SIAM J. IMAGING SCIENCES
Vol. 8, No. 1, pp. 95–125

A New Augmented Lagrangian Approach for L^1-mean Curvature Image Denoising*

M. Myllykoski†, R. Glowinski‡, T. Kärkkäinen†, and T. Rossi†

Abstract. Variational methods are commonly used to solve noise removal problems. In this paper, we present an augmented Lagrangian-based approach that uses a discrete form of the L^1-norm of the mean curvature of the graph of the image as a regularizer, discretization being achieved via a finite element method. When a particular alternating direction method of multipliers is applied to the solution of the resulting saddle-point problem, this solution reduces to an iterative sequential solution of four subproblems. These subproblems are solved using Newton's method, the conjugate gradient method, and a partial solution variant of the cyclic reduction method. The approach considered here differs from existing augmented Lagrangian approaches for the solution of the same problem; indeed, the augmented Lagrangian functional we use here contains three Lagrange multipliers "only," and the associated augmentation terms are all quadratic. In addition to the description of the solution algorithm, this paper contains the results of numerical experiments demonstrating the performance of the novel method discussed here.

Key words. alternating direction methods of multipliers, augmented Lagrangian method, image denoising, image processing, mean curvature, variational model

AMS subject classifications. 68U10, 94A08, 53A05, 35A15

DOI. 10.1137/140962164

1. Introduction.

Let Ω be a bounded domain of $\mathbb{R}^2$ (a rectangle in practice). In the simplest form, denoising is a process in which a given noisy image $f : \Omega \to \mathbb{R}$ is separated into the actual image $u : \Omega \to \mathbb{R}$ and the remaining noise $g : \Omega \to \mathbb{R}$, that is, $f = u + g$. Variational methods, e.g., partial differential equation (PDE)–based (Euler–Lagrange equation) and nonlinear and nonsmooth optimization–based (corresponding energy functional), introduce a special family of techniques for image restoration and denoising in the general field of image processing and computer vision [14, 49]. A landmark in such techniques is the work by Perona and Malik related to anisotropic diffusion [41, 42]. Since then, many formulations and corresponding algorithms have been proposed, analyzed, realized, and utilized to improve the quality or understandability of digital images.

Let V be the space of restored functions, and let us consider the following minimization

*Received by the editors March 24, 2014; accepted for publication (in revised form) November 17, 2014; published electronically January 15, 2015.
http://www.siam.org/journals/siims/8-1/96216.html

†Department of Mathematical Information Technology, University of Jyväskylä, FI-40014, Jyväskylä, Finland (mirko.myllykoski@jyu.fi, tommi.karkkainen@jyu.fi, tuomo.j.rossi@jyu.fi). The research of the first author was supported by the Academy of Finland, grant 252549.

‡Department of Mathematics, University of Houston, Houston, TX 77204 (roland@math.uh.edu). The research of this author was supported by the Institute of Advanced Studies at the Hong Kong University of Science and Technology, and the Department of Mathematical Information Technology at the University of Jyväskylä.

problem:

$$\begin{cases} u \in V, \\ \mathcal{J}(u) \le \mathcal{J}(v) \quad \forall v \in V. \end{cases} \tag{1.1}$$

Typically

$$\mathcal{J}(v) = \mathcal{J}_f(v) + \varepsilon \mathcal{J}_r(v); \tag{1.2}$$

i.e., $\mathcal{J}$ consists of two terms, namely, (i) $\mathcal{J}_f(v)$, whose role is to fit (in a suitable norm) the denoised function to the noisy data f (the fidelity term), and (ii) the regularizing term $\varepsilon \mathcal{J}_r(v)$, where ε (> 0) is the regularization coefficient; examples of functionals $\mathcal{J}_r$ will be given below. The value of ε can be determined from the noise variance [10] if this information is known, or using some suitable heuristics [34].

Well-known variational approaches for image denoising relying on (1.2) include (but are not restricted to) the following:

BV (or TV) regularization: To be able to restore and denoise images with discontinuous intensity, a regularization using a norm that is not embedded in $C(\bar{\Omega})$ is needed. During the last two decades, the image denoising scene has been dominated by a method using such a regularization. The Rudin–Osher–Fatemi (ROF) method [44] relies on a discrete variant (obtained by finite difference discretization) of the minimization problem (1.1) with $\mathcal{J}$ defined by

$$\mathcal{J}(v) = \varepsilon \int_\Omega |\nabla v| \, dx + \frac{1}{2} \int_\Omega |f - v|^2 dx. \tag{1.3}$$

A natural candidate for the function space V is the space $BV(\Omega)$ of the functions with bounded variation over Ω. Problem (1.1) with $\mathcal{J}$ defined by (1.3) and close variants of it have motivated a large literature where their denoising properties, approximation, and iterative solution have been extensively discussed. We refer the reader to, e.g., [33, 34, 35, 45] and references therein for further information.

Euler's elastica: Euler's elastica as a prior curve model for image restoration was introduced in [39]. The work was continued, in connection with image inpainting, in [13]. In particular, Ambrosio and Masnou [1, 2] advocated using as regularizer the level set of v via the functional

$$\mathcal{J}_r(v) = \int_\Omega \left[a + b \left(\nabla \cdot \frac{\nabla v}{|\nabla v|} \right)^2 \right] |\nabla v| dx. \tag{1.4}$$

Let us return for a moment to the ROF model. Although the functional in (1.3) is convex, the nonreflexivity of the space $BV(\Omega)$ makes its analysis nontrivial, particularly the behavior of its solution when $\varepsilon \to 0$. This issue and many others, such as the derivation of dual formulations to (1.1) with $\mathcal{J}$ defined by (1.3), its discretization, the implementation, and the convergence of iterative solution methods (of the splitting type), are thoroughly discussed in [31].

Actually, as pointed out by [38] (see also [3], [12], and [51]), the ROF model has some significant drawbacks, such as the loss of image contrast, the smearing of corners, and the staircase effect. To remedy these unfavorable properties, several cures have been proposed (see [51] for a list of related references), among them the one introduced in [51], namely, instead of (1.1) with $\mathcal{J}$ defined by (1.3) use the following model:

$$u = \arg\min_{v\in V} \varepsilon \int_\Omega \left| \nabla\cdot \frac{\nabla v}{\sqrt{1+|\nabla v|^2}} \right| dx + \frac{1}{s}\int_\Omega |f-v|^s\, dx, \tag{1.5}$$

commonly known these days as the L^1-mean curvature denoising model; $s \geq 1$, $s = 2$ being the most common choice. The fidelity term in (1.5) is of the simplest form compared to the proposed formulations. In particular, as depicted in [40], one can, via adding two linear transformations to this model, address other important image processing tasks related to deconvolution, inpainting, and superresolution. The rationale for (1.5) is discussed in much detail in [51]. However, to the best of our knowledge, the variational problem (1.5) is not yet fully understood mathematically due to the nonconvexity and nonsmoothness of the functional in (1.5) and to the fact that a natural choice for V is not clear. Concerning this last issue, taking $V = BV(\Omega)$ is suggested in [51]. We have, however, a problem with such a choice since we think that the definition of V has to include conditions on the second derivatives. We can only hope that a variant of [31] dedicated to problem (1.5) will appear in the near future.

Considering the above situation, our goals in this paper are more modest and purely finite dimensional algorithmic. They can be summarized as follows:

1. Taking advantage of the relative simplicity of the formalism of the continuous problem, we derive in section 2 a (necessarily formal) augmented Lagrangian algorithm. Our algorithm is a simplified variant of the one considered in [52] since we use a projection on a nonconvex set to treat a nonlinear constraint instead of treating it by penalty-duality, which would imply one extra augmentation functional and the related Lagrange multiplier. Thus, our algorithms involve three augmentation terms instead of four and three Lagrange multipliers instead of four. Indeed, when several Lagrange multipliers are used, one of the main issues is their adjustment to optimize convergence. We will return to the details of this reduction in section 2.
2. Taking advantage of the augmented Lagrangian algorithm described in section 2, we define in section 3 a discrete version of this algorithm to be applied to a (kind of) mixed finite element approximation of problem (1.5). We choose finite element methods for approximating the problem instead of finite differences since, when applied on uniform triangulations (like the one in Figure 1), in particular, these finite element methods automatically generate finite difference approximations with, among other attributes, good accuracy, stability, and monotonicity properties. Moreover, the variational derivation of Galerkin/finite element approximations (like the one we use here) makes them the perfect match for the solution of problems from Calculus of Variations, such as (1.1) with $\mathcal{J}$ defined by (1.3) and (1.5) (see [24] for other examples). Another advantage of finite element approximations is their ability to handle nonuniform meshes, adaptively or not, which may be of interest for some applications. More precisely, the minimization of the discrete counterpart of the functional in (1.5)

is transformed into the iterative sequential solution of four subproblems: one being smooth but nonlinear in $\mathbb{R}^2$, one purely explicit vertexwise, and two linear with positive definite and symmetric coefficient matrices of a scalar- (assuming $s = 2$) and vector-valued type.

3. In section 5 we apply the algorithms defined in section 4 to the solution of a variety of test problems of variable complexity in order to evaluate the capability of the methodology discussed in this paper. The actual solution process is somewhat simplified by assuming that $s = 2$ (which was also recommended by Zhu and Chan [51]).

2. Augmented Lagrangian formulation and basic solution algorithm.

2.1. Some preliminary observations. Despite the fact that problem (1.5) is not fully understood mathematically, we are going to take advantage of the simplicity of its formalism to derive a formal solution algorithm of the augmented Lagrangian type. This algorithm will be useful since in section 3 we will take it as a model to define a finite dimensional analogue dedicated to the solution of a finite element approximation of problem (1.5).

Actually, augmented Lagrangian techniques have a well-established role in analyzing constrained optimization problems as well as in deriving general solution algorithms for such problems [5, 10, 22, 25, 29, 32]. With BV-regularization a reformulation with an augmented Lagrangian method can introduce one or two new variables to deal with ∇u in the nonsmooth regularization term (e.g., [15, 48, 40] and the references provided in the reviews therein) or additionally to represent u in the fidelity term [11, 50]. In the latter case, three subproblems (alternating directions) typically appear and are then solved using linear solvers, explicit formulae (projection or shrinking), and nonlinear optimization methods (for nonsmooth fidelity). Moreover, when more than one additional variable is introduced, one typically applies varying regularization parameters for the penalized constraint (e.g., [11, 50]). More examples where many variational formulations (including Euler's elastica) for image processing have been efficiently treated using augmented Lagrangian approaches can be found in, e.g., [20, 30, 46, 50].

As already mentioned in section 1, an augmented Lagrangian algorithm was used in [52] for the solution of (1.5). The augmented Lagrangian approach we apply in this paper is of the ALG-2 type [22, 29] (better known as ADMM for alternating direction methods of multipliers). The basic idea of ADMM and the convergence proof in a convex situation were presented in the 1970s by Glowinski and Marrocco [27] and Gabay and Mercier [23]. Augmented Lagrangian methods, with partly similar ingredients for the solution of other challenging problems, are described, e.g., in [8, 16, 17, 19, 18, 21, 28]; most of these problems are nonconvex, as is the one discussed here.

The solution method in [52] is also of the ALG-2 (or ADMM) type, but it uses a different (and, we think, more complicated) augmented Lagrangian functional as the functional involves four augmentation functionals and four Lagrange multipliers. What made us uncomfortable was the Lagrange multiplier treatment of the nonlinear nonsmooth condition $|\hat{\mathbf{q}}_1| - \hat{\mathbf{q}}_1 \cdot \hat{\mathbf{q}}_2 = 0$, where $\hat{\mathbf{q}}_1$ and $\hat{\mathbf{q}}_2$ belong (formally) to $(L^2(\Omega))^3$. The related condition in our approach is

$$\mathbf{q}_2 = \frac{\mathbf{q}_1}{\sqrt{1 + |\mathbf{q}_1|^2}}, \tag{2.1}$$

where $\mathbf{q}_1$ and $\mathbf{q}_2$ belong (formally) to $(L^2(\Omega))^2$. The discrete analogue of the aforementioned

equality will be treated by projection, avoiding those two terms associated with it in the augmented Lagrangian and reducing to three the number of Lagrange multipliers. This is a notable simplification considering that the adjustment of these parameters is one of the main issues associated with ADMM-type methods (it is discussed in [19] in a particular case). Also, in our approach, all the constraints treated by penalty-duality are linear, which is not the case in [52] (one of the constraints there is not only nonlinear but also nonsmooth).

Remark 1. At present, there is no available theory (as far as we know) for the convergence of ADMM methods for nonconvex problems, even in a finite dimension. Currently, the most popular publication concerning augmented Lagrangian and ADMM algorithms is certainly [7], a large review article (>100 pages) uniquely concerned with finite dimensional problems. The part of the article dedicated to nonconvex problems covers four (inconclusive) pages, suggesting that convergence proofs are difficult to obtain in most nonconvex cases. However, various investigations concerning nonconvex problems and comparisons with known solutions have shown the capability of augmented Lagrangian methods at solving nonconvex problems (as shown, for example, in [18] and [29]). This certainly encouraged us to apply this methodology to the solution of problem (1.5).

2.2. An augmented Lagrangian formulation. Assuming that a minimizer exists, the minimization problem (1.5) is clearly equivalent to

$$\begin{cases} (u, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \psi) \in E, \\ j(u, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \psi) \le j(v, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \varphi) \quad \forall\, (v, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \varphi) \in E, \end{cases} \tag{2.2}$$

where

$$j(v, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \varphi) = \varepsilon \int_\Omega |\varphi|\, dx + \frac{1}{s} \int_\Omega |f - v|^s\, dx. \tag{2.3}$$

Above,

$$E = \left\{ (v, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \varphi) : v \in V, (\mathbf{q}_1, \mathbf{q}_2) \in (L^2(\Omega))^{2\times 2}, \mathbf{q}_3 \in H(\Omega; \mathrm{div}), \right. \\ \left. \varphi \in L^2(\Omega),\ \mathbf{q}_1 = \nabla v,\ \mathbf{q}_2 = \frac{\mathbf{q}_1}{\sqrt{1 + |\mathbf{q}_1|^2}},\ \mathbf{q}_3 = \mathbf{q}_2,\ \varphi = \nabla\cdot \mathbf{q}_3 \right\}, \tag{2.4}$$

where

$$H(\Omega; \mathrm{div}) = \left\{ \mathbf{q} \in (L^2(\Omega))^2 : \nabla\cdot \mathbf{q} \in L^2(\Omega) \right\}. \tag{2.5}$$

Remark 2. In section 1, we mentioned that the choice of V in the denoising model (1.5) is a critical issue. Actually, a reasonable candidate is (for its simplicity) the Sobolev space $W^{2,1}(\Omega)$ since the two terms defining the cost functional in (1.5) make sense in that space (we recall that from the Rellich–Kondrachov compact imbedding theorem the injection of $W^{2,1}(\Omega)$ in $L^q(\Omega)$ is compact $\forall q \in [1, +\infty)$). From a practical (but formal) point of view, there is an advantage to taking $V = H^2(\Omega) (= W^{2,2}(\Omega))$ for the following reasons: (i) $H^2(\Omega)$ is a Hilbert

space; (ii) $H^2(\Omega)$ being dense in $W^{2,1}(\Omega)$, one obtains the same infimium when one minimizes the cost functional in (1.5) over these two spaces; and (iii) the above two spaces lead to the same discrete minimization problem.

Let us define

$$\Upsilon = [V \times \mathbf{E}_{12} \times H(\Omega;\mathrm{div}) \times L^2(\Omega)] \times [(L^2(\Omega))^2 \times (L^2(\Omega))^2 \times L^2(\Omega)], \tag{2.6}$$

with

$$\mathbf{E}_{12} = \left\{ (\mathbf{q}_1, \mathbf{q}_2) \in \left(L^2(\Omega)\right)^{2\times 2} : \mathbf{q}_2 = \frac{\mathbf{q}_1}{\sqrt{1+|\mathbf{q}_1|^2}} \right\}. \tag{2.7}$$

With problem (2.2) we associate the following augmented Lagrangian functional $\mathcal{L}_{\mathbf{r}} : \Upsilon \to \mathbb{R}$ (with $\mathbf{r} = (r_1, r_2, r_3), r_i > 0\ \forall i = 1, 2, 3$):

$$\begin{aligned} \mathcal{L}_{\mathbf{r}}(v, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \varphi; \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \mu_3) = {} & \varepsilon \int_\Omega |\varphi| dx + \frac{1}{s}\int_\Omega |f - v|^s dx \\ & + \frac{r_1}{2}\int_\Omega |\nabla v - \mathbf{q}_1|^2 dx + \int_\Omega \boldsymbol{\mu}_1 \cdot (\nabla v - \mathbf{q}_1) dx \\ & + \frac{r_2}{2}\int_\Omega |\mathbf{q}_2 - \mathbf{q}_3|^2 dx + \int_\Omega \boldsymbol{\mu}_2 \cdot (\mathbf{q}_2 - \mathbf{q}_3) dx \\ & + \frac{r_3}{2}\int_\Omega |\nabla\cdot \mathbf{q}_3 - \varphi|^2 dx + \int_\Omega \mu_3 (\nabla\cdot \mathbf{q}_3 - \varphi) dx, \end{aligned} \tag{2.8}$$

where $(\mathbf{q}_1, \mathbf{q}_2) \in \mathbf{E}_{12}$.

Now, suppose that the augmented Lagrangian $\mathcal{L}_{\mathbf{r}}$ has a saddle-point

$$\omega = (u, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \psi; \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \lambda_3) \in \Upsilon, \tag{2.9}$$

that is

$$\begin{aligned} \mathcal{L}_{\mathbf{r}}(u, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \psi; \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \mu_3) &\le \mathcal{L}_{\mathbf{r}}(u, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \psi; \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \lambda_3) \\ &\le \mathcal{L}_{\mathbf{r}}(v, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \varphi; \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \lambda_3), \end{aligned} \tag{2.10}$$

for all $(v, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \varphi; \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \mu_3) \in \Upsilon$.

It can be easily shown that if ω is a saddle-point of $\mathcal{L}_{\mathbf{r}}$ over Υ, then u is a solution of the minimization problem (1.5) and

$$\mathbf{p}_1 = \nabla u,\ \mathbf{p}_2 = \frac{\mathbf{p}_1}{\sqrt{1+|\mathbf{p}_1|^2}},\ \mathbf{p}_3 = \mathbf{p}_2,\ \text{and}\ \psi = \nabla\cdot \mathbf{p}_3. \tag{2.11}$$

2.3. The basic algorithm. A natural candidate for the solution of the saddle-point problem (2.9)–(2.10) is a particular ADMM called ALG-2 by various practitioners (see, e.g., [4, 22, 24, 29]). Among the several algorithms of the ALG-2 type, the following is considered in this paper.

Algorithm 1.

Input: $f, \varepsilon, s, \mathbf{r}, N$
Initialize$(u^0, \mathbf{p}_1^0, \mathbf{p}_2^0, \mathbf{p}_3^0, \psi^0; \boldsymbol{\lambda}_1^0, \boldsymbol{\lambda}_2^0, \lambda_3^0)$
for $n = 0, \ldots, N$ ***do***

$$(\mathbf{p}_1^{n+1}, \mathbf{p}_2^{n+1}) = arg\ min_{(\mathbf{q}_1,\mathbf{q}_2)\in \mathbf{E}_{12}} \mathcal{L}_{\mathbf{r}}\left(u^n, \mathbf{q}_1, \mathbf{q}_2, \mathbf{p}_3^n, \psi^n; \boldsymbol{\lambda}_1^n, \boldsymbol{\lambda}_2^n, \lambda_3^n\right) \tag{2.12}$$

$$\mathbf{p}_3^{n+1} = arg\ min_{\mathbf{q}_3\in H(\Omega;div)} \mathcal{L}_{\mathbf{r}}\left(u^n, \mathbf{p}_1^{n+1}, \mathbf{p}_2^{n+1}, \mathbf{q}_3, \psi^n; \boldsymbol{\lambda}_1^n, \boldsymbol{\lambda}_2^n, \lambda_3^n\right) \tag{2.13}$$

$$\psi^{n+1} = arg\ min_{\varphi\in L^2(\Omega)} \mathcal{L}_{\mathbf{r}}\left(u^n, \mathbf{p}_1^{n+1}, \mathbf{p}_2^{n+1}, \mathbf{p}_3^{n+1}, \varphi; \boldsymbol{\lambda}_1^n, \boldsymbol{\lambda}_2^n, \lambda_3^n\right) \tag{2.14}$$

$$u^{n+1} = arg\ min_{v\in V} \mathcal{L}_{\mathbf{r}}\left(v, \mathbf{p}_1^{n+1}, \mathbf{p}_2^{n+1}, \mathbf{p}_3^{n+1}, \psi^{n+1}; \boldsymbol{\lambda}_1^n, \boldsymbol{\lambda}_2^n, \lambda_3^n\right) \tag{2.15}$$

$$\boldsymbol{\lambda}_1^{n+1} = \boldsymbol{\lambda}_1^n + r_1(\nabla u^{n+1} - \mathbf{p}_1^{n+1})$$
$$\boldsymbol{\lambda}_2^{n+1} = \boldsymbol{\lambda}_2^n + r_2(\mathbf{p}_2^{n+1} - \mathbf{p}_3^{n+1})$$
$$\lambda_3^{n+1} = \lambda_3^n + r_3(\nabla\cdot \mathbf{p}_3^{n+1} - \psi^{n+1})$$

if *stopping criterion is satisfied* ***then***
return $(u^{n+1}, \mathbf{p}_1^{n+1}, \mathbf{p}_2^{n+1}, \mathbf{p}_3^{n+1}, \psi^{n+1})$
end if
end for
return *ERROR*

A more explicit formulation of subproblem (2.12) reads as follows:

$$\begin{aligned}(\mathbf{p}_1^{n+1}, \mathbf{p}_2^{n+1}) = \arg\min_{(\mathbf{q}_1,\mathbf{q}_2)\in \mathbf{E}_{12}} \Bigg[&\frac{1}{2}\int_\Omega \left(r_1|\mathbf{q}_1|^2 + r_2|\mathbf{q}_2|^2\right) dx \\ &- \int_\Omega (r_1\nabla u^n + \boldsymbol{\lambda}_1^n)\cdot \mathbf{q}_1 dx - \int_\Omega (r_2\mathbf{p}_3^n - \boldsymbol{\lambda}_2^n)\cdot \mathbf{q}_2 dx\Bigg].\end{aligned} \tag{2.16}$$

Similarly, the minimization problem (2.13) is equivalent to the following well-posed linear variational problem in $H(\Omega;\mathrm{div})$:

$$\begin{aligned}&\mathbf{p}_3^{n+1} \in H(\Omega;\mathrm{div}),\\ r_2\int_\Omega \mathbf{p}_3^{n+1}\cdot\mathbf{q}\, dx + r_3\int_\Omega \nabla\cdot\mathbf{p}_3^{n+1}\nabla\cdot\mathbf{q}\, dx &= \int_\Omega \left(r_2\mathbf{p}_2^{n+1} + \boldsymbol{\lambda}_2^n\right)\cdot\mathbf{q}\, dx\\ &+ \int_\Omega \left(r_3\psi^n - \lambda_3^n\right)\nabla\cdot\mathbf{q}\, dx,\ \forall \mathbf{q}\in H(\Omega;\mathrm{div}).\end{aligned} \tag{2.17}$$

Next, a more explicit formulation of the minimization problem (2.14) is given by

$$\begin{aligned}\psi^{n+1} = \arg\min_{\varphi\in L^2(\Omega)} \Bigg[&\varepsilon\int_\Omega |\varphi|\, dx + \frac{r_3}{2}\int_\Omega |\varphi|^2\, dx\\ &- \int_\Omega \left(r_3\nabla\cdot\mathbf{p}_3^{n+1} + \lambda_3^n\right)\varphi\, dx\Bigg].\end{aligned} \tag{2.18}$$

Finally, the minimization problem (2.15) is nothing but equivalent to the following well-posed, nonlinear, elliptic, variational problem (linear if $s = 2$):

$$
\begin{aligned}
& u^{n+1} \in V, \\
& r_1 \int_\Omega \nabla u^{n+1} \cdot \nabla v \, dx + \int_\Omega \left| u^{n+1} - f \right|^{s-2} (u^{n+1} - f) \, v \, dx \\
& \qquad = \int_\Omega \left(r_1 \mathbf{p}_1^{n+1} - \boldsymbol{\lambda}_1^n \right) \cdot \nabla v \, dx \quad \forall v \in V.
\end{aligned} \tag{2.19}
$$

Remark 3. Assuming the minimizing sequences converge to a limit, we do not know in which space the related weak convergence takes place and if the functional under consideration has the weak lower semicontinuity property necessary to guaranty the convergence to a minimizer. Indeed, since our main concern is mostly to find a simpler alternative to the method discussed in [52], we skip this theoretical aspect of the problem.

Remark 4. As mentioned in the introduction, the augmented Lagrangian approach discussed in this section is largely formal, unlike its finite element realization discussed in section 3. This formality yields to various variational crimes, one of them being to take φ in $L^2(\Omega)$, and consequently $\mathbf{q}_3$ in $H(\Omega; \mathrm{div})$, while the natural functional space for φ is obviously $L^1(\Omega)$. Actually, a similar variational crime is committed when associating (as done by various practitioners today) with the functional $\mathcal{J}$ defined by (1.3) the augmented Lagrangian

$$
\begin{aligned}
\mathcal{L}_r(v, \mathbf{q}; \boldsymbol{\mu}) = \varepsilon \int_\Omega |\mathbf{q}| \, dx + \frac{1}{2} \int_\Omega |f - v|^2 \, dx \\
+ \frac{r}{2} \int_\Omega |\nabla v - \mathbf{q}|^2 \, dx + \int_\Omega \boldsymbol{\mu} \cdot (\nabla v - \mathbf{q}) \, dx,
\end{aligned} \tag{2.20}
$$

which is well suited to operations in $H^1(\Omega)$, but definitely not in $BV(\Omega)$, which is the natural space in which to minimize the above functional $\mathcal{J}$. Of course, the finite dimensional analogues of (2.20) make sense, authorizing, for example, the use of ADMM to solve the corresponding minimization problem.

3. Finite element realization.

3.1. Generalities. The rationale for using finite elements instead of finite differences was given in the introduction. An inspection of relations (2.16)–(2.19) shows that none of them explicitly involve derivatives of an order higher than one, implying that finite element spaces consisting of piecewise polynomial functions are well suited for defining a discrete analogue of Algorithm 1. Moreover, the expected lack of smoothness of the solutions (or quasi solutions) of problem (1.5) strongly suggests employing low degree polynomials (of a degree typically less than or equal to one). Actually, the approximation we will use is of the mixed type, like those used, for example, in [9, 16, 21, 31]; it allows solving a nonsmooth fourth-order elliptic problem using approximation spaces commonly used for the solution of second-order elliptic problems. For a thorough discussion of mixed finite element methods and some applications, see [6].

Concerning the solution of problem (1.5), we assume that Ω is a rectangle and denote by $\partial\Omega$ the boundary of Ω. Since Ω is polygonal, we can triangulate it using a standard finite

element triangulation $\mathcal{T}_h$ (verifying therefore the assumptions listed in, e.g., [24]). A typical finite element triangulation (uniform here) is shown in Figure 1.

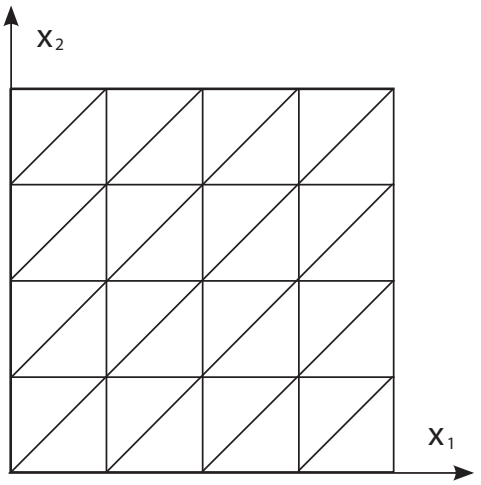

Figure 1. *A uniform triangulation of* $\Omega = (0,1)^2$.

We denote by Σ_h (respectively, Σ_{0h}) the finite set of the vertices of $\mathcal{T}_h$ (respectively, the finite set of the vertices of $\mathcal{T}_{0h}$ that do not belong to $\partial\Omega$). From now on we will assume that

$$\Sigma_{0h} = \{P_j\}_{j=1}^{N_{0h}} \text{ and } \Sigma_h = \Sigma_{0h} \cup \{P_j\}_{j=N_{0h}+1}^{N_h}, \tag{3.1}$$

where N_{0h} (respectively, N_h) is the number of elements of Σ_{0h} (respectively, Σ_h). Finally, we denote by Ω_j the polygon that is the union of those triangles of $\mathcal{T}_h$ that have P_j as a common vertex, and by $|\Omega_j|$ the measure of Ω_j.

3.2. Fundamental finite element spaces and the discrete divergence operator. Following Remark 2, we assume from now on that $V = H^2(\Omega)$. Using the appropriate discrete Green's formula, there is no difficulty in approximating the saddle-point problem (2.9)–(2.10) using classical C^0-conforming finite element spaces. To approximate the spaces $H^1(\Omega)$ and $H^2(\Omega)$, we will use

$$V_h = \left\{ v \in C^0(\bar{\Omega}) : v|_T \in P_1 \ \forall\, T \in \mathcal{T}_h \right\}. \tag{3.2}$$

Above, P_1 is the space of the polynomials of two variables of degree less than or equal to one.

Now, for $j = 1, \ldots, N_h$, let us uniquely define the shape function w_j associated with the vertex P_j by

$$\begin{cases} w_j \in V_h, \\ w_j(P_j) = 1, \\ w_j(P_k) = 0 \ \forall k,\ 1 \le k \le N_h,\ k \ne j. \end{cases} \tag{3.3}$$

The set $\mathcal{B}_h = \{w_j\}_{j=1}^{N_h}$ is a vector basis of V_h, and we have

$$v = \sum_{j=1}^{N_h} v(P_j) w_j \quad \forall v \in V_h. \tag{3.4}$$

Other finite element spaces will prove useful in what follows. The first, denoted by V_{0h}, is the subspace of V_h consisting of the functions vanishing on $\partial\Omega$, that is,

$$V_{0h} = \{v \in V_h : v(P_j) = 0 \, \forall j = N_{0h} + 1, \ldots, N_h\}; \tag{3.5}$$

we clearly have

$$v = \sum_{j=1}^{N_{0h}} v(P_j) w_j \quad \forall v \in V_{0h}. \tag{3.6}$$

The other space, denoted by $\mathbf{Q}_h$, is defined by

$$\mathbf{Q}_h = \left\{\mathbf{q} \in (L^\infty)^2 : \mathbf{q}|_T \in (P_0)^2 \, \forall \, T \in \mathcal{T}_h\right\}, \tag{3.7}$$

where P_0 is the space of those polynomials that are constant. We clearly have

$$\mathbf{q} = \sum_{T \in \mathcal{T}_h} (\mathbf{q}|_T)_{\chi_T} \quad \forall \mathbf{q} \in \mathbf{Q}_h, \tag{3.8}$$

where χ_T is the characteristic function of T and

$$\nabla V_h \subset \mathbf{Q}_h. \tag{3.9}$$

The linear space $\mathbf{Q}_h$ is a suitable candidate for the approximation of the space $H(\Omega; \mathrm{div})$, the main issue being to properly approximate the divergence of an arbitrary element of $\mathbf{Q}_h$. Suppose that $\mathbf{q} \in H(\Omega; \mathrm{div})$ and $v \in H_0^1(\Omega)$; we have (from the divergence theorem)

$$\int_\Omega \nabla\cdot \mathbf{q} \, v \, dx = -\int_\Omega \mathbf{q} \cdot \nabla v \, dx \quad \forall (v, \mathbf{q}) \in H_0^1(\Omega) \times H(\Omega; \mathrm{div}). \tag{3.10}$$

Suppose now that $\mathbf{q} \in \mathbf{Q}_h$; relation (3.10) suggests defining the discrete divergence operator div_h by: $\forall \mathbf{q} \in \mathbf{Q}_h$ we have

$$\begin{aligned} &\mathrm{div}_h \mathbf{q} \in V_{0h}, \\ &\int_\Omega (\mathrm{div}_h \mathbf{q}) \, v \, dx = -\int_\Omega \mathbf{q} \cdot \nabla v \, dx \quad \forall v \in V_{0h}, \end{aligned} \tag{3.11}$$

or equivalently $\forall \mathbf{q} \in \mathbf{Q}_h$, we have

$$\begin{aligned} &\mathrm{div}_h \mathbf{q} \in V_{0h}, \\ &\int_{\Omega_j} (\mathrm{div}_h \mathbf{q}) w_j \, dx = -\int_{\Omega_j} \mathbf{q} \cdot \nabla w_j \, dx \quad \forall j = 1, \ldots, N_{0h}. \end{aligned} \tag{3.12}$$

Since the functions $\mathbf{q}$ and ∇w_j are constant over the triangles of $\mathcal{T}_h$, the integrals on the right-hand sides of the equations in (3.12) can be computed exactly (and easily). On the other hand, to simplify the computation of the integrals on the left-hand sides, we advocate using the trapezoidal rule to compute (approximately this time) these integrals; we then obtain

$$(\mathrm{div}_h\mathbf{q})(P_j) = -\frac{3}{|\Omega_j|}\int_{\Omega_j} \mathbf{q}\cdot\nabla w_j\, dx \quad \forall j = 1,\dots,N_{0h}. \tag{3.13}$$

Remark 5. Albeit satisfactory conceptually, the use of the discrete Green's formulas (as done above to approximate the divergence operator) may lead to spurious oscillations (see [9] for dramatic evidence of this unwanted phenomenon), particularly when combined with low-order approximations, as in the case here. In order to eliminate (or at least strongly dampen) these unwanted oscillations, we advocate the following regularization (some say also stabilization) procedure: replace (3.12) or (3.13) by: $\forall \mathbf{q} \in \mathbf{Q}_h$ we have

$$\begin{aligned} &\mathrm{div}_h\mathbf{q} \in V_{0h}, \\ C\sum_{T\in\mathcal{T}_h}\int_T |T|\nabla(\mathrm{div}_h\mathbf{q})\cdot\nabla v\, dx + \int_\Omega (\mathrm{div}_h\mathbf{q})\, v\, dx &= -\int_\Omega \mathbf{q}\cdot\nabla v\, dx \quad \forall v \in V_{0h}, \end{aligned} \tag{3.14}$$

with $C(>0)$; boundary layer thickness considerations suggest $C \approx 1$. The above kind of Tychonov regularization procedure has been successful when applied to the solution of the Dirichlet problem for the Monge–Ampère equation in two dimensions, using mixed finite element approximations based on low-order C^0-conforming finite element spaces; see [9] for further details. Actually, it has not been tested yet for the solution of problem (1.5).

3.3. Discrete Lagrangian and discretized subproblems. Since our goal in this paper is to compute and approximate the solution of problem (1.5), using a discrete variant of Algorithm 1, a first step in that direction is to define an approximation of the augmented Lagrangian (2.8). The candidate functional $\mathcal{L}_{\mathbf{r}h} : (V_h \times \mathbf{Q}_h^3 \times V_{0h}) \times (\mathbf{Q}_h^2 \times V_{0h}) \to \mathbb{R}$ proposed in this

paper is the following:

$$\begin{aligned}\mathcal{L}_{\mathbf{r}h}(v,\mathbf{q}_1,\mathbf{q}_2,\mathbf{q}_3,\varphi;\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\mu_3) = {}& \frac{\varepsilon}{3}\sum_{j=1}^{N_{0h}}|\Omega_j||\varphi(P_j)| \\ &+ \frac{1}{3s}\sum_{j=1}^{N_h}|\Omega_j||f(P_j)-v(P_j)|^s \\ &+ \frac{r_1}{2}\int_\Omega|\nabla v-\mathbf{q}_1|^2dx+\int_\Omega\boldsymbol{\mu}_1\cdot(\nabla v-\mathbf{q}_1)dx \\ &+ \frac{r_2}{2}\int_\Omega|\mathbf{q}_2-\mathbf{q}_3|^2dx+\int_\Omega\boldsymbol{\mu}_2\cdot(\mathbf{q}_2-\mathbf{q}_3)dx \\ &+ \frac{r_3}{6}\sum_{j=1}^{N_{0h}}|\Omega_j||(\mathrm{div}_h\mathbf{q}_3)(P_j)-\varphi(P_j)|^2 \\ &+ \frac{1}{3}\sum_{j=1}^{N_{0h}}|\Omega_j|\mu_3(P_j)\left[(\mathrm{div}_h\mathbf{q}_3)(P_j)-\varphi(P_j)\right],\end{aligned} \tag{3.15}$$

where $(\mathbf{q}_1,\mathbf{q}_2)\in\mathbf{E}_{12h}$ with

$$\mathbf{E}_{12h}=\left\{(\mathbf{q}_1,\mathbf{q}_2)\in\mathbf{Q}_h^2:\mathbf{q}_2=\frac{\mathbf{q}_1}{\sqrt{1+|\mathbf{q}_1|^2}}\right\}. \tag{3.16}$$

Based on $\mathcal{L}_{\mathbf{r}h}$, subproblem (2.12) can be approximated by

$$\begin{aligned}(\mathbf{p}_1^{n+1},\mathbf{p}_2^{n+1}) = \arg\min_{(\mathbf{q}_1,\mathbf{q}_2)\in\mathbf{E}_{12h}}\Bigg[&\frac{1}{2}\int_\Omega\left(r_1|\mathbf{q}_1|^2+r_2|\mathbf{q}_2|^2\right)dx \\ &-\int_\Omega(r_1\nabla u^n+\boldsymbol{\lambda}_1^n)\cdot\mathbf{q}_1dx-\int_\Omega(r_2\mathbf{p}_3^n-\boldsymbol{\lambda}_2^n)\cdot\mathbf{q}_2dx\Bigg].\end{aligned} \tag{3.17}$$

Since functional (3.17) does not contain derivatives of $\mathbf{q}_1$ and $\mathbf{q}_2$, its minimization can be performed pointwise (in practice on the triangles of the finite element triangulation $\mathcal{T}_h$). This leads to the solution, a.e. in Ω, of a four-dimensional problem of the following type:

$$(\mathbf{x}_1,\mathbf{x}_2)=\arg\min_{(\mathbf{y}_1,\mathbf{y}_2)\in\mathbf{e}_{12}}\left[\frac{1}{2}\left(r_1|\mathbf{y}_1|^2+r_2|\mathbf{y}_2|^2\right)-\mathbf{b}_1\cdot\mathbf{y}_1-\mathbf{b}_2\cdot\mathbf{y}_2\right], \tag{3.18}$$

with $\mathbf{b}_1,\mathbf{b}_2\in\mathbb{R}^2$ and

$$\mathbf{e}_{12}=\left\{(\mathbf{y}_1,\mathbf{y}_2)\in\mathbb{R}^2\times\mathbb{R}^2:\mathbf{y}_2=\frac{\mathbf{y}_1}{\sqrt{1+|\mathbf{y}_1|^2}}\right\}. \tag{3.19}$$

The term $\mathbf{y}_2$ from (3.18) can be easily eliminated using the nonlinear relation in (3.19). This leads to the following unconstrained (and nonconvex) two-dimensional problem:

$$\mathbf{x}_1=\arg\min_{\mathbf{y}\in\mathbb{R}^2}\left[\frac{|\mathbf{y}|^2}{2}\left(r_1+\frac{r_2}{1+|\mathbf{y}|^2}\right)-\left(\mathbf{b}_1+\frac{\mathbf{b}_2}{\sqrt{1+|\mathbf{y}|^2}}\right)\cdot\mathbf{y}\right]. \tag{3.20}$$

Since the objective function in (3.20) is differentiable, an obvious choice for the solution of the problem is Newton's method.

Problem (3.20) can be reduced even further by observing that if $\mathbf{x}$ is a solution of (3.20), then it follows from the Schwarz inequality in $\mathbb{R}^2$ that $\mathbf{x}$ and the vector $\mathbf{b}_1 + \frac{\mathbf{b}_2}{\sqrt{1+|\mathbf{x}|^2}}$ are positively co-linear, that is, there is $\alpha \geq 0$ such that

$$\mathbf{x}_1 = \alpha \left(\mathbf{b}_1 + \frac{\mathbf{b}_2}{\sqrt{1+|\mathbf{x}_1|^2}} \right). \tag{3.21}$$

It follows from (3.20) and (3.21) that

$$\begin{aligned} \mathbf{x}_1 &= \alpha \left(\mathbf{b}_1 + \frac{\mathbf{b}_2}{\sqrt{1+\rho^2}} \right), \\ (\rho, \alpha) &= \arg\min_{(\sigma,\tau)\in A} \left[\frac{\tau^2}{2} \left| \mathbf{b}_1 + \frac{\mathbf{b}_2}{\sqrt{1+\sigma^2}} \right|^2 \left(r_1 + \frac{r_2}{1+\sigma^2} \right) \right. \\ &\quad \left. - \tau \left| \mathbf{b}_1 + \frac{\mathbf{b}_2}{\sqrt{1+\sigma^2}} \right|^2 \right], \end{aligned} \tag{3.22}$$

where

$$A = \left\{ (\sigma, \tau) \in \mathbb{R}^+ \times \mathbb{R}^+ : \tau \left| \mathbf{b}_1 + \frac{\mathbf{b}_2}{\sqrt{1+\sigma^2}} \right| = \sigma \right\}. \tag{3.23}$$

Now clearly, due to (3.23), we have

$$\alpha = \frac{\rho}{\left| \mathbf{b}_1 + \frac{\mathbf{b}_2}{\sqrt{1+\rho^2}} \right|}, \tag{3.24}$$

and thus

$$\rho = \arg\min_{\sigma\in\mathbb{R}^+} \left[\frac{\sigma^2}{2} \left(r_1 + \frac{r_2}{1+\sigma^2} \right) - \sigma \left| \mathbf{b}_1 + \frac{\mathbf{b}_2}{\sqrt{1+\sigma^2}} \right| \right]. \tag{3.25}$$

Similarly, subproblem (2.17) can be approximated by

$$\begin{aligned} &\mathbf{p}_3^{n+1} \in \mathbf{Q}_h, \\ &r_2 \int_\Omega \mathbf{p}_3^{n+1} \cdot \mathbf{q}\, dx + \frac{r_3}{3} \sum_{j=1}^{N_{0h}} |\Omega_j| (\mathrm{div}_h \mathbf{p}_3^{n+1})(P_j)(\mathrm{div}_h \mathbf{q})(P_j) \\ &= \int_\Omega (r_2 \mathbf{p}_2^{n+1} + \boldsymbol{\lambda}_2^n) \cdot \mathbf{q}\, dx + \frac{1}{3} \sum_{j=1}^{N_{0h}} |\Omega_j| (r_3 \psi^n - \lambda_3^n)(P_j)(\mathrm{div}_h \mathbf{q})(P_j) \\ &\qquad \forall \mathbf{q} \in \mathbf{Q}_h. \end{aligned} \tag{3.26}$$

The bilinear functional on the left-hand side of (3.26) being positive definite and symmetric, an obvious choice for the solution of the above problem is the conjugate gradient algorithm, initialized with the vector-valued function $\mathbf{p}_3^n$.

Subproblem (2.18) can be approximated by

$$\psi = \arg\min_{\varphi \in V_{0h}} \left[\varepsilon \sum_{j=1}^{N_{0h}} |\Omega_j||\varphi(P_j)| + \frac{r_3}{2} \sum_{j=1}^{N_{0h}} |\Omega_j||\varphi(P_j)|^2 - \sum_{j=1}^{N_{0h}} |\Omega_j|(r_3 \mathrm{div}_h \mathbf{p}_3^{n+1} + \lambda_3^n)(P_j)\varphi(P_j) \right]. \tag{3.27}$$

Let $X_j^n = (r_3 \mathrm{div}_h \mathbf{p}_3^{n+1} + \lambda_3^n)(P_j)$. The closed form solution of subproblem (3.27) is given by

$$\psi^{n+1}(P_j) = \frac{1}{r_3} \operatorname{sgn}(X_j^n) \max(0, |X_j^n| - \varepsilon) \quad \forall j = 1, \ldots, N_{0h}. \tag{3.28}$$

Finally, subproblem (2.19) can be approximated by

$$\begin{aligned} & u^{n+1} \in V_h, \\ & r_1 \int_\Omega \nabla u^{n+1} \cdot \nabla v \, dx + \frac{1}{3} \sum_{j=1}^{N_h} |\Omega_j||(u^{n+1} - f)(P_j)|^{s-2}(u^{n+1} - f)(P_j)v(P_j) \\ & \qquad = \int_\Omega (r_1 \mathbf{p}_1^{n+1} - \boldsymbol{\lambda}_1^n) \cdot \nabla v \, dx \quad \forall v \in V_h. \end{aligned} \tag{3.29}$$

Problem (3.29) could be solved, for $1 \le s < 2$, for example, using a semismooth Newton method or a nonlinear overrelaxation method like that discussed in [26]. Alternately, if $s = 2$, problem (3.29) reduces to

$$\begin{aligned} & u^{n+1} \in V_h, \\ & r_1 \int_\Omega \nabla u^{n+1} \cdot \nabla v \, dx + \frac{1}{3} \sum_{j=1}^{N_h} |\Omega_j|(u^{n+1} - f)(P_j)v(P_j) \\ & \qquad = \int_\Omega (r_1 \mathbf{p}_1^{n+1} - \boldsymbol{\lambda}_1^n) \cdot \nabla v \, dx \quad \forall v \in V_h. \end{aligned} \tag{3.30}$$

It that case, a wide variety of methods could be applied. This article advocates a method called radix-4 partial solution variant of the cyclic reduction (PSCR) [36, 37, 43, 47] in the cases where the discretization mesh is orthogonal. In other cases, subproblem (3.30) could be solved, for example, using multigrid-type methods.

4. Implementation.

Subproblems excluding the first one can be addressed in a straightforward fashion. Subproblem (3.26) is solved using the conjugate gradient method without any preconditioning since we use a uniform mesh in practice. The subproblem (3.30) is solved using the radix-4 PSCR method. Finally, the solution of subproblem (3.27) is a simple triangle-wise operation.

Before a digital image $\hat{v} : \{1, \dots, \hat{W}\} \times \{1, \dots, \hat{H}\} \to \mathbb{R}$ can be presented as a member of the finite element space V_h, we must first choose how we are going to deal with the dimensions of Ω and the spatial discretization step h. When one of these is chosen, the other is also fixed. We decided to normalize the dimensions of $\Omega = (0, W) \times (0, H)$ by setting $\max(W, H) = 1$. As a result, the spatial discretization step h is fixed to $1/(\max(\hat{W}, \hat{H}) - 1)$, implying $h \ll 1$ in practice. The fact that the spatial discretization step depends on the pixel size of the image could potentially have an effect on the final denoised image. See Remark 7 for further discussion.

4.1. Solution of the first subproblem. Apparently the most involved part of the discrete analogue of Algorithm 1 is the solution of the first subproblem (3.17). We could solve either the two-dimensional form (3.20) or the one-dimensional form (3.25). Depending on $\mathbf{b}_1, \mathbf{b}_2$ and r_1, r_2 both forms can have multiple local minimas due to the nonconvex nature of the mean curvature. Our actual realization first applies Newton's method to the two-dimensional form (3.20) starting from obvious initial guess $\mathbf{p}_1^n(x)$. Assuming that the method converges and the achieved solution is actually a local minimum, we then use the explicit relation (3.21) to test the obtained solution candidate. Only then is the solution candidate accepted.

If Newton's method fails, we proceed with the one-dimensional form (3.25) and apply the well-known bisection method. Some fine-tuning is needed because the best local mimima should be obtained to guarantee overall convergence. Hence, our actual heuristic algorithm for the minimization of the one-dimensional form (3.25) reads as follows:

Algorithm 2.

$\eta = \lceil \log_4(\sqrt{2}h^{-1}) \rceil$
$L = \{[4^{k-1}, 4^k] : k = 1, \dots, \eta\} \cup \{\{0\}, [0, 1], [4^\eta, +\infty[\}$
$K = \left\{ bisection(\Psi, l, \frac{1}{10}) \in \mathbb{R} : l \in L \right\}$
$\bar{k} = arg\ min_{k \in K}\ \Psi(k)$
$\rho = bisection(\Psi, [\bar{k} - \frac{1}{10}, \bar{k} + \frac{1}{10}], \sigma)$

$$\mathbf{p}_{bisection}(x) = \frac{\rho}{\left| \mathbf{b}_1 + \frac{\mathbf{b}_2}{\sqrt{1+\rho^2}} \right|} \left(\mathbf{b}_1 + \frac{\mathbf{b}_2}{\sqrt{1+\rho^2}} \right)$$

Above, Ψ is the objective function from the one-dimensional form (3.25), and the function $bisection(\Psi, l, \varsigma)$ applies the bisection search to the function Ψ on the interval l with accuracy tolerance ς. If interval l is unbounded, then interval l is adjusted by moving the right-hand side boundary in such a way that the value of the function Ψ is larger on the right-hand side boundary than it is on the left-hand side boundary. In addition, h is the spatial discretization step and σ (> 0) is the requested accuracy. Finally, we set

$$\mathbf{p}_1^{n+1}(x) = \arg\ \min_{\mathbf{q} \in \{\mathbf{p}_{\text{newton}}, \mathbf{p}_{\text{bisection}}, \mathbf{p}_1^n(x)\}} \Phi(\mathbf{q}), \tag{4.1}$$

where Φ is the objective function from (3.20) and $\mathbf{p}_{\text{newton}}$ is the solution (or the result of the last iteration) obtained by Newton's method.

We obtained this heuristic algorithm by observing the cases where Newton's method failed to solve the two-dimensional form (3.20) and by investigating the behavior of the one-dimensional form (3.25) on those cases. Since $\mathbf{p}_1^n$ begins to approximate the gradient of u^n as the discrete analogue of Algorithm 1 progresses and the solution of the one-dimensional

form (3.25) is the length of $\mathbf{p}_1^n$, it makes sense to concentrate on the interval $[0, \sqrt{2}h^{-1}]$ (assuming $\mathrm{Im}(u^n) \subset [0,1]$). The overall shape of the graph motivated us to divide the interval into subintervals as described above. In rare cases, the global mimima was actually located outside the range $[0, \sqrt{2}h^{-1}]$, which also had to be taken into account.

In practice, the combination of the two aforementioned algorithms works well in almost all cases. Newton's method converges quickly for most of the triangles of $\mathcal{T}_h$; for the few triangles for which Newton's method fails, one uses Algorithm 2. The cases where both algorithms fail are rare and are limited to individual triangles.

4.2. Initialization and stopping criterion. Concerning the use of the discrete variant of Algorithm 1, several issues will be addressed, obvious ones being initialization, stopping criterion, and the choice of the Lagrange multipliers $\mathbf{r}$ and ε. The choice of parameters $\mathbf{r}$ and ε is addressed in section 5, and thus only the initialization and stopping criterion will be discussed here.

A number of different kinds of initialization methods were considered; the most prominent were

$$u^0 = f,\ \mathbf{q}_1^0 = \nabla u^0,\ \mathbf{q}_2^0 = \frac{\mathbf{q}_1^0}{\sqrt{1+|\mathbf{q}_1^0|^2}},\ \mathbf{q}_3^0 = \mathbf{q}_2^0,\ \psi^0 = \nabla\cdot\mathbf{q}_3^0, \quad \boldsymbol{\lambda}_1^0 = \boldsymbol{\lambda}_2^0 = 0,\ \lambda_3^0 = 0, \tag{4.2}$$

and

$$u^0 = 0,\ \mathbf{q}_1^0 = \mathbf{q}_2^0 = \mathbf{q}_3^0 = 0,\ \psi^0 = 0, \quad \boldsymbol{\lambda}_1^0 = \boldsymbol{\lambda}_2^0 = 0,\ \lambda_3^0 = 0. \tag{4.3}$$

In the case of initialization (4.2), only the term corresponding to the mean curvature of the image surface is nonzero in the functional $\mathcal{L}_{\mathbf{r}h}$; and, in the case of initialization (4.3), only the term $|u-f|^s$ is nonzero. These differences play a major role in the overall behavior of the algorithm; the effect of both initializations will be discussed in further detail in section 5.

The second issue to be addressed is the stopping criterion. Several candidates were considered, such as

$$\frac{|u^n - u^{n+1}|_\infty}{|u^n|_\infty} < \delta,\ \frac{|u^n - u^{n+1}|_2}{|u^n|_2} < \delta,\ \text{and}\ \frac{|\mathcal{J}(u^n) - \mathcal{J}(u^{n+1})|}{|\mathcal{J}(u^n)|} < \delta, \tag{4.4}$$

where $\delta > 0$. The following criterion was found to be the most straightforward:

$$\frac{|\mathcal{L}_{\mathbf{r}h}(\omega^n) - \mathcal{L}_{\mathbf{r}h}(\omega^{n+1})|}{|\mathcal{L}_{\mathbf{r}h}(\omega^n)|} < \delta, \tag{4.5}$$

where $\omega^n = (u^n, \mathbf{p}_1^n, \mathbf{p}_2^n, \mathbf{p}_3^n, \psi^n; \boldsymbol{\lambda}_1^n, \boldsymbol{\lambda}_2^n, \lambda_3^n)$.

The aforementioned criterion works well as long as the Lagrange multipliers r_1, r_2, and r_3 are selected to be large enough so that they accurately enforce the equality constraints in (2.4).

5. Numerical results. In order to demonstrate the functionality of the discrete variant of Algorithm 1, we applied it against a large variety of test problems. These problems include synthetic and photographic images. For all these test problems, the values of the noise function g are uniformly distributed on the closed interval $[-p, p]$, $p > 0$. We took $\delta = 10^{-4}$ for the stopping criterion defined by (4.5). All images are grayscale, and the original images are scaled to the range $[0, 1]$.

5.1. Choice of initialization method and parameters. The behavior of the algorithm varied drastically depending on its initialization. Initialization (4.2) had a tendency to cause extremely slow convergence and an imperceptible low decrease of the value of the objective function in (1.5). By adjusting the parameters ε and $\mathbf{r}$ accordingly, reasonably good results were obtained in some cases. Usually this required a large parameter ε and small values for the Lagrange multipliers r_1, r_2, and r_3. However, each Lagrange multiplier combination worked only for a specific problem, and the undertaking of finding a Lagrange multiplier combination applicable to all problems proved futile. When the method was successfully initialized this way, it retained a considerable amount of detail while leaving some residual noise.

On the other hand, the initialization (4.3) caused a completely different behavior as the convergence was much faster, particularly during the first few tens of iterations. Finding a suitable Lagrange multiplier combination was difficult. The equality constraints in (2.4) and a crude dimensional analysis suggest that if $|\nabla u|^2 \ll 1$, the augmentation functionals behave such that $\mathbf{p}_1 \sim h^{-1}, \mathbf{p}_2 \sim h^{-1}, \mathbf{p}_3 \sim h^{-1}$, and $\psi \sim h^{-2}$. Thus, taking into account the homogeneity considerations, we should choose the following: $\varepsilon \sim h^2, r_1 \sim h^2, r_2 \sim h^2$, and $r_3 \sim h^4$. On the other hand, the same analysis suggests that for $|\nabla u|^2 \gg 1$, the augmentation functionals behave such that $\mathbf{p}_1 \sim h^{-1}, \mathbf{p}_2 \sim 1, \mathbf{p}_3 \sim 1$, and $\psi \sim h^{-1}$, and thus we should choose the following: $\varepsilon \sim h, r_1 \sim h^2, r_2 \sim 1$, and $r_3 \sim h^2$. In addition, from the formulation of subproblem (3.26), it can be seen that $h^2 r_2 \approx r_3$ could be a suitable choice because it would balance the left-hand side terms. Otherwise the conjugate gradient method would converge extremely slowly.

However, by applying the discrete analogue of Algorithm 1 against a large number of test problems and observing the residuals associated with the equality constraints in (2.4), we concluded that we should choose $\varepsilon \sim h$, $r_1 \sim h, r_2 \sim 1$, and $r_3 \sim h^2$. This was due to the fact that the equality constraint associated with the Lagrange multiplier r_1 did not converge when $r_1 \sim h^2$. Further testing leads us to the following Lagrange multiplier combination:

$$\begin{aligned} \varepsilon &= r_0 h, \\ r_1 &= 10 r_0 h, \\ r_2 &= 5 r_0, \\ r_3 &= 5 r_0 h^2, \end{aligned} \tag{5.1}$$

where $r_0 (> 0)$ is a parameter that can be tuned depending on the amount of noise.

In (5.1), the Lagrange multipliers r_1, r_2, and r_3 are large enough to accurately enforce the equality constraints in (2.4) while keeping the convergence speed reasonable. When one combines the initialization (4.3) with the above-mentioned Lagrange multipliers, the method removes a considerable amount of noise while filtering out some detail. From these observations

it was decided to use initialization (4.3) and the Lagrange multiplier combination (5.1) for all examples and comparisons.

Remark 6. A "simple" way to fix the problem with the selection of ε is to take

$$\varepsilon = Ch^{1+\frac{1}{1+|\nabla u|^2}}, \tag{5.2}$$

with C on the order of 1 (other nonlinear functions of $|\nabla u|$ are possible). We intend to investigate this approach in another paper. A simple way to use this variable in space parameter ε is as follows:

1. Solve the image restoration problem using a simpler method (based on BV-regularization, for example). Call u_0 the solution to this problem.
2. Set $\varepsilon_0 = Ch^{1+\frac{1}{1+|\nabla u_0|^2}}$.
3. For $i = 1, \ldots, M$, solve

$$u_i = \arg\min_{v\in V} \int_\Omega \varepsilon_{i-1} \left| \nabla \cdot \frac{\nabla v}{\sqrt{1+|\nabla v|^2}} \right| dx + \frac{1}{s}\int_\Omega |f-v|^s \, dx \tag{5.3}$$

and set $\varepsilon_i = Ch^{1+\frac{1}{1+|\nabla u_i|^2}}$. We believe that $M = 2$ should be enough.

Remark 7. In our approach, $\Omega = (0, W) \times (0, H)$ is normalized such that $\max(W, H) = 1$. This means that the spatial discretization step h depends on the pixel dimensions of the image. However, this issue was treated very differently in [51] and [52]. In these papers, the spatial discretization step h was chosen first and, thus, in contrast to our approach, the dimensions of Ω depend on the pixel dimensions of the image.

As pointed out in [51] and [52], the spatial discretization step h plays an important role in the behavior of the method. This is due to the fact that h affects the magnitude of the gradient. In order to justify the choices made in this paper we investigated the behavior of the discrete regularization term

$$k(v)(P_j) = \left| \left(\mathrm{div}_h \frac{\nabla v}{\sqrt{1+|\nabla v|^2}} \right)(P_j) \right|, \; v \in V_h, P_j \in \Sigma_{0h}. \tag{5.4}$$

Our goal was to find out how $k(v)$ behaves pointwise as a function of h. This refines the view on how h should be chosen. We generated a large number of test images containing only random noise and analyzed the obtained data statistically. In addition, we modified our implementation so that the parameter h we consider in the remainder of the current remark is consistent with the one used in [51], [52], implying that h may be chosen freely, and tested various values of this parameter (including some larger than 1).

Figures 2, 3, and 4 show the obtained results. In Figure 2, the pixel values are uniformly distributed over various intervals; i.e., the figure shows how the regularization term reacts to the changes in the "intensity" of the noise. In Figure 3, the nonzero pixel values are uniformly distributed in the interval $[-0.2, 0.2]$, but only a certain percentage of the pixels contains nonzero values; i.e., the figure shows how the regularization term reacts to the changes in the "quantity" of the noise. Figure 4 shows the same results in a scaled form.

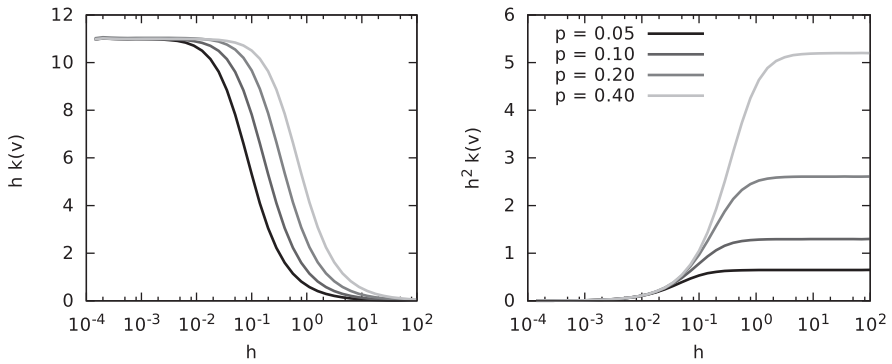

Figure 2. *The average pointwise behavior of the discrete regularization term $k(v)$ as a function of h. The pixel values are uniformly distributed in the interval $[-p, p]$.*

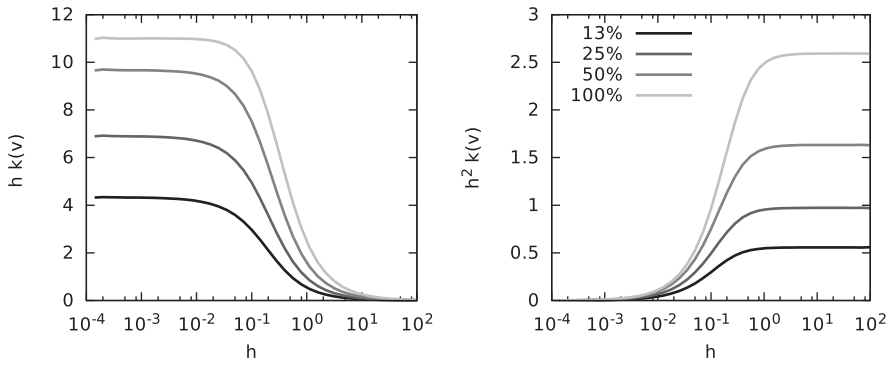

Figure 3. *The average pointwise behavior of the discrete regularization term $k(v)$ as a function of h. Only a certain percentage of the pixels contains nonzero values.*

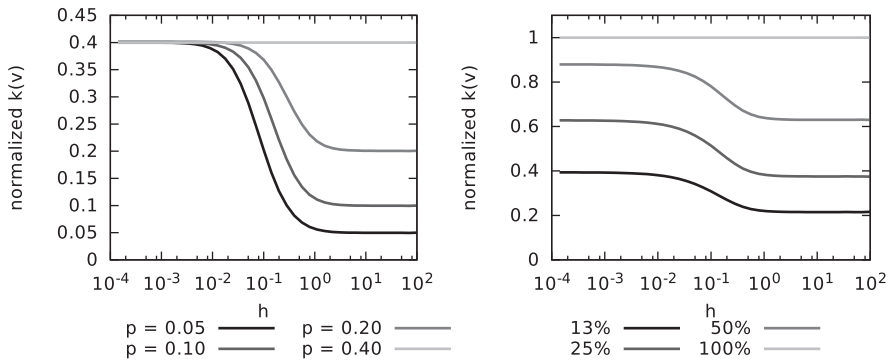

Figure 4. *Scaled versions of Figures 2 (on the left) and 3 (on the right).*

It is clear that $k(v)$ exhibits three different behaviors when the value of h is varied:

1. If h is small enough ($\sim 10^{-3}$), then $k(v) \sim h^{-1}$, which is consistent with the aforementioned dimensional analysis. Another defining property is that the regularization term $k(v)$ is unable to differentiate between low intensity and high intensity noise.

This has three main consequences: First, the regularization term is almost always very large when dealing with photographic images; thus the model wants to produce overly smooth solutions. As a result, the parameter ε must be small to preserve small details. Second, it is likely that the model may misidentify some noise as a true jump in the data and preserve it. This observation was also noted in [51] and [52]. Third, the regularization term probably becomes even more difficult to deal with because the transition from a smooth image to a noisy image is extremely steep. This could also explain why the initialization (4.2) did not work: If $u^0 = f$, then the value of the regularization term would be the same almost everywhere at u^0, and a transition to even slightly lower "energy" would require significant smoothing of the image. In other words, it is nearly impossible to find a descending path from u^0 to the global minimizer because the regularization term is "flat" near u^0 $(= f)$. It should be noted that the regularization term can still differentiate between low and high quantity of noise.

2. If h is large enough $(\sim 10^1)$, then $k(v) \sim h^{-2}$, which is again consistent with the aforementioned dimensional analysis. The results show that the value of the regularization term is directly proportional to the intensity of the noise (see, in particular, Figure 4). This is not particularly surprising because $k(v) \sim \triangle v$ when $|\nabla v|^2 \ll 1$. Numerical experiments indicate that our method works really well when $h = 1$, but the output images are quite blurred, as expected, since the regularization term favors shallow gradients over the steeper ones. In addition, the initialization (4.2) actually works even better than the initialization (4.3) in the sense that the method converges much faster. Both initializations also lead to the same solution in most cases. Again, this is not a surprise because the nonconvex part of the regularization term does not have a major impact when $|\nabla v|^2 \ll 1$.
3. If $h \sim 10^{-1}$, then the regularization term does not have a clear asymptotic behavior. This parameter range is clearly the most interesting because of the way the regularization term reacts to the changes in the intensity of the noise. The value of h determines how steep the transition from a smooth image to a noisy image is, and, thus, it has a significant effect on the final output image. Unfortunately, it is far from clear how h should be chosen. In principle, the parameter h determines how the regularization term reacts to noise, and the parameter ε determines how strongly this reaction is taken into account. However, in practice, there seems to be some overlap between these two parameters.

Our choice of h falls into the first category if it is assumed that the number of pixels is large. Although this choice has many disadvantages, the effects of which can be seen in some of the numerical results presented in this paper, we believe that our choice is justified, at least in the context of this paper, for the following reasons: (i) If the input image contains substantial amount of noise, then only a limited amount of information can be recovered even under the best conditions. By taking this into consideration, a solution that is a little too smooth is not a big disadvantage. In addition, if the original image is smooth, then small h is a reasonable choice. (ii) If $h \ll 1$, then the two terms in the objective function can be easily balanced by selecting $\varepsilon \sim h$. This means that tuning the parameters is going to be a much easier task. Considering that the goals of this paper are purely algorithmic, we do not

want to focus too much on such tuning. (iii) Since the objective function is likely to be more challenging to deal with when h is chosen to be small, problems with small h can be seen as benchmark problems. In that sense, small h is is well suited for the goals of this paper.

5.2. Examples and comparisons. Figure 5 shows the results obtained while applying the implementation against one of the synthetic test images (Test9). This synthetic test image contains only simple patterns and shapes. The purpose of this test image is to show that the algorithm works effectively in the sense that edges, corners, and image contrast are preserved. Here the noise parameter p was 0.2, and the parameter r_0 was 0.015. The value of the objective function in (1.5) at noisy image f was 0.027159 and $\mathcal{J}(u^0) = 0.364139$. The convergence to the given tolerance was achieved after 291 iterations with $\mathcal{J}(u^{291}) = 0.007054$. The l^2-residual between the vector representations of the original and noisy images was 58.95. A similar residual between the original and output images was 9.049. The algorithm was able to eliminate practically all noise, and the output image is almost indistinguishable from the original image. However, when the difference between the output and noisy images is examined more closely, it is clear that the algorithm had some minor difficulties with the diagonal tips of the star.

Figure 6 shows similar results for a second synthetic test image (Test6). This synthetic test image contains many different challenges (sine waves, high and low contrast text, gradients, edges, corners, and narrow bounces) to the algorithm. Here $p = 0.2$, $r_0 = 0.004$, $\mathcal{J}(f) = 0.004503$, and $\mathcal{J}(u^0) = 0.207021$. The convergence to the given tolerance was achieved after 215 iterations with $\mathcal{J}(u^{215}) = 0.004130$. The residuals between the noisy and original and the output and original images were 58.16 and 14.35, respectively. The algorithm filtered out a considerable amount of noise, and it is very clear that the algorithm does not have problems with sine waves or gradients. However, the algorithm had difficulties in preserving certain details, such as low-contrast text and the narrowest bounces in the surface.

Figure 7 shows results for a photographic image (Barbara). This test image is particularly challenging because it contains image details and noise on similar scales. Here, $p = 0.2$, $r_0 = 0.005$, $\mathcal{J}(f) = 0.008823$, and $\mathcal{J}(u^0) = 0.134979$. After 268 iterations, the value of the objective function in (1.5) was 0.008394. The residuals between the noisy and original and the output and original images were 58.86 and 35.50, respectively. Again, the algorithm filtered out a considerable amount of noise, but some details were lost in the process as it is clear that the algorithm was unable to resolve the stripes and grids in the table cloth and pants. This can be seen clearly in the intersection plot.

The values of the objective function along the iterations are plotted in Figures 8, 9, and 10. The figures also include the following normalized residuals:

$$
\begin{aligned}
R_1^n &= \frac{1}{2}\frac{|\nabla u^n - \mathbf{p}_1^n|}{\max(|\nabla u^n|, |\mathbf{p}_1^n|)},\\
R_2^n &= \frac{1}{2}\frac{|\mathbf{p}_2^n - \mathbf{p}_3^n|}{\max(|\mathbf{p}_2^n|, |\mathbf{p}_3^n|)},\\
R_3^n &= \frac{1}{2}\frac{|\mathrm{div}_h \mathbf{p}_3^n - \psi^n|}{\max(|\mathrm{div}_h \mathbf{p}_3^n|, |\psi^n|)}.
\end{aligned}
\tag{5.5}
$$

In all three cases, the value of the objective function drops sharply during the first few tens of

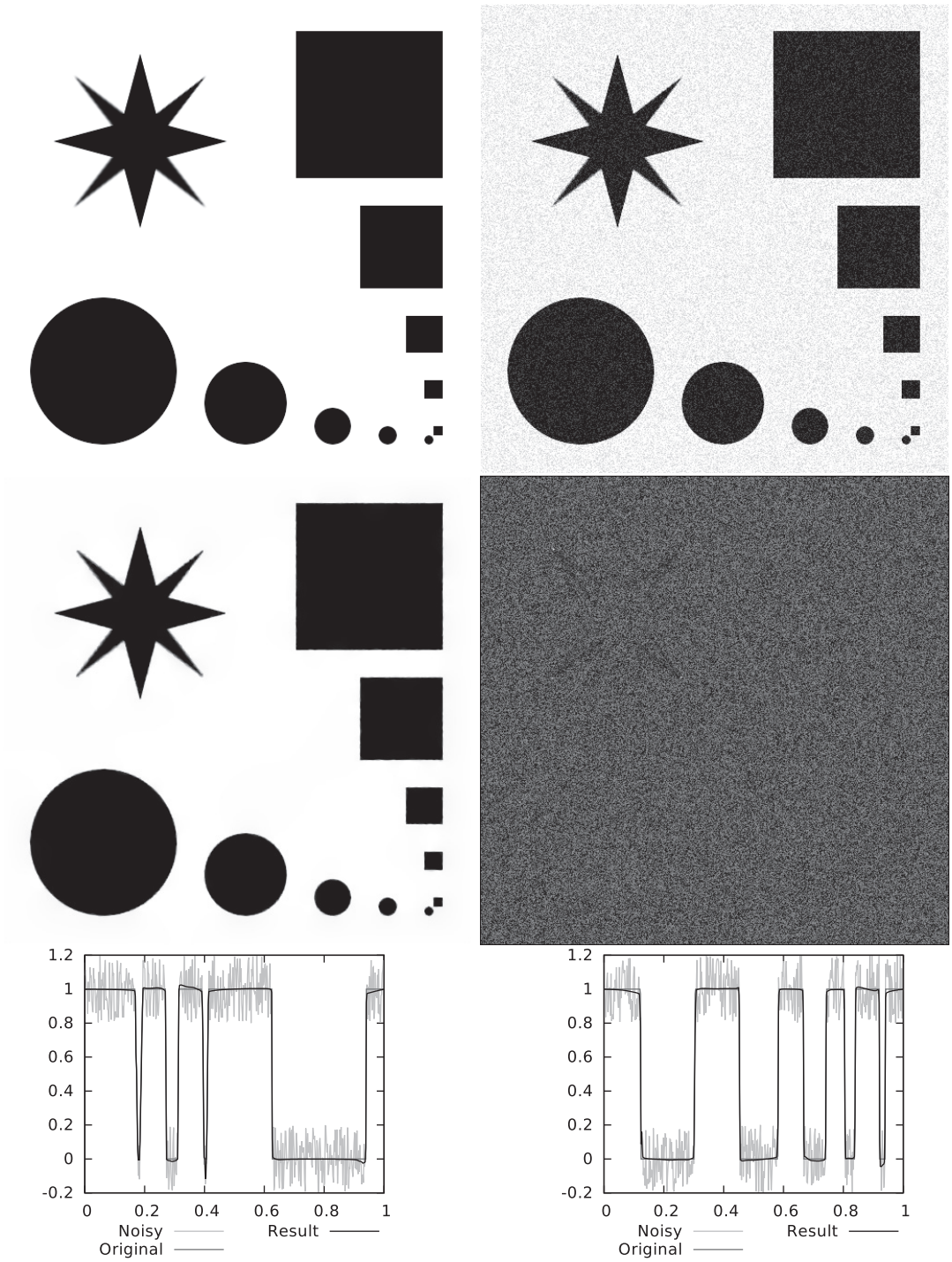

Figure 5. *A synthetic test image (Test9) containing simple shapes. From top left: Original image, noisy image ($p = 0.2$), output image, difference between the output and noisy images, and two horizontal intersections.*

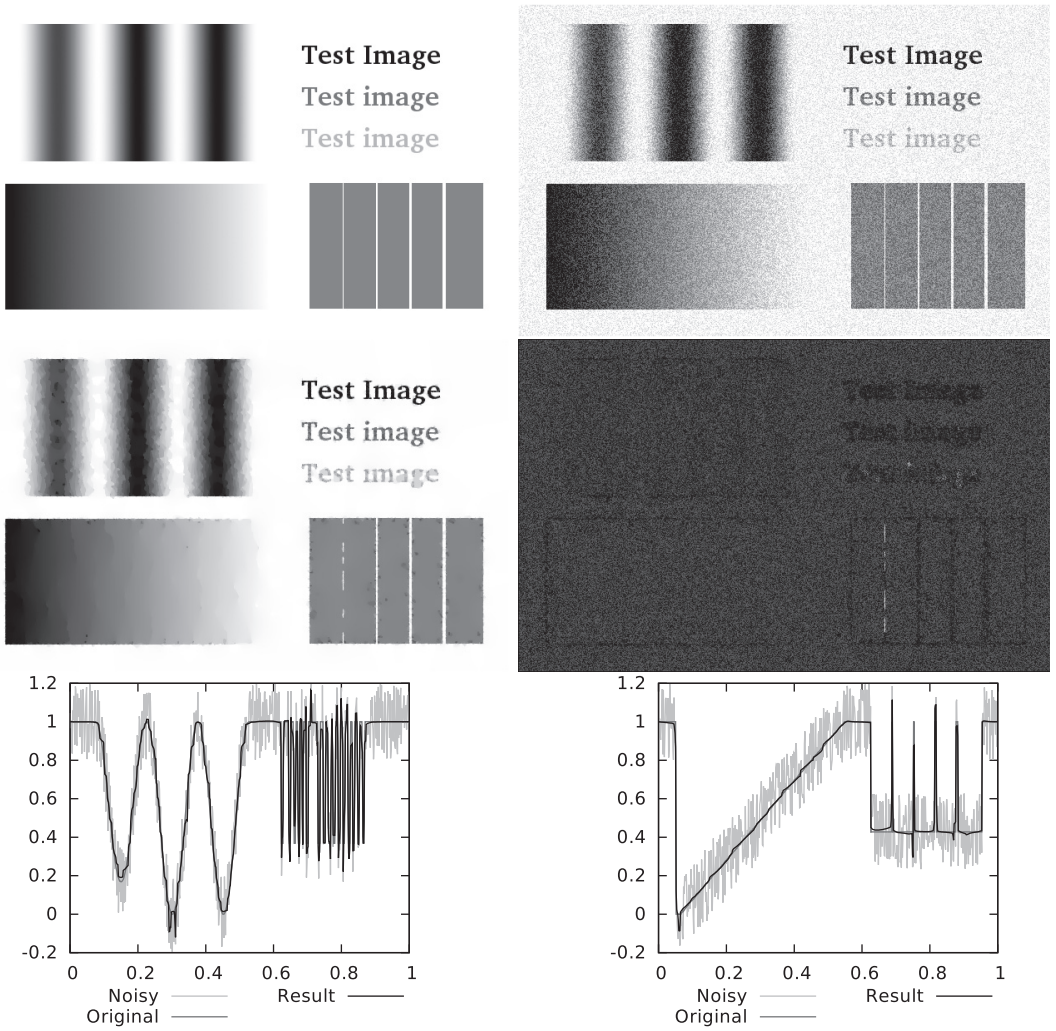

Figure 6. *A synthetic test image (Test6) containing sine waves, text, gradients, and narrow bounces. From top left: Original image, noisy image ($p = 0.2$), output image, difference between the output and noisy images, and two horizontal intersections.*

iterations. However, in the case of the last two, it takes a few tens of iterations more before the value of the objective function drops below $\mathcal{J}(f)$. Also, the decrease is not monotonic as the value of the objective function jumps momentarily after a few iterations. This jump takes place at the same time as the value of the normalized residual R_1^n jumps and was observed with almost all the test images in varying extent. Similar behavior was also presented in [52].

Figure 11 shows a comparison between different test problems with varying amounts of noise. The relevant parameters, objective function values, iteration counts, and residuals are shown in Table 1. It is clear that the algorithm performed commendably when $p = 0.05$ or $p = 0.1$. In the case of $p = 0.2$, more details were lost in the process, and when $p = 0.4$, almost all small details were lost.

Finally, Figure 12 visualizes the influence of r_0 on the resulting denoised image. The

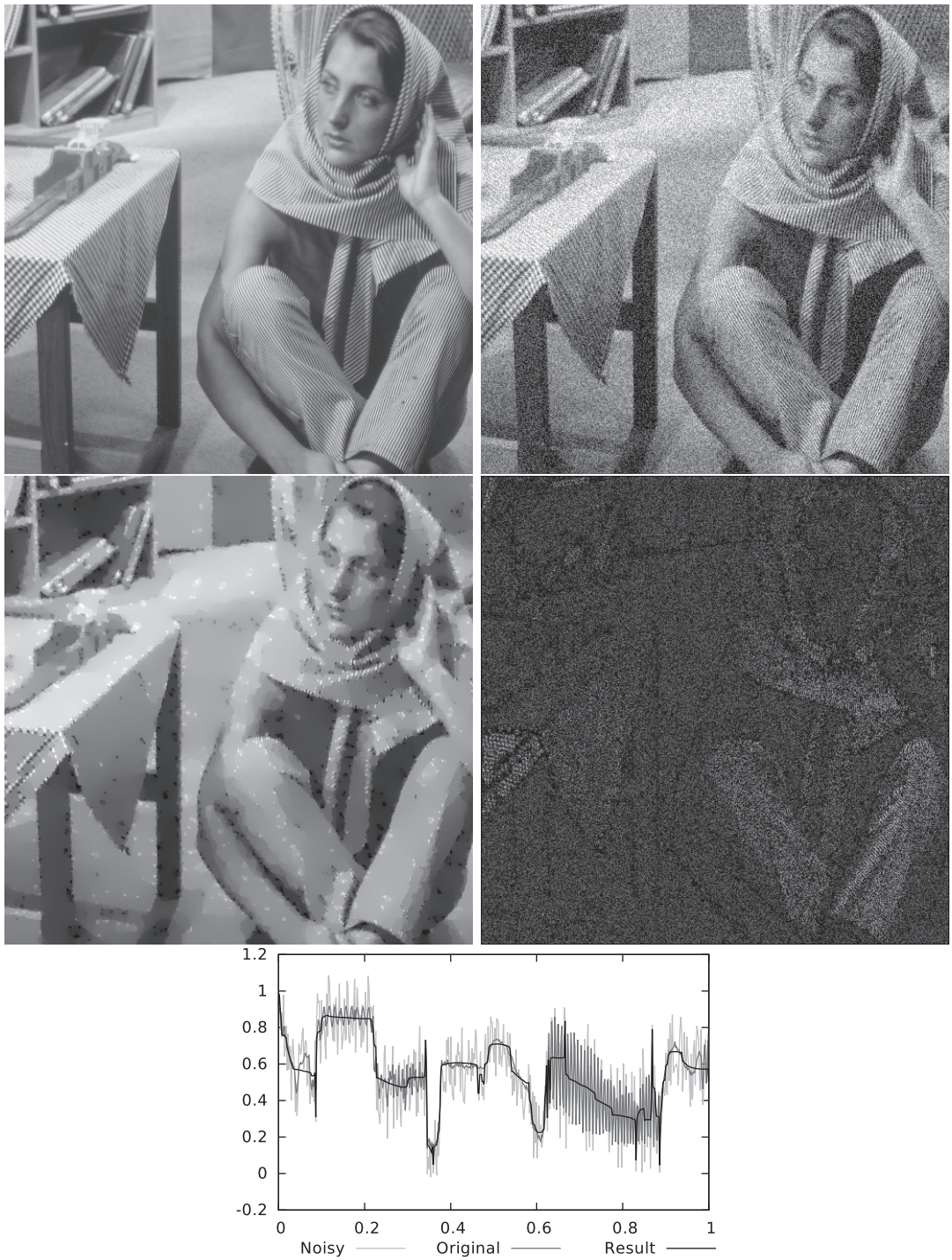

Figure 7. *A photographic test image (Barbara). From top left: Original image, noisy image ($p = 0.2$), output image, difference between the output and noisy images, and horizontal intersection from the center of the image.*

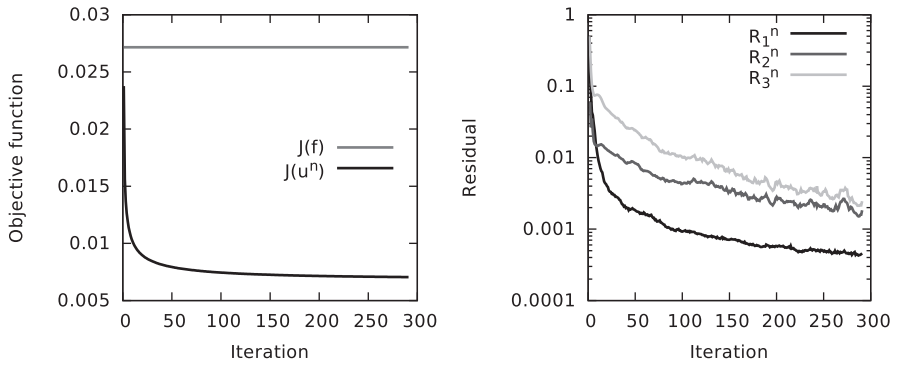

Figure 8. *Values of the objective function in* (1.5) *and normalized residuals* (5.5) *along the iterations for a synthetic test image (Test*9*).*

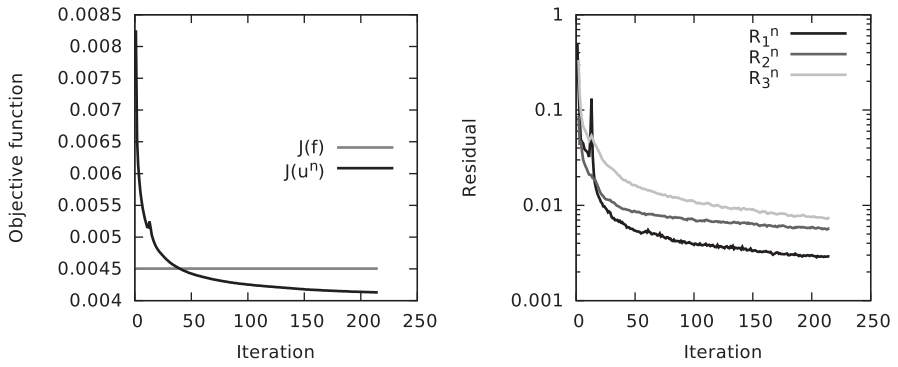

Figure 9. *Values of the objective function in* (1.5) *and normalized residuals* (5.5) *along the iterations for a synthetic test image (Test*6*).*

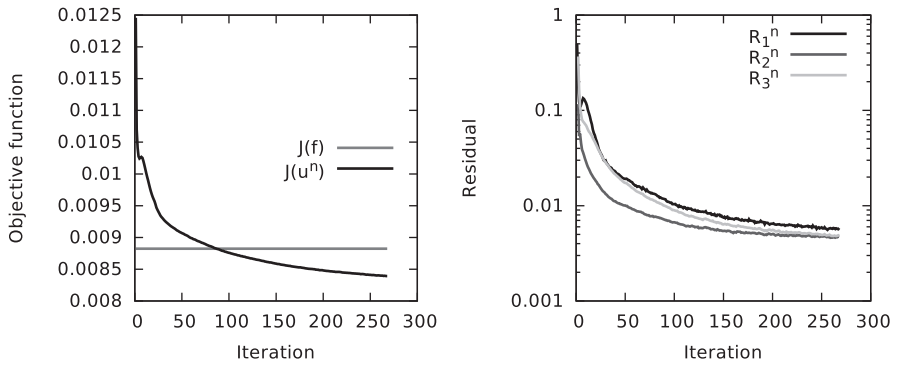

Figure 10. *Values of the objective function in* (1.5) *and normalized residuals* (5.5) *along the iterations for the photographic test image (Barbara).*

relevant objective function values, iteration counts, and residuals are shown in Table 2. In all cases, $p = 0.2$. It is clear that the optimal value of r_0 is somewhere near 0.005. These results illustrate a clear trend that can be observed in all considered test images: the larger the value of r_0 is, the better the algorithm behaves. Actually, when $r_0 = 0.001$, the value of

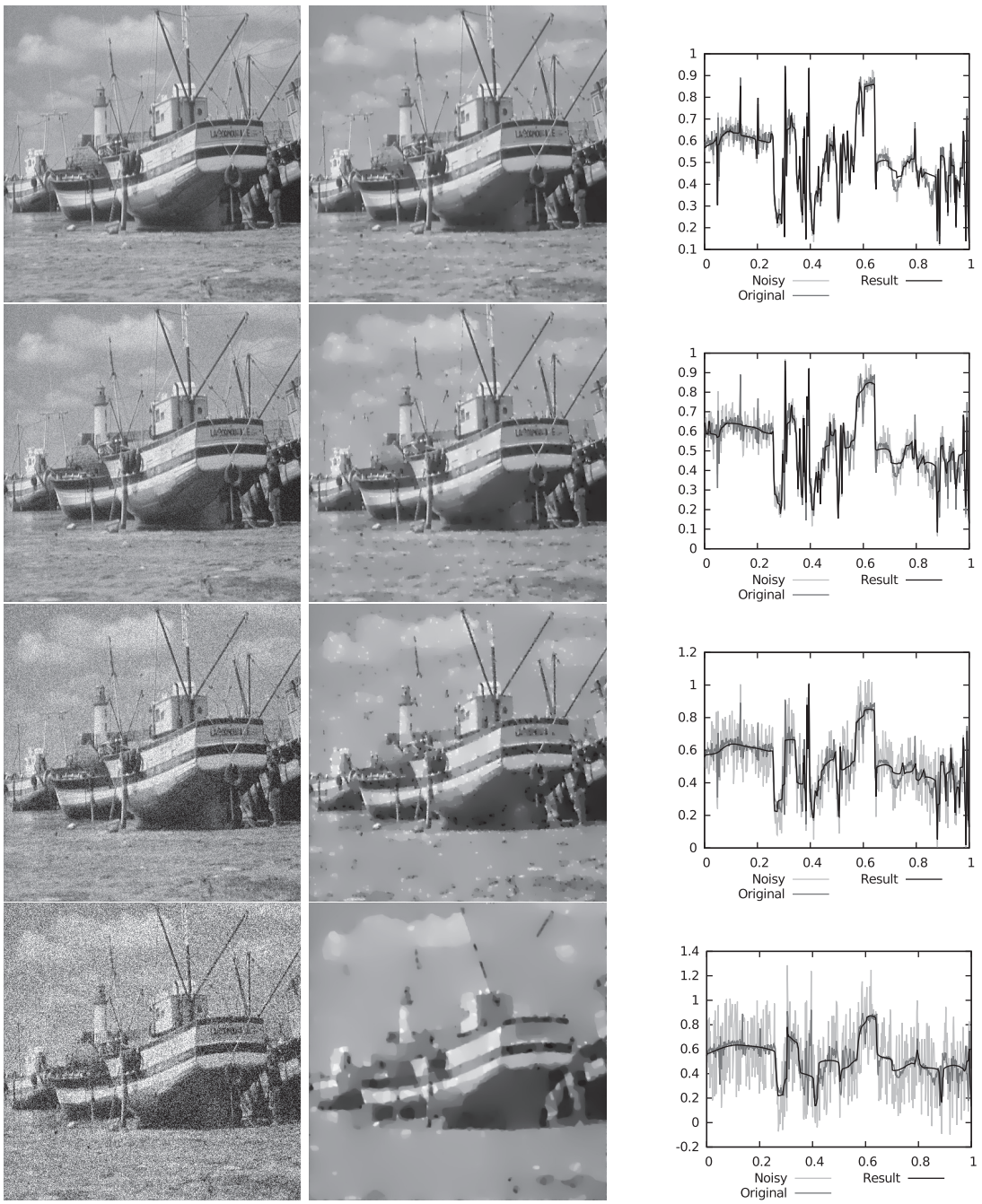

Figure 11. *A comparison between different test problems with varying amounts of noise:* $p = 0.05$ *(*$r_0 = 0.001$*),* $p = 0.1$ *(*$r_0 = 0.002$*),* $p = 0.2$ *(*$r_0 = 0.005$*), and* $p = 0.4$ *(*$r_0 = 0.02$*). From left to right: The noisy image, the output image, and a horizontal intersection from the center of the image.*

the objective function at the achieved solution u^n is higher than it is at the noisy image f, although the algorithm appears to be working properly otherwise. On the other hand, a large value of r_0 means that more detail is lost in the process. Thus, selecting an optimal value for r_0 is a difficult balancing act. Fortunately, it appears that the optimal value of the parameter r_0 mainly depends on the amount of noise and does not depend as strongly on the image itself. The previous sentence holds true at least when success is measured by the l^2-norm. If success is estimated by visual inspection, then it appears that r_0 must be selected image dependently, because even a small change in the value of the parameter r_0 can have a visible impact on the final result. This is consistent with comments in [52]. However, the values provided in Table 2 work fairly well for most images.

All presented numerical results show that the L^1-mean curvature allows both smooth transitions and large jumps without staircasing. In two dimensions the output results do not appear perfect, but the one-dimensional intersections demonstrate desired overall behavior; the qualitative challenges and difficulties are mostly related to the mixtures of image details and noise on similar scales.

Table 1

A comparison between different test problems with varying amounts of noise, with n denoting the number of iterations necessary to achieve convergence and u^n the achieved solution. The Residuals column shows the residuals between the vector presentations of the original and noisy images and the original and output images.

p	r_0	$\mathcal{J}(f)$	$\mathcal{J}(u^0)$	$\mathcal{J}(u^n)$	Iterations	Residuals
0.05	0.001	0.001626	0.145999	0.000831	150	14.72, 13.10
0.1	0.002	0.003435	0.147198	0.002228	173	29.48, 18.53
0.2	0.005	0.008909	0.152132	0.007485	255	58.99, 26.49
0.4	0.020	0.036206	0.172260	0.029445	181	117.8, 37.31

Table 2

A comparison between different values of r_0. The Residuals column shows the residuals between the vector presentations of the original and noisy images and the original and output images. The function u^n is the achieved solution.

r_0	$\mathcal{J}(f)$	$\mathcal{J}(u^0)$	$\mathcal{J}(u^n)$	Iterations	Residuals
0.001	0.001795	0.142086	0.004210	193	58.93, 39.59
0.005	0.008979	0.142142	0.007126	220	58.92, 21.13
0.01	0.017945	0.142136	0.007911	180	58.85, 21.91
0.05	0.089864	0.141968	0.010577	242	58.87, 33.91

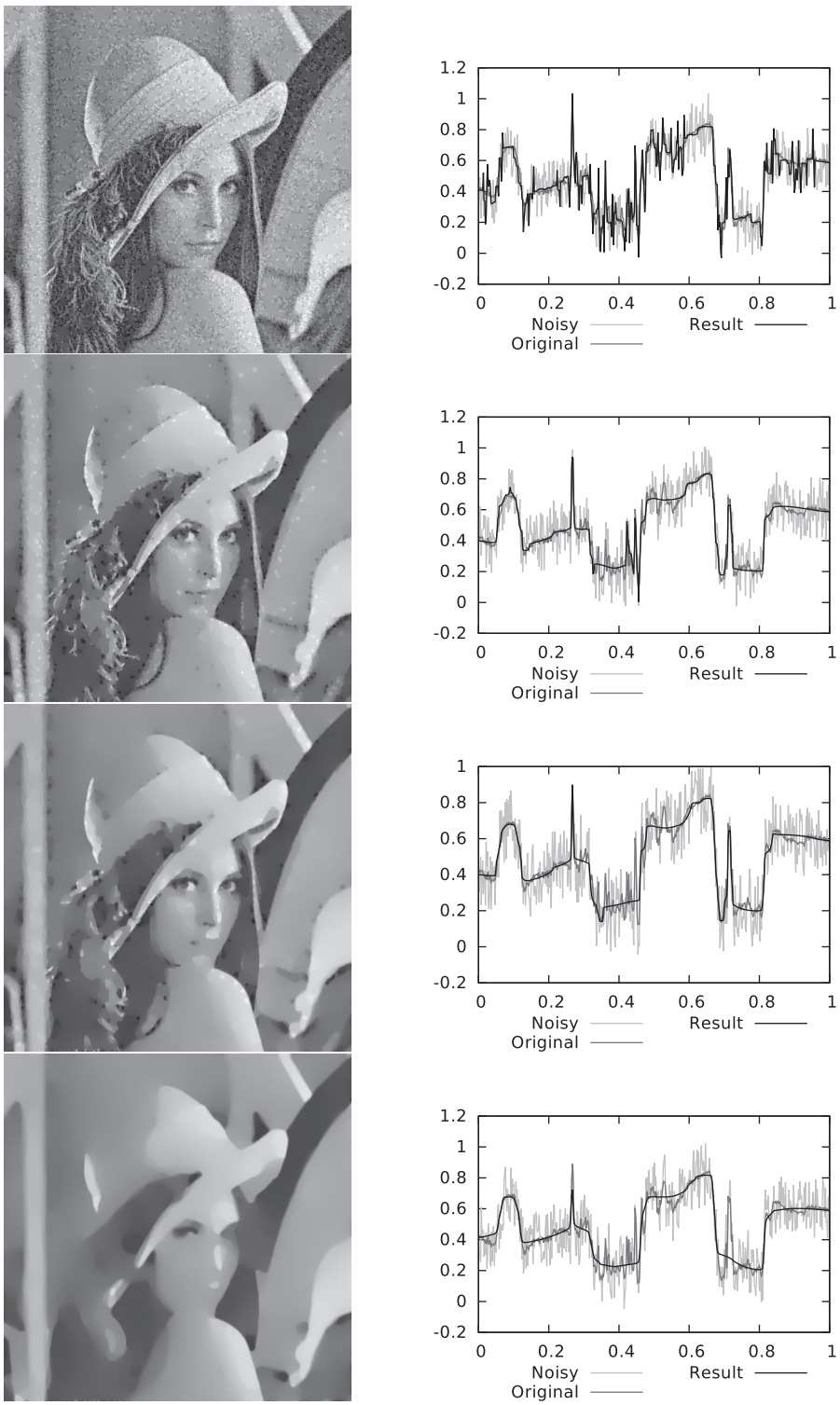

Figure 12. *A comparison between different values of* r_0*:* $0.001, 0.005, 0.01$*, and* 0.05*. The left side shows the output images, and the right side shows the horizontal intersections from the center of the image.*

6. Conclusions. This paper presents an image denoising algorithm based on an augmented Lagrangian approach that uses the L^1-mean curvature of the image surface as a regularizer. The main difference between this paper and existing literature (e.g., [52]) is that our methodology relies on a novel augmented Lagrangian functional where the equality constraints treated by augmentation-duality are all linear, resulting in different (and simpler) subproblems. The functionality of the proposed algorithm was demonstrated by applying it against a large set of different types of test problems, some of which were presented in further detail. Based on the numerical experiments, it can be concluded that the algorithm can remove considerable amounts of noise within a reasonable number of iterations. The cpu time used by our implementation is dominated by the solution of the first subproblem; thus we feel that the effort of improving our method should be directed toward this subproblem.

Acknowledgment. The authors thank anonymous reviewers for their valuable feedback. The presentation of the paper was significantly improved thanks to their comments and suggestions.

REFERENCES

[1] L. Ambrosio and S. Masnou, *A direct variational approach to a problem arising in image reconstruction*, Interfaces Free Bound., 5 (2003), pp. 63–81.
[2] L. Ambrosio and S. Masnou, *On a variational problem arising in image reconstruction*, in Free Boundary Problems, Internat. Ser. Numer. Math. 147, Birkhäuser, Basel, 2004, pp. 17–26.
[3] G. Bellettini, V. Caselles, and M. Novaga, *The total variation flow in* $\mathbb{R}^N$, J. Differential Equations, 184 (2002), pp. 475–525.
[4] J. D. Benamou and Y. Brenier, *A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem*, Numer. Math., 84 (2000), pp. 375–393.
[5] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.
[6] D. Boffi, M. Fortin, and F. Brezzi, *Mixed Finite Element Methods and Applications*, Springer Series in Computational Mathematics, Springer, Berlin, Heidelberg, 2013.
[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn., 3 (2011), pp. 1–122.
[8] A. Caboussat, R. Glowinski, and V. Pons, *An augmented Lagrangian approach to the numerical solution of a non-smooth eigenvalue problem*, J. Numer. Math., 17 (2009), pp. 3–26.
[9] A. Caboussat, T. Glowinski, and D.C. Sorensen, *A least-squares method for the numerical solution of the Dirichlet problem for the Monge-Ampére equation in two dimensions*, ESAIM Control Optim. Calc. Var., 19 (2013), pp. 780–810.
[10] A. Chambolle and P. L. Lions, *Image recovery via total variational minimization and related problems*, Numer. Math., 76 (1997), pp. 167–188.
[11] R. Chan, M. Tao, and X. Yuan, *Constrained total variation deblurring models and fast algorithms based on alternating direction method of multipliers*, SIAM J. Imaging Sci., 6 (2013), pp. 680–697.
[12] T. Chan, S. Esedoglu, F. Park, and A. Yip, *Total variation image restoration: Overview and recent developments*, in Handbook of Mathematical Models in Computer Vision, Ni. Paragios, Y. Chen, and O. Faugeras, eds., Springer, New York, 2006, pp. 17–31.
[13] T. F. Chan, S. H. Kang, and J. Shen, *Euler's elastica and curvature-based inpainting*, SIAM J. Appl. Math., 63 (2002), pp. 564–592.
[14] T. F. Chan and J. J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*, SIAM, Philadelphia, 2005.

[15] Y. Chen, W. Hager, F. Huang, D. Phan, X. Ye, and W. Yin, *Fast algorithms for image reconstruction with application to partially parallel MR imaging*, SIAM J. Imaging Sci., 5 (2012), pp. 90–118.
[16] E. J. Dean and R. Glowinski, *Numerical methods for fully nonlinear elliptic equation of the Monge-Ampére type*, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 1344–1386.
[17] E. J. Dean, R. Glowinski, and G. Guidoboni, *On the numerical simulation of Bingham visco-plastic flow: Old and new results*, J. Non-Newtonian Fluid Mech., 142 (2007), pp. 36–62.
[18] E. J. Dean and R. Glowinski, *An augmented Lagrangian approach to the numerical solution of the Dirichlet problem for the elliptic Monge-Ampére equation in two dimensions*, Electron. Trans. Numer. Anal., 22 (2006), pp. 71–96.
[19] F. Delbos, J. Ch. Gilbert, R. Glowinski, and D. Sinoquet, *Constrained optimization of seismic reflection tomography: A Gauss-Newton augmented Lagrangian approach*, Geophys. J. Int., 164 (2006), pp. 670–684.
[20] Y. Duan, Y. Wang, X.-C. Tai, and J. Hahn, *A fast augmented Lagrangian method for Euler's elastica model*, in Scale Space and Variational Methods in Computer Vision, Lecture Notes in Comput. Sci. 6667, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 144–156.
[21] X. Feng, R. Glowinski, and M. Neilan, *Recent developments in numerical methods for fully nonlinear second order partial differential equations*, SIAM Rev., 55 (2013), pp. 205–267.
[22] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary Value Problems*, North–Holland, Amsterdam, 1983.
[23] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*, Comput. Math. Appl., 2 (1976), pp. 17–40.
[24] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer, New York, 1984.
[25] R. Glowinski, *Finite Element Methods for Imcompressible Viscous Flow,*. in Handbook of Numerical Analysis, vol. 9, North-Holland, Amsterdam, 2003, pp. 3–1176.
[26] R. Glowinski, T. Kärkkäinen, T. Valkonen, and A. Ivannikov, *Nonsmooth SOR for L^1-fitting: Convergence study and discussion of related issues*, J. Sci. Comput., 37 (2008), pp. 103–138.
[27] R. Glowinski and A. Marrocco, *Sur l'approximation par éléments finis d'ordre un, et la résolution par pénalisation-dualité d'une classe de problemes de Dirichlet non linéaires*, RAIRO Anal. Numér., 9 (1975), pp. 41–76.
[28] R. Glowinski and A. Quaini, *On an inequality of C. Sundberg: A computational investigation via nonlinear programming*, J. Optim. Theory Appl., 158 (2013), pp. 739–772.
[29] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, SIAM, Philadelphia, 1989.
[30] J. Hahn, C. Wu, and X.-C. Tai, *Augmented Lagrangian method for generalized TV-Stokes model*, J. Sci. Comput., 50 (2012), pp. 235–264.
[31] M. Hintermüller, C. N. Rautenberg, and J. Hahn, *Functional-analytic and numerical issues in splitting methods for total variation-based image reconstruction*, Inverse Problems, 30 (2014), 055014.
[32] K. Ito and K. Kunisch, *Lagrange Multiplier Approach to Variational Problems and Applications*, SIAM, Philadelphia, 2008.
[33] T. Kärkkäinen and K. Majava, *Nonmonotone and monotone active-set methods for image restoration, part* 1: *Convergence analysis*, J. Optim. Theory Appl., 106 (2000), pp. 61–80.
[34] T. Kärkkäinen and K. Majava, *Nonmonotone and monotone active-set methods for image restoration, part* 2: *Numerical results*, J. Optim. Theory Appl., 106 (2000), pp. 81–105.
[35] T. Kärkkäinen, K. Majava, and M. M. Mäkelä, *Comparison of formulations and solution methods for image restoration problems*, Inverse Problems, 17 (2001), pp. 1977–1995.
[36] Y. A. Kuznetsov, *Numerical methods in subspaces*, Vychislitel'-nye Processy i Sistemy II, 37 (1985), pp. 265–350.
[37] Yu. A. Kuznetsov and T. Rossi, *Fast direct method for solving algebraic systems with separable symmetric band matrices*, East-West J. Numer. Math., 4 (1996), pp. 53–68.
[38] Y. Meyer, *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations: The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*, AMS, Providence, RI, 2001.
[39] D. Mumford, *Elastica and computer vision*, in Algebraic Geometry and Its Applications, Springer-Verlag, New York, 1994, pp. 491–506.

[40] M. Ng, P. Weiss, and X. Yuan, *Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods*, SIAM J. Sci. Comput., 32 (2010), pp. 2710–2736.
[41] P. Perona and J. Malik, *Scale space and edge detection using anisotropic diffusion*, in Proceedings of IEEE Workshop on Computer Vision, IEEE, Washington, DC, 1987, pp. 16–27.
[42] P. Perona and J. Malik, *Scale-space and edge detection using anisotropic diffusion*, IEEE Trans. Pattern Anal. Mach. Intell., 12 (1990), pp. 629–639.
[43] T. Rossi and J. Toivanen, *A parallel fast direct solver for block tridiagonal systems with separable matrices of arbitrary dimension*, SIAM J. Sci. Comput., 20 (1999), pp. 1778–1796.
[44] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
[45] Y. Shi, L.-L. Wang, and X.-C. Tai, *Geometry of total variation regularized l^p-model*, J. Comput. Appl. Math., 236 (2012), pp. 2223–2234.
[46] X.-C. Tai, J. Hahn, and G. J. Chung, *A fast algorithm for Euler's elastica model using augmented Lagrangian method*, SIAM J. Imaging Sci., 4 (2011), pp. 313–344.
[47] P. Vassilevski, *Fast algorithm for solving a linear algebraic problem with separable variables*, Comptes Rendus de Academie Bulgare des Sciences, 37 (1984), pp. 305–308.
[48] Y. Wang, J. Yang, W. Yin, and Y. Zhang, *A new alternating minimization algorithm for total variation image reconstruction*, SIAM J. Imaging Sci., 1 (2008), pp. 248–272.
[49] J. Weickert, *Anisotropic Diffusion in Image Processing*, B.G. Teubner, Stuttgart, Germany, 1998.
[50] C. Wu and X.-C. Tai, *Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models*, SIAM J. Imaging Sci., 3 (2010), pp. 300–339.
[51] W. Zhu and T. Chan, *Image denoising using mean curvature of image surface*, SIAM J. Imaging Sci., 5 (2012), pp. 1–32.
[52] W. Zhu, X.-C. Tai, and T. Chan, *Augmented Lagrangian method for a mean curvature based image denoising model*, Inverse Probl. Imaging, 7 (2013), pp. 1409–1432.