products, one with A and one with A^T.

We turn to the description of a multiresolution method based on MR-II and a termination criterion analogous to Stopping Rule 2.1. Let $W_1 \subset W_2 \subset \cdots \subset W_\ell$ be a sequence of nested linear subspaces of $\mathbb{R}^n$ of increasing dimensions with $W_\ell = \mathbb{R}^n$. We refer to the subspaces W_j as levels, with W_1 being the coarsest level and W_ℓ the finest. The dimension of W_j is denoted by n_j. In the computed examples, we let

$$n_{j+1} = 4n_j \qquad \forall j. \tag{24}$$

Each level is equipped with a weighted least-squares norm analogous to (19). Specifically, W_j has a norm of the form (19) with n replaced by n_j.

Define, for $1 \le i < \ell$, the *restriction operators* $R_i : \mathbb{R}^n \to W_i$ as follows. Let $M_{i+1} : W_{i+1} \to W_i$ denote the averaging operator in W_{i+1} that replaces groups of four adjacent pixels in the image represented by b^δ_{i+1} by one pixel, whose value is the average of the values of the four pixels it replaces. Introduce

$$b^\delta_i = M_{i+1} b^\delta_{i+1}, \qquad R_i = M_{i+1} M_{i+2} \dots M_\ell, \qquad 1 \le i < \ell. \tag{25}$$

For notational convenience, we let $b^\delta_\ell = b^\delta$ and $R_\ell = I$. Then

$$b^\delta_i = R_i b^\delta, \qquad 1 \le i \le \ell.$$

Define

$$\hat{b}_i = R_i \hat{b}, \qquad A_i = R_i A R_i^*, \qquad 1 \le i \le \ell, \tag{26}$$

where R_i^* is the adjoint of R_i. Thus, A_i is the restriction of A to W_i. Since averaging is a smoothing operator, we expect the vector b^δ_{i-1} to contain less high-frequency noise than b^δ_i.

Example 2.1. Assume that the entries of the noise e in b^δ are uncorrelated random variables with zero mean. Then the entries of

$$e_i = [e_i^{(1)}, e_i^{(2)}, \dots, e_i^{(n_i)}]^T = b^\delta_i - \hat{b}_i$$

are uncorrelated and have zero mean for all i. Let $\|e_i\|_V^2$ denote the average variance of the components $e_i^{(j)}$ of e_i; i.e.,

$$\|e_i\|_V^2 = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathrm{Var}(e_i^{(j)}).$$

Then

$$\|e_i\|_V^2 = \frac{1}{n_i}\sum_{j=1}^{n_i} \mathrm{Var}(e_i^{(j)}) = \frac{1}{16n_i}\sum_{j=1}^{n_{i+1}} \mathrm{Var}(e_{i+1}^{(j)}) = \frac{1}{4}\|e_{i+1}\|_V^2,$$

where we have used (24) and the fact that the components $e_i^{(j)}$ are the average of four entries of e_{i+1}. Thus, $\|e_i\|_V = \frac{1}{2}\|e_{i+1}\|_V$. Approximating $\|e_i\|_V$ by $\|e_i\|$ for all i yields

$$\|e_i\| \approx \frac{1}{2}\|e_{i+1}\|. \tag{27}$$

We also require *prolongation operators* from level $i-1$ to level i for all i. Both linear and nonlinear prolongation operators will be used. In the computed examples, the linear prolongation operators $P_i : W_{i-1} \to W_i$, $1 < i \leq \ell$, are defined by piecewise linear interpolation, while the nonlinear prolongations $S_iP_i : W_{i-1} \to W_i$, $1 < i \leq \ell$, are designed to be edge-preserving. Here the $S_i : W_i \to W_i$ are nonlinear edge-preserving smoothing operators. Different choices of smoothing operators and further details are discussed in section 3.

The multiresolution methods of the present paper are cascadic and are based on the MR-II iterative method. Thus, the multiresolution methods first determine an approximate solution of $A_1u = b_1^\delta$ in W_1 using MR-II. The iterations with MR-II are terminated as soon as an iterate that satisfies a stopping rule related to the discrepancy principle has been determined. This iterate is mapped from W_1 into W_2 by the nonlinear edge-preserving prolongation S_2P_2. A correction of this mapped iterate in W_2 is computed by MR-II. Again, the MR-II-iterations are terminated by a stopping rule related to the discrepancy principle. The approximate solution in W_2 determined in this fashion is mapped into W_3 by the edge-preserving nonlinear prolongation S_3P_3. The computations are continued in this manner until an approximation of $\hat{u}$ has been determined in $W_\ell = \mathbb{R}^n$. We refer to these methods as multiresolution MR-II (MR-MR-II) methods. They are described by the following algorithm. The multiresolution methods used in this paper differ in the choice of smoothing operators S_i and in the number of levels used. One may, of course, also choose prolongation operators P_i different from piecewise linear interpolation; however, this is not explored in the present paper.

Algorithm 2.2 (MR-MR-II).

```
Input: $A$, $b^\delta$, $\delta$, $\ell \geq 1$ (number of levels);
Output: approximate solution $u_\ell \in W_\ell$ of (15);
Determine $A_i$ and $b_i^\delta$ from (26) for $1 \leq i \leq \ell$;
$u_0 := 0$;
for $i := 1, 2, \ldots, \ell$ do
        $u_{i,0} := S_iP_iu_{i-1}$;
        $\vartriangle u_{i,m_i} := MR\text{-II}(A_i, b_i^\delta - A_iu_{i,0})$;
        Correction step:  $u_i := u_{i,0} + \vartriangle u_{i,m_i}$;
endfor
$u_\ell := S_\ell u_\ell$;
```

In the algorithm above $MR\text{-II}(A_i, b_i^\delta - A_iu_{i,0})$ denotes the computation of the approximate solution $\vartriangle u_{i,m_i}$ of

$$A_iz_i = b_i^\delta - A_iu_{i,0} \tag{28}$$

by m_i MR-II-iterations, using initial iterate ${}_\Delta u_{i,0} = 0$. For $i = 1$, the algorithm sets $u_{1,0} = 0$. We remark that the matrix A_i typically has the same structure as A_{i+1} when the unknowns on each level are enumerated in the same fashion. For instance, if $A_\ell = A$ is a symmetric block Toeplitz matrix with Toeplitz blocks, then so are the A_i, $1 \le i < \ell$. Also, if A is banded, then so are the matrices A_i, $1 \le i < \ell$. In our applications, the A_i are symmetric Toeplitz matrices with Toeplitz blocks and may be indefinite.

Our stopping rule on each level is based on the assumption that there are constants c_i independent of δ, such that

$$\|b_i^\delta - \hat{b}_i\| \le c_i\delta, \qquad 1 \le i \le \ell, \tag{29}$$

where δ satisfies (20). Relation (27) suggests the choice

$$c_i = \frac{1}{2}c_{i+1}, \qquad 1 \le i \le \ell, \tag{30}$$

with $c_\ell = \gamma$, where γ is the same as in (22). We use these values of c_i in the computed examples of section 4.

Stopping Rule 2.3. *Let δ and the c_i be the same as in (20) and (29), and denote the iterates determined by MR-II applied to the solution of (28) by ${}_\Delta u_{i,m}$, $m = 1, 2, \ldots$, with initial iterate ${}_\Delta u_{i,0} = 0$ and $u_{1,0} = 0$. Terminate the iterations on level i as soon as an iterate ${}_\Delta u_{i,m_i}$ that satisfies*

$$\|b_i^\delta - A_i u_{i,0} - A_i {}_\Delta u_{i,m_i}\| \le c_i\delta$$

has been determined, where $m_i = m_i(\delta)$ denotes the termination index.

Cascadic MR-II-based multiresolution methods related to those of Algorithm 2.2 are discussed in [18]; the methods in [18] differ from the MR-MR-II methods described by Algorithm 2.2 in that they use piecewise linear prolongation operators only and the restriction operators are defined by interpolation instead of by local averaging. Theorem 3.3 and Corollary 3.2 in [18] show that, under suitable conditions, the MR-II-based multiresolution methods considered in [18] are regularization methods in the sense that the computed solution on each level converges to the minimal-norm solution of the noise-free problem for the corresponding level. This is analogous to the property (23) of 1-level MR-II.

The proofs of Theorem 3.3 and Corollary 3.2 in [18] carry over to the MR-MR-II methods of the present paper when $S_i = I$ for all i. These results are shown for solutions and operators defined in infinite-dimensional Hilbert spaces and require the δ_i to be chosen in a particular order. Since images are represented by pixels, image restoration problems live in finite-dimensional spaces. We will show that the MR-MR-II methods defined by Algorithm 2.2 with $S_i = I$ for all i are regularization methods for finite-dimensional problems. Our proof allows the reduction of all δ_i simultaneously. Specifically, we let $\delta_i = \delta$ for all i and then let $\delta \searrow 0$.

Theorem 2.4. *Assume that the equation*

$$A_i u = \hat{b}_i \tag{31}$$

is consistent and has minimal-norm solution $\hat{u}_i$ for each i. In particular, $\hat{u}_\ell = \hat{u}$. Let $S_i = I$ for all i and let the linear projections P_i satisfy

$$\mathcal{R}(P_i) \subset \mathcal{R}(A_i), \qquad 1 < i \le \ell. \tag{32}$$

Algorithm 2.2 *computes the approximate solution of ℓ linear systems of equations with noise-contaminated right-hand sides. Assume that the associated noise-free right-hand sides $\hat{b}_i$ are generic; i.e., $\hat{b}_i$ is assumed not to be in an invariant subspace of A_i of low dimension. Let the errors in the projected right-hand sides be of the form*

$$b_i^\delta - \hat{b}_i = c_i\delta\, e_i, \qquad 1 \leq i \leq \ell, \tag{33}$$

where $e_i \in W_i$ is a unit vector independent of δ and the $c_i > 0$ are the constants in (29)*. Terminate the iterations with MR-*II *in Algorithm* 2.2 *on levels $1, 2, \ldots, \ell$ according to Stopping Rule* 2.3*. This yields the iterates u_i for levels $1 \leq i \leq \ell$. Then the MR-MR-*II *method described by Algorithm* 2.2 *is a regularization method on each level, i.e.,*

$$\lim_{\delta \searrow 0} \|u_i - \hat{u}_i\| = 0, \qquad 1 \leq i \leq \ell. \tag{34}$$

Proof. It suffices to show the theorem for $\ell = 2$ levels. A proof for $\ell > 2$ levels can be established similarly. Let A_i have $1 \leq k_i \leq n_i$ distinct nonvanishing eigenvalues. By assumption, the right-hand side $\hat{b}_1$ of

$$A_1 u = \hat{b}_1 \tag{35}$$

is generic, and, therefore, the solution of (35) by MR-II requires k_1 iterations. The iterates determined by MR-II live in $\mathcal{R}(A_1)$, and it follows that the solution $\hat{u}_1$ computed by MR-II is of minimal norm. Thus,

$$\hat{u}_1 = A_1^\dagger \hat{b}_1, \tag{36}$$

where $A_1^\dagger$ denotes the Moore–Penrose pseudoinverse of A_1. When A_1 is nonsingular, $A_1^\dagger = A_1^{-1}$.

We are interested in the behavior of the computed solution as $\delta \searrow 0$. We therefore may choose the initial $\delta > 0$ small enough so that Stopping Rule 2.3 yields the maximal number of iterations on both levels. We then consider the limit of the computed solution as δ is reduced further.

For $\delta > 0$ sufficiently small, the perturbed right-hand side b_1^δ also is generic, and satisfaction of Stopping Rule 2.3 requires that k_1 MR-II-iterations be carried out. The computed solution can be expressed as

$$u_1 = A_1^\dagger b_1^\delta. \tag{37}$$

The entries of u_1 are continuous functions of the entries of b_1^δ, which, in view of (33), are continuous functions of δ, provided that $\delta > 0$ is sufficiently small. This shows that $u_1 \rightarrow \hat{u}_1$ as $\delta \searrow 0$.

We turn to level $\ell = 2$ and apply MR-II to the solution of

$$A_2 z_2 = b_2^\delta - A_2 P_2 u_1. \tag{38}$$

By assumption, the noise-free vector $\hat{b}_2 - A_2 P_2 \hat{u}_1$ associated with the right-hand side in (38) is generic, and, therefore, so is the right-hand side in (38) for all $\delta > 0$ sufficiently small. This implies that, for all $\delta > 0$ sufficiently small, MR-II applied to the solution of (38) requires that k_2 iterations be carried out to satisfy Stopping Rule 2.3. The computed solution $\triangle u_{2,k_2}$ can be expressed as

$$\triangle u_{2,k_2} = A_2^\dagger(b_2^\delta - A_2P_2u_1),$$

and, hence,

$$u_2 = P_2u_1 + \triangle u_{2,k_2} = (I - A_2^\dagger A_2)P_2u_1 + A_2^\dagger b_2^\delta = A_2^\dagger b_2^\delta,$$

where the last equality follows from the fact that $(I - A_2^\dagger A_2)P_2 = 0$, which is a consequence of (32). In view of (31), we have $\hat{u}_2 = A_2^\dagger \hat{b}_2$ and, therefore,

$$u_2 - \hat{u}_2 = A_2^\dagger(b_2^\delta - \hat{b}_2),$$

which establishes (34) for $i = 2$. ■

It is essential in the above proof that MR-II terminate after finitely many steps with a minimal-norm least-squares solution. The proof carries over to other iterative methods with this property, such as the conjugate gradient method applied to the normal equations. However, as already mentioned above the latter method typically requires more matrix-vector product evaluations than MR-II. We therefore use MR-II in the present paper. Note that the standard conjugate gradient method cannot be applied, because the method is not guaranteed to yield the minimal-norm least-squares solution when the matrix is singular.

The property (34) for the multiresolution method is analogous to (23) for MR-II and justifies its use. However, this property does not guarantee that the restored images are of high quality. To achieve the latter, we use nonlinear smoothing operators S_i and carry out local smoothing described in subsection 3.3.

We remark that the multiresolution methods presented in this section differ significantly from cascadic and other multilevel methods designed for the solution of well-posed boundary value problems for PDEs; see e.g., [1, 20] and references therein for discussions of the latter kinds of methods. One reason for this is that in the problems considered in the present paper, the solution of discretized Fredholm integral equations of the first kind, highly oscillatory eigenvectors typically are associated with eigenvalues close to the origin and have to be damped; see, e.g., [11, Chapter 1] for a discussion. Multilevel methods developed for the solution of well-posed boundary value problems for PDEs damp eigenvectors associated with large eigenvalues; see, e.g., [1, 20]. Therefore iterative methods used for the latter problems, such as variants of the Gauss–Seidel method, cannot be used for the solution of the problems considered in the present paper.

The use of cascadic multilevel methods is natural for the problems considered in this paper, because the following hold:

- The smaller problems, i.e., linear systems of equations $A_iu_i = b_i^\delta$ with i small, typically are not very ill-conditioned. It is therefore often possible to compute fairly accurate approximations of the restriction of the desired vector $\hat{u}$ to W_i even in the presence of error in b_i^δ. The computed solution in W_i, after prolongation, generally furnishes a fairly accurate approximation of the desired solution in W_{i+1}.
- It is easy to apply stopping rules based on the discrepancy principle.

3. Edge-preserving prolongation operators.

The cascadic multilevel method proposed in [18] applies prolongation by piecewise linear interpolation. While these piecewise linear prolongation operators work well in some contexts, they are not well suited for image restoration, because the restored images obtained with these prolongation operators often suffer from attenuation of high frequencies and checkerboard effects; see section 4 for illustrations. This section describes nonlinear edge-preserving prolongation operators, which yield more accurate restorations than the piecewise linear prolongation operators used in [18] for very little additional computational work. The nonlinear prolongation operators $S_iP_i : W_{i-1} \rightarrow W_i$ combine the piecewise linear prolongation operator $P_i : W_{i-1} \rightarrow W_i$, defined by piecewise linear interpolation, with the nonlinear edge-preserving smoothing operator $S_i : W_i \rightarrow W_i$. The smoothing operator reduces the shortcomings of piecewise linear prolongation.

The smoothing operators S_i are determined by integration of PDEs of the form

$$\frac{\partial u}{\partial t} = D(u) \tag{39}$$

for a short time-interval, where $D(u)$ is the regularization operator in (8) defined by either (13) or (14).

Let u_{i-1} denote the computed solution determined in the *Correction step* of Algorithm 2.2 on level $i-1$. The piecewise linear prolongation operator P_i yields $P_iu_{i-1} \in W_i$. Application of the smoothing operator S_i to P_iu_{i-1} gives the initial approximation $u_{i,0}$ of the desired solution on level i; see Algorithm 2.2. The operator S_i is defined implicitly by integration of the nonlinear system of differential equations in W_i,

$$\frac{du}{dt} = L(u)u, \qquad u \in W_i, \qquad u^0 = P_iu_{i-1}, \tag{40}$$

with Dirichlet boundary condition $u = 0$ and initial function u^0. Here $L(u)u$ is the same discretization of $D(u)$ as in (17).

The nonlinear system of differential equations (40) is designed to determine an element in W_i that has edges close to those of u_{i-1} in W_{i-1}. Integration is performed by carrying out fewer than 10 time-steps with an explicit method. The small number of time-steps avoids difficulties due to numerical instability. Moreover, the computational effort required for integration is small. The following subsections discuss the discretization of the operator $D(u)$ in (39).

3.1. Discretization of $D_1(u)$.

We describe the discretization of the operator $D = D_1$, defined by (13) and used in (39), by finite differences and proceed similarly as in [4]. The discretization defines the matrix $L(u)$ in (40). Each row of $L(u)$ is associated with a pixel and has, generically, five nonvanishing entries. We enumerate the pixels with index pairs (i,j). Consider a generic row associated with pixel (i,j). Its five nonvanishing entries are determined by the values of u at pixel (i,j) and at the four adjacent pixels in the horizontal and vertical directions. We refer to the latter pixels as E(ast) $(i+1,j)$, W(est) $(i-1,j)$, N(orth) $(i,j+1)$, and S(outh) $(i,j-1)$; see Figure 1 for the enumeration of the pixels and associated u-values. The row of $L(u)$ corresponding to pixel (i,j) is given by

$$\{l_{ij,S}, \ldots, l_{ij,E}, -(l_{ij,S} + l_{ij,E} + l_{ij,W} + l_{ij,N}), l_{ij,W}, \ldots, l_{ij,N}\}, \tag{41}$$

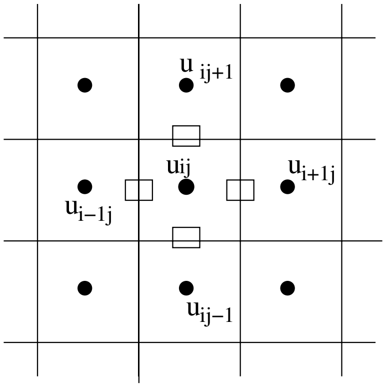

Figure 1. *Stencil for the finite difference discretization of* D_1.

where the coefficients associated with the pixels W(est) and E(ast) of (i,j) are of the form

$$l_{ij,W} = \frac{2s_{ij,E}}{s_{ij,W}+s_{ij,E}}, \qquad l_{ij,E} = \frac{2s_{ij,W}}{s_{ij,W}+s_{ij,E}}, \tag{42}$$

with

$$s_{ij,W} = (\delta^2_{i-1/2,j}(u)+\epsilon)^{q/2}, \qquad s_{ij,E} = (\delta^2_{i+1/2,j}(u)+\epsilon)^{q/2}. \tag{43}$$

Here $\delta^2_{i-1/2,j}(u)$ denotes the finite difference approximation of $|\nabla u|^2$ evaluated at the midpoint between the pixels $(i-1,j)$ and (i,j). Similarly, $\delta^2_{i+1/2,j}(u)$ denotes the finite difference approximation of $|\nabla u|^2$ evaluated at the midpoint between the pixels $(i+1,j)$ and (i,j). These midpoints are marked by rectangles in Figure 1. For example,

$$\begin{aligned}\delta^2_{i-1/2,j}(u) &= (u_{ij}-u_{i-1,j})^2 + \left(\frac{1}{2}\left(\frac{u_{i-1,j+1}+u_{i,j+1}}{2} - \frac{u_{i-1,j-1}+u_{i,j-1}}{2}\right)\right)^2 \\ &= (u_{ij}-u_{i-1,j})^2 + \frac{1}{16}(u_{i-1,j+1}+u_{i,j+1}-u_{i-1,j-1}-u_{i,j-1})^2.\end{aligned}$$

In the computed examples reported in section 4, we let $q = 1.5$ and choose ϵ to be a small constant in (43).

The coefficients $l_{ij,N}$ and $l_{ij,S}$ in (41) are evaluated similarly, where we note that the operator (13) is symmetric with regard to the partial derivatives $\partial/\partial x$ and $\partial/\partial y$. At boundary pixels, u is set to zero. The discretization outlined is a nine-point discretization scheme since, generically, u-values at pixel (i,j) and at all the eight surrounding pixels are used. The pixel (i,j) and its eight neighbors are marked by solid dots in Figure 1.

3.2. Discretization of $D_2(u)$. This subsection discusses finite difference discretization of the operator $D = D_2$, defined by (14) and used in (39). The matrix $L(u)$ in (40) obtained by the discretization has, generically, five nonvanishing entries in each row. In the row associated with pixel (i,j), these entries are determined by the values of u at pixel (i,j) and at the four adjacent pixels in the horizontal and vertical directions, denoted by N, S, E, W. A typical row of $L(u)$ associated with pixel (i, j) is of the form (41) with elements

$$l_{ij,E}(u) = \frac{g_{ij} + g_{i+1,j}}{2}, \qquad l_{ij,W}(u) = \frac{g_{ij} + g_{i-1,j}}{2},$$

where g_{ij} represents the discretization of $g(|\nabla u|^2)$ in (14). The partial derivatives are approximated by central finite differences, giving

$$g_{ij} = g\left(\left(\frac{u_{i+1,j} - u_{i-1,j}}{2}\right)^2 + \left(\frac{u_{i,j+1} - u_{i,j-1}}{2}\right)^2\right).$$

Expressions for $l_{ij,S}(u)$ and $l_{ij,N}(u)$ can be derived similarly; see [22] for details.

3.3. Local smoothing. Nonlinear models (8) based on the Perona–Malik or weighted TV-norm regularization operators (13) or (14), respectively, may yield restored images which suffer from "staircasing"; i.e., the restored images display large flat regions separated by artificial boundaries instead of a desired smooth surface. Mild forms of staircasing also can be discerned in the images determined by the nonlinear prolongation operators S_iP_i. Following Buades, Coll, and Morel [2], we reduce this problem by local least-squares smoothing. At each pixel x, we solve the local least-squares approximation problem

$$(44) \qquad \min_{a_j \in \mathbb{R}} \sum_{y \in \Omega_8(x)} \left((S_iP_iu_{i-1})(y) - p(y)\right)^2 \omega(x, y), \qquad p(y) = a_0 + a_1y_1 + a_2y_2,$$

where $\Omega_8(x)$ denotes the punctured eight-pixel neighborhood of the pixel x made up of the pixels closest to but different from x,

$$(45) \qquad \omega(x, y) = e^{-(u_{i,0}(y) - u_{i,0}(x))^2}$$

is a weight function, and $(S_iP_iu_{i-1})(y)$ denotes the entry of $S_iP_iu_{i-1}$ corresponding to pixel y. Moreover, y has the coordinates y_1 and y_2.

Let p denote the solution of (44). Note that p depends on x. The computation of p and evaluation of $p(x)$ for each pixel x is quite inexpensive. Let $u_{i,0}(x)$ denote the entry of the vector $u_{i,0}$ corresponding to pixel x. In Algorithm 2.2, we let $u_{i,0}(x) = p(x)$ for all pixels x (instead of $u_{i,0} = S_iP_iu_{i-1}$). This amounts to local smoothing of the entries of $S_iP_iu_{i-1}$. Due to the weight function (45), this smoothing does not blur edges significantly. For notational simplicity, we include the local smoothing of this subsection in the definition of the operator S_i. The local smoothing is applied on all levels.

We refer to application of the nonlinear prolongation operator S_iP_i as D_1-prolongation and of the smoother S_i as D_1-smoothing when $L(u)$ in (40) is defined by D_1. Analogously, when $L(u)$ is defined by D_2, we refer to application of S_iP_i as D_2-prolongation and of S_i as D_2-smoothing.

4. Numerical examples. We present several numerical examples, which illustrate properties of the multiresolution methods of this paper. Comparisons with the multilevel methods described in [18] and with the nonlinear model (8) also are reported.

The gray-scale images to be restored are represented by $m \times m$ pixels with each pixel stored in 8 bits. This allows pixel values in the interval $[0, 255]$. The pixels are ordered row-wise and stored in a vector of dimension $n = m^2$. Let $\hat{u} \in \mathbb{R}^n$ represent a blur- and noise-free image. We generate an associated blurred and noise-free image $\hat{b}$ by multiplying $\hat{u}$ by a block Toeplitz matrix $A \in \mathbb{R}^{n \times n}$ with Toeplitz blocks. The matrix A represents a Gaussian blurring operator and is generated with the MATLAB function `blur.m` from Regularization Tools [10]. This function has two parameters, `band` and `sigma`. The former specifies the half-bandwidth of the Toeplitz blocks and the latter the variance of the Gaussian point spread function. The larger `sigma` is, the more blurring. Enlarging `band` increases the storage requirement, the arithmetic work required for the evaluation of matrix-vector products with A, and to some extent the blurring.

A blur- and noise-contaminated image $b^\delta \in \mathbb{R}^n$ is obtained by adding an error vector $e \in \mathbb{R}^n$ to $\hat{b}$. Thus,

$$b^\delta = A\hat{u} + e.$$

The corrupted image $b^\delta \in \mathbb{R}^n$ is assumed to be available, and we would like to determine the blur- and noise-free image $\hat{u}$. In our experiments, e has normally distributed entries with mean zero, scaled to yield a desired noise-level

$$\nu = \frac{\|e\|}{\|\hat{u}\|},$$

from which we determine the value of δ in (20).

We denote the number of levels used in Algorithm 2.2 by ℓ. The vectors b_i^δ, $1 \leq i < \ell$, are determined by local averaging of the entries; cf. (25). The matrices A_i are defined by (26). The entries of the A_i can easily be computed for decreasing values of i. Note that the matrices A_i do not have to be stored; it suffices to define functions for the evaluation of matrix-vector products with the A_i. These products can be computed efficiently by using the structure of the A_i.

In all examples with multiresolution methods, we integrate the system of nonlinear differential equations (40) by Euler's method. Numerical experiments suggest that 5 time-steps of size 0.2 is appropriate for images contaminated by a moderate amount of noise (noise-level $1 \cdot 10^{-2}$) and 10 time-steps is suitable for images that have been contaminated by a fairly large amount of noise (noise-level $1 \cdot 10^{-1}$). The computed results are not sensitive to the number of time-steps in this range.

We assume that δ is a fairly accurate estimate of the norm of the noise, $\|e\|$, in the right-hand side b^δ and therefore let $\gamma = 1.01$ in Stopping Rule 2.1. Moreover, $c_\ell = \gamma$ in Stopping Rule 2.3. The other coefficients c_i in the latter stopping rule are determined by (30).

The displayed restored images provide a qualitative comparison of the performance of the proposed multiresolution methods for different prolongation operators. A quantitative comparison is given by the peak signal-to-noise ratio (PSNR):

$$\text{PSNR}(u_\ell, \hat{u}) = 20 \log_{10} \frac{255}{\|u_\ell - \hat{u}\|} \text{ dB}, \tag{46}$$

where $\hat{u}$ is the blur- and noise-free image and u_ℓ the restored image determined by Algorithm 2.2. Note that the norm $\|u_\ell - \hat{u}\|$ is the root mean squared error (RMSE) of $u_\ell - \hat{u}$; cf. (19).

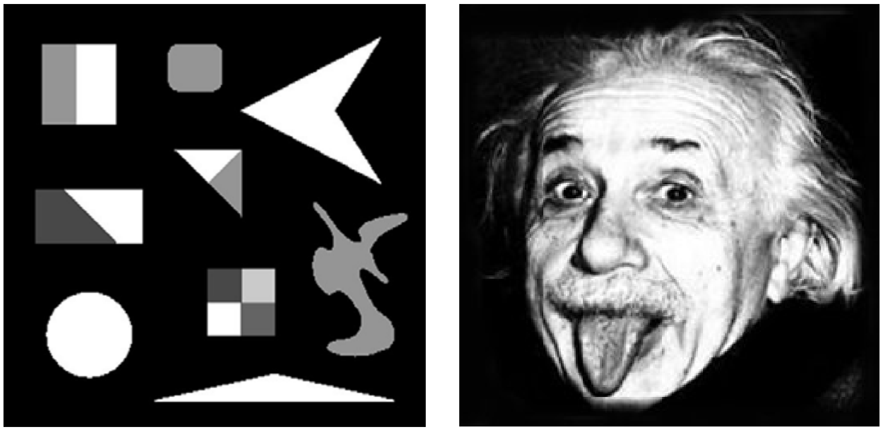

Figure 2. *Blur- and noise-free images for Examples 4.1 and 4.2.*

The numerator 255 is the largest pixel-value represented. We remark that PSNR-values do not always agree with visual judgment. All computations are carried out in MATLAB with machine epsilon about $\epsilon \approx 2 \cdot 10^{-16}$.

Example 4.1. The noise- and blur-free image used in this example is shown in Figure 2 (left). It is represented by 512×512 pixels. The black background makes it natural to use Dirichlet boundary conditions with boundary value zero. Figure 3 provides a qualitative comparison of images restored by Algorithm 2.2 with $\ell = 5$ levels and different prolongation operators; in particular, the figure displays the performance of piecewise linear prolongation as well as of nonlinear D_1- and D_2-based prolongations. The D_2-based prolongation operators are seen to provide the most accurate restoration, while piecewise linear prolongation gives the worst restoration; the restored image determined with piecewise linear prolongation shows artifacts and poorly defined boundaries.

The last row of Table 1 reports PSNR-values for the restorations shown in Figure 3. The image determined with D_2-based prolongation achieves the highest PSNR-value in agreement with the qualitative comparison furnished by Figure 3. Table 1 shows results for several noise-levels ν and number of levels ℓ in Algorithm 2.2. The columns marked “lin prlg” show results achieved with piecewise linear prolongation. Here and elsewhere in this section, $\ell = 1$ denotes the basic 1-level MR-II method applied on the finest level without smoothing. The results are reported in the columns marked “lin prlg” even though no prolongation is carried out.

Table 1 shows the multiresolution method with 3 levels to give larger PSNR-values than the 1-level method. Moreover, the accuracy achieved by the multiresolution method with 3 levels is about the same as with 5 levels. The numbers of iterations on the finest levels for the 3- and 5-level multiresolution methods are about the same. This, as well as numerical experiments with other images, indicates that 3 to 4 levels is appropriate for many image restoration problems. The quality of the restored images does not deteriorate by choosing more than 3 to 4 levels, but generally does not increase either. The smallest number of levels that yields restored images of the highest or close to the highest quality is related to the image content in the sense that the reduced image on the coarsest level should contain many of the structures (details) which characterize the finest image.

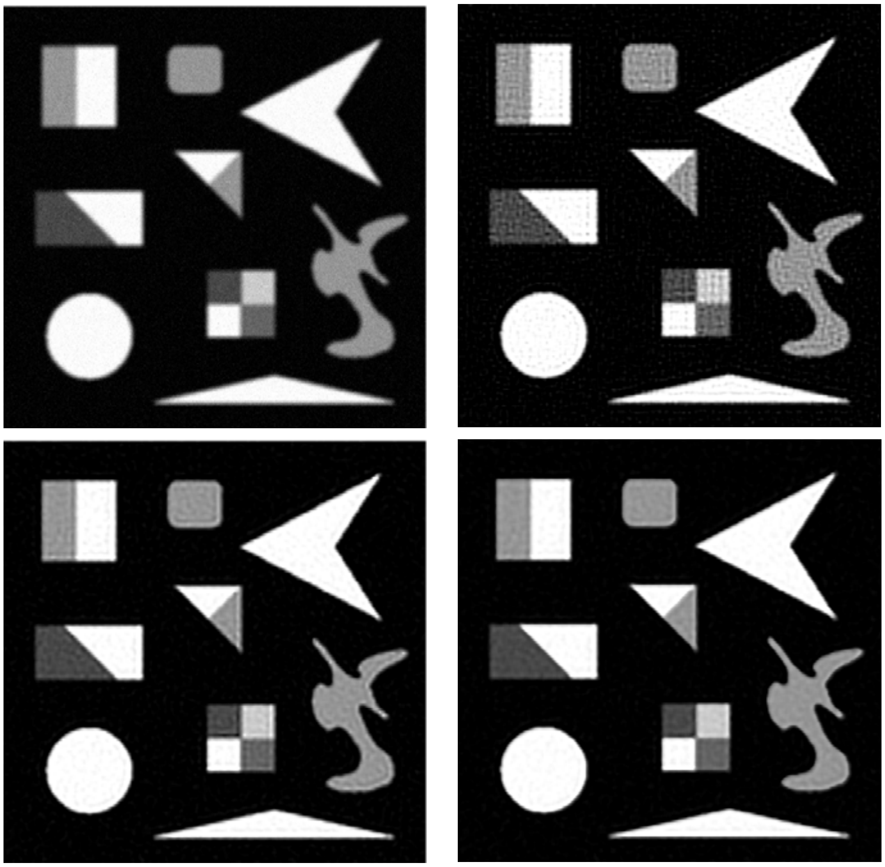

Figure 3. *Example 4.1: Blurred and noisy image with noise-level $\nu = 1 \cdot 10^{-1}$ and blur parameters* `band` $= 5$ *and* `sigma` $= 3$ *(top left); restored images with piecewise linear prolongation (top right), with nonlinear D_1-prolongation (bottom left), and with nonlinear D_2-prolongation (bottom right). Number of levels $\ell = 5$.*

Table 1

Example 4.1: PSNR and number of iterations (iter) as functions of the number of multiresolution levels ℓ and noise-levels ν for the same point spread function determined by `band` $= 5$ *and* `sigma` $= 3$.

ℓ	ν	PSNR lin prlg	PSNR D_1-prlg	PSNR D_2-prlg	iter lin prlg	iter D_1-prlg	iter D_2-prlg
1	$1 \cdot 10^{-2}$	28.84	-	-	8	-	-
3	$1 \cdot 10^{-2}$	29.41	29.89	30.80	9 12 6	9 12 6	9 12 6
5	$1 \cdot 10^{-2}$	29.41	29.89	30.80	1 1 9 12 6	1 1 9 12 6	1 1 9 12 6
1	$5 \cdot 10^{-2}$	26.53	-	-	4	-	-
3	$5 \cdot 10^{-2}$	26.76	26.91	27.97	5 4 2	5 4 2	5 4 3
5	$5 \cdot 10^{-2}$	26.76	26.91	27.94	1 1 5 4 2	1 1 5 4 2	1 1 5 4 3
1	$1 \cdot 10^{-1}$	25.70	-	-	3	-	-
3	$1 \cdot 10^{-1}$	26.29	26.43	27.13	4 3 2	4 3 2	4 3 2
5	$1 \cdot 10^{-1}$	26.28	26.43	27.12	1 1 4 3 2	1 1 4 3 2	1 1 3 3 2

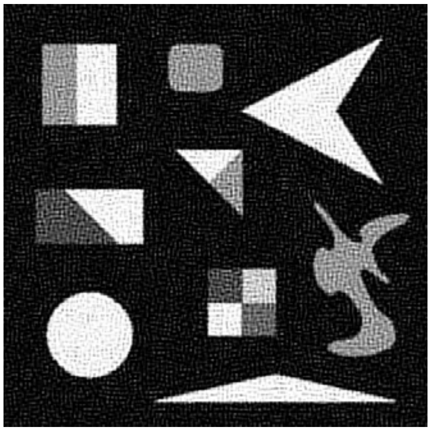

Figure 4. *Example 4.1: Effect of too many iterations. The image is determined by application of* 5 1*-level MR-*II *iterations to the restoration of the contaminated image shown in Figure 3 (top left). Propagated noise is clearly visible. Stopping Rule 2.1 suggests that only* 3 *iterations be carried out.*

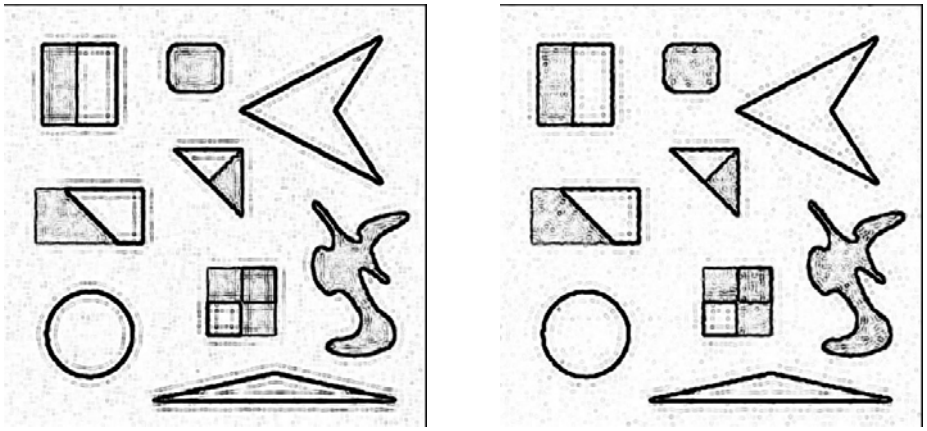

Figure 5. *Example 4.1: Edge maps for images restored by* 1*-level MR-*II *applied on the finest level (left) and by Algorithm 2.2 with* $\ell = 3$ *levels with linear prolongation (right). The available corrupted image is contaminated by blur and noise determined by* `band` $= 5$, `sigma` $= 5$, *and* $\nu = 5 \cdot 10^{-2}$.

The 1-level MR-II method applied to the restoration of the blurred and noisy image of Figure 3 (top left) yields after 3 iterations a restored image with PSNR 25.70; see Table 1. This number of iterations is determined by Stopping Rule 2.1. If we ignore this stopping rule and carry out 2 additional iterations, then the restored image in Figure 4 is obtained. The latter image is seen to be of fairly poor quality. It has PSNR 23.01. We conclude that it is important not to carry out too many iterations.

In order to gain further insight into the performance of Algorithm 2.2, we display edge maps determined by the edge detector of `gimp`, a public domain software tool for image processing, to restored images. The image to be restored is a blurred and noisy version of Figure 2 (left), corresponding to the parameters `band` $= 5$ and `sigma` $= 5$ of `blur.m`; the noise-level is $\nu = 5 \cdot 10^{-2}$. Figure 5 displays edge maps for restorations determined by 1-level

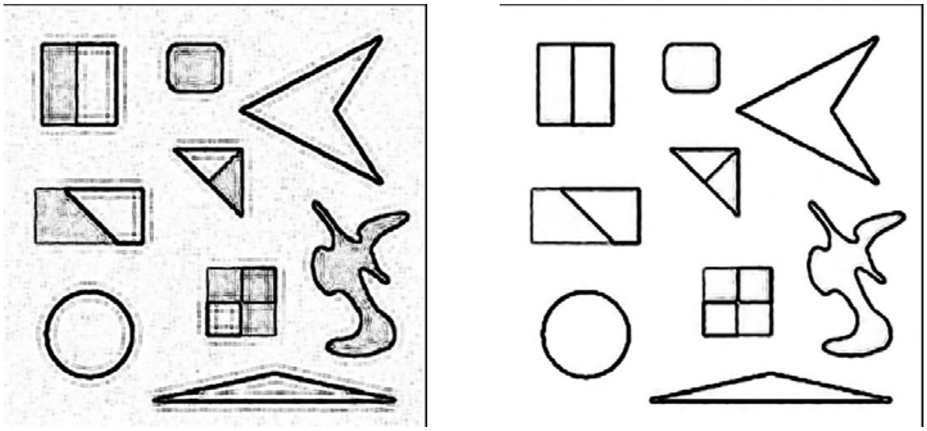

Figure 6. *Example 4.1: Edge maps for images restored by 1-level MR-II followed by D_2-smoothing (left) and by Algorithm 2.2 with $\ell = 3$ levels and D_2-prolongation (right). The available contaminated image is the same as for Figure 5.*

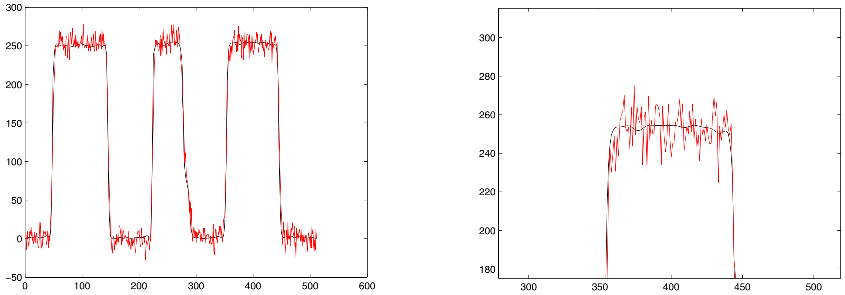

Figure 7. *Example 4.1: The smooth (black) graph displays the cross-section for the restored image determined by Algorithm 2.2 with $\ell = 3$ levels and D_2-prolongation (left). The right-hand figure shows a blow-up. The jagged (red) graphs show the corresponding cross-sections for the available contaminated image. The latter is contaminated by blur and noise determined by* $\texttt{band} = 5$, $\texttt{sigma} = 5$, *and* $\nu = 5 \cdot 10^{-2}$.

MR-II (left) and Algorithm 2.2 with $\ell = 3$ and piecewise linear prolongation. The 3-level multiresolution method yields the edge map with fewest artifacts. Figure 6 displays edge maps for images restored by 1-level MR-II followed by D_2-smoothing (left) and by Algorithm 2.2 with $\ell = 3$ levels and D_2-prolongation (right). A comparison of the left-hand side edge maps of Figures 5 and 6 shows that D_2-smoothing applied to the 1-level MR-II method does not provide a significantly improved edge map. The edge map of the highest quality by far is shown by Figure 6 (right). This comparison suggests that the use of several levels and a suitable prolongation operator is important for achieving accurate restorations.

The smoothing and edge-preserving effect of D_2-prolongation is also illustrated in Figure 7. The left-hand side of this figure displays the cross-section of a restored image (smooth black graph) along with the corresponding cross-section of the contaminated image (jagged red graph). The latter image is a blurred and noisy version of Figure 2 (left), corresponding to the parameters $\texttt{band} = 5$, $\texttt{sigma} = 5$, and $\nu = 5 \cdot 10^{-2}$. The restored image is determined

Table 2

Example 4.1: PSNR and number of iterations (iter) for several values of the parameters `band` *and* `sigma` *and fixed noise-level $\nu = 5 \cdot 10^{-2}$. Columns 3–4 report results for 1-level MR-II, columns 5–6 for Algorithm 2.2 with $\ell = 3$ levels and D_1-prolongation, and columns 7–8 for Algorithm 2.2 with $\ell = 3$ levels and D_2-prolongation.*

		PSNR	iter	PSNR	iter	PSNR	iter
`band`	`sigma`	1-level	1-level	D_1-prlg	D_1-prlg	D_2-prlg	D_2-prlg
3	3	29.45	3	30.08	1 3 2	30.88	1 3 2
3	5	29.23	3	30.03	1 3 2	30.84	1 3 2
5	3	26.61	4	26.85	5 4 2	27.98	5 4 3
5	5	26.48	4	27.58	16 4 3	28.49	16 4 3
9	3	25.83	5	26.10	4 4 3	26.78	4 4 3
9	5	23.68	6	24.32	8 5 3	25.14	8 5 4

by Algorithm 2.2 with $\ell = 3$ levels and D_2-prolongation. Figure 7 (right) shows a blow-up of the graphs on the left-hand side. The restored cross-section is seen to be fairly noise-free.

The improved quality achieved by the multilevel methods, when compared with 1-level MR-II, stems from the fact that the multilevel methods have available a better initial approximate solution when starting with the iterations on the finest level. This is important because the linear systems of equations (15) generally are numerically singular and therefore have numerically nonunique solutions; i.e., they have many not necessarily close approximate solutions with tiny residual error. The purpose of the nonlinear prolongation operators used on each level is to steer the computations toward a desirable approximate solution. Specifically, they are chosen to remove undesirable oscillations in the restored images.

Table 2 compares 1-level MR-II (columns 3–4) with Algorithm 2.2 using $\ell = 3$ levels with D_1- and D_2-prolongations (columns 5–6 and 7–8, respectively). The table shows results for several images that have been contaminated by blur corresponding to the values of the parameters `blur` and `sigma` displayed in columns 1–2. The noise-level is $\nu = 5 \cdot 10^{-2}$ for all images. The PSNR-values for images determined by Algorithm 2.2 with D_2-prolongation are the highest. We remark that the computational work, as measured by the number of matrix-vector product evaluations on the finest level, is smaller for Algorithm 2.2 than for 1-level MR-II. ■

Example 4.2. We illustrate the performance of Algorithm 2.2 when applied to the restoration of 256 × 256-pixel images with many small-scale details. The blur- and noise-free image is shown in Figure 2 (right). Contamination by severe blur, determined by the parameters `band` $= 5$ and `sigma` $= 3$ of the function `blur.m`, and by additive Gaussian noise of noise-level $\nu = 1 \cdot 10^{-1}$ yields the image in Figure 8 (top left). The restoration obtained with Algorithm 2.2 using $\ell = 4$ levels and piecewise linear prolongation is shown in Figure 8 (top right). The coarser levels have 128 × 128, 64 × 64, and 32 × 32 pixels. The restoration determined with Algorithm 2.2 using $\ell = 4$ levels and D_1-prolongation is shown in Figure 8 (bottom left); the image determined when D_2-prolongation is used instead is displayed in Figure 8 (bottom right). The images determined with D_1- and D_2-prolongations are seen to be of somewhat higher quality than the images obtained with piecewise linear prolongation.

We now compare Algorithm 2.2 to the nonlinear model (8). The contaminated image to be restored is shown in Figure 9; it is determined by `band` $= 3$, `sigma` $= 1$, and $\nu = 1 \cdot 10^{-2}$. The nonlinear PDE model (8) with $D = D_2$ is discretized by finite differences in space; time integration is carried out by the forward Euler method, advancing 500 time-steps of size $\tau = 2 \cdot$

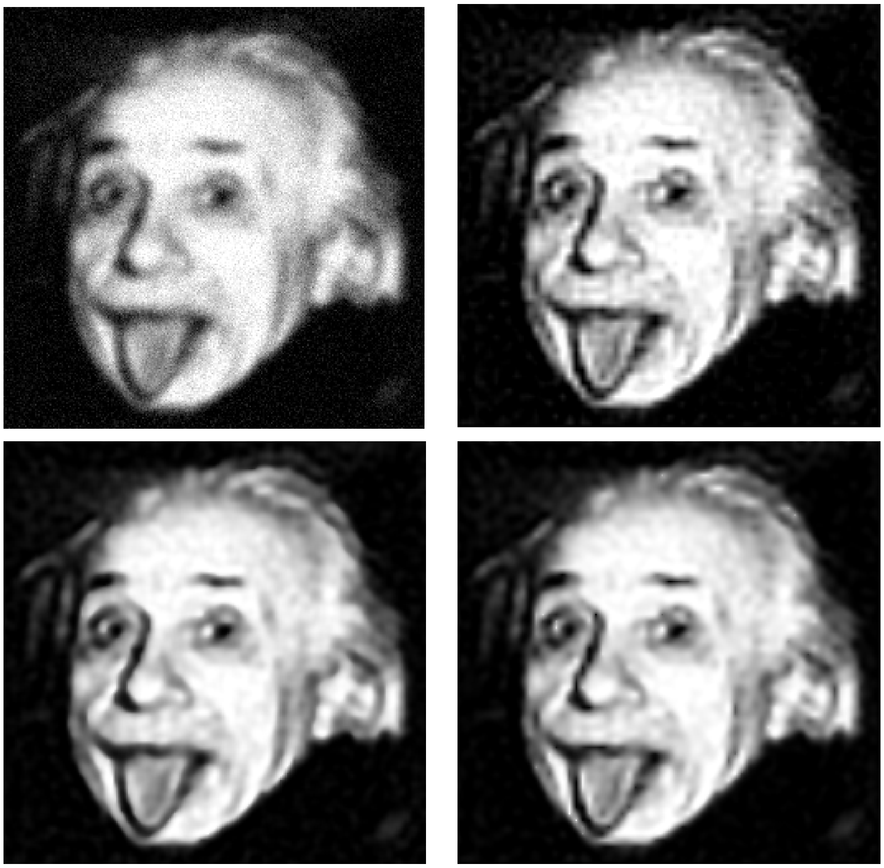

Figure 8. *Example* 4.2: *Blurred and noisy image corresponding to* `band` $= 5$, `sigma` $= 3$, *and* $\nu = 1 \cdot 10^{-1}$ *(top left), image restored by Algorithm* 2.2 *with* $\ell = 4$ *levels and piecewise linear prolongation (top right),* D_1*-prolongation (bottom left), and* D_2*-prolongation (bottom right).*

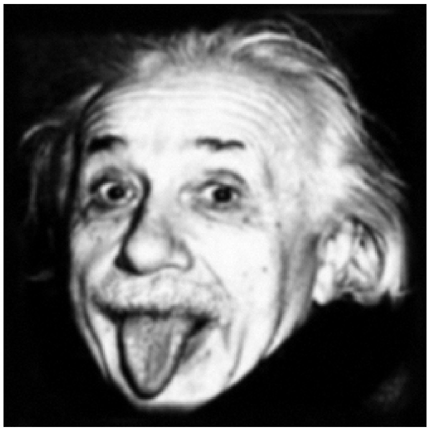

Figure 9. *Example* 4.2: *Image contaminated by blur and noise corresponding to* `band` $= 3$, `sigma` $= 1$, *and* $\nu = 1 \cdot 10^{-2}$.

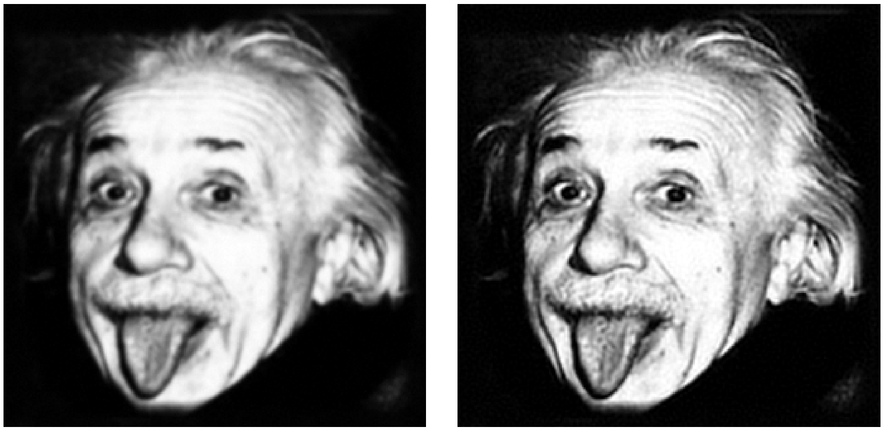

Figure 10. *Example 4.2: Image restored by the nonlinear PDE model (8) carrying out* 500 *time-steps with an explicit Euler method (left), and image restored by Algorithm 2.2 with $\ell = 3$ levels and D_2-prolongation (right).*

10^{-4}. These computations require the evaluation of 1000 matrix-vector products with the matrix A. Figure 10 (left) shows the restored image determined in this fashion. Algorithm 2.2 with $\ell = 3$ levels and D_2-prolongation yields the image in Figure 10 (right). The computation of the latter image requires only 4 matrix-vector product evaluations on the finest level; on the coarsest level 1 iteration is carried out and on the middle level 2. Thus, the computational effort required by Algorithm 2.2 is much smaller than for the nonlinear PDE model. Moreover, comparing the images of Figure 10 shows the restoration determined by Algorithm 2.2 to be more accurate. The PSNR-values provide a quantitative comparison; the image restored by the nonlinear PDE model has PSNR $= 28.84$, while the image determined by Algorithm 2.2 has PSNR $= 35.85$, a significantly larger value. We remark that the relatively poor performance of the nonlinear model (8) may be due to the fact that there are two regularization parameters, α and the length of the time-interval of integration. The optimal values of these parameters are difficult to determine in general nonlinear evolution PDE models (8), (13), and (14). In particular cases, such as for the TV-model, a steady state solution and an optimal α-value can be determined; see [5].

Tables 3 and 4 report applications of Algorithm 2.2 with $\ell = 3$ levels to the restoration of perturbed versions of the image in Figure 2 (right). Each table corresponds to one point spread function and different noise-levels ν and lists PSNR-values for the restored images, the prolongation operator used, and the number of iterations required on each level (iter). As can be expected, the quality of the restored images is higher (larger PSNR-values) when the noise-level in the available contaminated image is lower. Comparison of entries corresponding to the same noise-level in Tables 3 and 4 illustrates the very good performance of Algorithm 2.2 for the restoration of images with little blur.

Finally, we remark that the step-size for the PDE model has to be chosen smaller the more the image is blurred in order to avoid instability problems during the integration in time. Therefore, we chose fairly little blur in the image restored by the PDE model. The matrices A_i for this example are not very ill-conditioned. The multiresolution methods can easily be applied to the restoration of both mildly and severely blurred images. ■

Table 3

Example 4.2: PSNR and number of iterations (iter) for different noise-levels ν and prolongation operators. The point spread function is defined by `band` $= 5$ *and* `sigma` $= 3$.

	PSNR	PSNR	iter	iter
ν	D_1-prlg	D_2-prlg	D_1-prlg	D_2-prlg
$1 \cdot 10^{-2}$	26.09	26.35	9 11 5	9 11 6
$5 \cdot 10^{-2}$	24.27	24.49	5 4 2	5 4 3
$1 \cdot 10^{-1}$	23.69	23.74	4 2 2	4 3 2

Table 4

Example 4.2: PSNR and number of iterations (iter) for different noise-levels ν and prolongation operators. The point spread function is defined by `band` $= 3$ *and* `sigma` $= 3$.

	PSNR	PSNR	iter	iter
ν	D_1-prlg	D_2-prlg	D_1-prlg	D_2-prlg
$1 \cdot 10^{-2}$	29.07	29.10	1 18 4	1 18 4
$5 \cdot 10^{-2}$	27.13	26.96	1 3 2	1 3 2
$1 \cdot 10^{-1}$	25.49	25.22	1 2 1	1 2 1

5. Conclusion. Several methods for the restoration of images which have been contaminated by blur and noise are compared. The blurring operator and an estimate of the norm of the noise are assumed to be known. Visual inspection of the restored images shown in section 4 and quantitative evaluation of the results reported in the tables indicate that multiresolution methods with nonlinear edge-preserving prolongation operators yield more accurate restorations with generally less computational work than the MR-II method applied on the finest level only. Multiresolution methods also can provide restorations of higher accuracy than fully nonlinear models (8) and require significantly less computational effort.

We observe that, in general, multiresolution methods are applied to the solution of well-posed problems, because they can reduce the computational work significantly, when compared with 1-level solution methods. In the application to image restoration, the most important feature of our multiresolution method with nonlinear edge-preserving prolongation operators is the improved quality in the computed restorations, when compared with 1-level solution methods.

Acknowledgment. The second author would like to thank Fiorella Sgallari for making possible an enjoyable stay in Bologna, during which work on this paper was carried out.

REFERENCES

[1] F. A. Bornemann and P. Deuflhard, *The cascadic multigrid method for elliptic problems*, Numer. Math., 75 (1996), pp. 135–152.

[2] A. Buades, B. Coll, and J. M. Morel, *The staircasing effect in neighborhood filters and its solution*, IEEE Trans. Image Process., 15 (2006), pp. 1499–1505.

[3] D. Calvetti, B. Lewis, and L. Reichel, *On the choice of subspace for iterative methods for linear discrete ill-posed problems*, Int. J. Appl. Math. Comput. Sci., 11 (2001), pp. 1069–1092.

[4] Y. Cha and S. Kim, *Edge-forming methods for image zooming*, J. Math. Imaging Vis., 25 (2006), pp. 353–364.

[5] T. F. Chan and J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*, SIAM, Philadelphia, 2005.

[6] M. Donatelli and S. Serra-Capizzano, *On the regularizing power of multigrid-type algorithms*, SIAM J. Sci. Comput., 27 (2006), pp. 2053–2076.

[7] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[8] A. Handlovicová, K. Mikula, and F. Sgallari, *Semi-implicit complementary volume scheme for solving level set like equations in image processing and curve evolution*, Numer. Math., 93 (2003), pp. 675–695.

[9] M. Hanke, *Conjugate Gradient Type Methods for Ill-Posed Problems*, Longman Scientific and Technical, Harlow, UK, 1995.

[10] P. C. Hansen, *Regularization tools: A Matlab package for analysis and solution of discrete ill-posed problems*, Numer. Algorithms, 6 (1994), pp. 1–35.

[11] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, SIAM, Philadelphia, 1997.

[12] U. Hämarik and R. Palm, *On rules for stopping the conjugate gradient type methods in ill-posed problems*, Math. Model. Anal., 12 (2007), pp. 61–70.

[13] 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.

[14] S. Morigi, L. Reichel, F. Sgallari, and F. Zama, *Iterative methods for ill-posed problems and semiconvergent sequences*, J. Comput. Appl. Math., 193 (2006), pp. 157–167.

[15] M. K. Ng, R. H. Chan, and W.-C. Tang, *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM J. Sci. Comput., 21 (1999), pp. 851–866.

[16] P. Perona and J. Malik, *Scale-space and edge detection using anisotropic diffusion*, IEEE Trans. Pattern Anal. Mach. Intell., 12 (1990), pp. 629–639.

[17] L. Reichel and H. Sadok, *A new L-curve for ill-posed problems*, J. Comput. Appl. Math., to appear.

[18] L. Reichel and A. Shyshkov, *Cascadic multilevel methods for ill-posed problems*, J. Comput. Appl. Math., to appear.

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

[20] U. Trottenberg, C. W. Osterlee, and A. Schüler, *Multigrid*, Academic Press, Orlando, FL, 2001.

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

[22] J. Weickert, B. M. H. Romeny, and M. A. Viergever, *Efficient and reliable schemes for nonlinear diffusion filtering*, IEEE Trans. Image Process., 7 (1998), pp. 398–410.

[23] M. Welk, D. Theis, T. Brox, and J. Weickert, *PDE-based deconvolution with forward-backward diffusivities and diffusion tensors*, in Scale-Space and PDE Methods in Computer Vision, Lecture Notes in Comput. Sci. 3459, R. Kimmel, N. Sochen, and J. Weickert, eds., Springer, Berlin, 2005, pp. 585–597.