Minimization of a Detail-preserving Regularization Functional for Impulse Noise Removal*

Jian-Feng Cai [†]   Raymond H. Chan[‡]   Carmine Di Fiore[§]

Abstract

Recently, a powerful two-phase method for restoring images corrupted with high level impulse noise has been developed. The main drawback of the method is the computational efficiency of the second phase which requires the minimization of a non-smooth objective functional. However, it was pointed out in [Chan, Ho, Leung, and Nikolova, *Proc. ICIP 2005*, pp. 125–128] that the non-smooth data-fitting term in the functional can be deleted since the restoration in the second phase is applied to noisy pixels only. In this paper, we study the analytic properties of the resulting new functional $\mathcal{F}$. We show that $\mathcal{F}$, which is defined in terms of edge-preserving potential functions φ_α, inherits many nice properties from φ_α, including the first and second order Lipschitz continuity, strong convexity, and positive definiteness of its Hessian. Moreover, we use these results to establish the convergence of optimization methods applied to $\mathcal{F}$. In particular, we prove the global convergence of some conjugate gradient-type methods and of a recently proposed low complexity quasi-Newton algorithm. Numerical experiments are given to illustrate the convergence and efficiency of the two methods.

Keywords: Image Processing, Variational Method, Optimization.

AMS subject classification: 68U10, 65K10.

1 Introduction

Image denoising is a typical inverse problem and is hard to solve. We must use some a priori knowledge about the images and noises in the denoising procedure. For the images, the usual assumption is that they consist of edges and homogeneous regions, which yields the popular edge-preserving regularization functionals [2, 4, 11, 17]. For the noises, there are several models. The most commonly used one is the Gaussian noise model, in which every pixel is affected by the noise. Another one is the impulse noise model, in which only a portion of the pixels is contaminated by the noise and information on the true value of the pixels are completely lost. The impulse noise is caused by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or transmission in a noisy channel, see [5, p. 90 and p. 330] for instance. The impulse noise can be further categorized in two types: the salt-and-pepper noise, for which the noisy pixels can take only the maximal and

1

minimal pixel values; and the random-valued impulse noise, for which the noisy pixels can take any random values between the maximal and the minimal pixel values. In both types of noise, the noisy pixels are assumed to be randomly distributed in the image.

There are many methods for restoring images corrupted by impulse noise, some of which are based on nonlinear digital filters [1]. The median filter was once the most popular nonlinear filter for removing impulse noise, because of its good denoising power [5, pp. 90–94] and computational efficiency [18]. However, its performance is not good when the noise level is high since some details and edges of the original image appear smeared in the restored one [26]. Several remedies have been proposed, such as the adaptive [19], the multi-state [12] and homogeneity information based [27] median filters. They first locate possible noisy candidates and then replace them with median values or their variants. They can detect the noisy pixels even at a high noise level. However, they can not restore such pixels satisfactorily because they do not take into account local image features such as the possible presence of edges. Hence details and edges are not recovered well, especially when the noise level is high.

Recently, a different technique based on a variational framework has been proposed in [24] to deal with impulse noise. Variational methods based on the minimization of edge-preserving regularization functionals have already been used in [4, 11, 29] to preserve the edges and the details in images corrupted by Gaussian noise. However, these methods fail in the presence of impulse noise, because they alter all pixels in the image, including uncorrupted ones. In the variational method proposed in [24], the restored image was defined as the minimizer of a cost function which includes a non-smooth ℓ_1 data-fitting term. It was shown that the method involves an implicit detection of the pixels contaminated by impulse noise and it preserves edges during denoising. However, in order to detect large noisy connected regions, it requires a greater weight of the regularization term in the cost function. This causes the distortion of some pixels near edges.

In [8], Chan et al. proposed a two-phase method which combines the variational method [24] with the adaptive median filter method (AMF) [19], thereby avoiding the drawbacks of these methods. More precisely, the noise candidates are first identified by AMF; then these noise candidates are restored by minimizing an objective functional $\mathcal{G}$ with an ℓ_1 data-fitting term and a regularization term involving an edge-preserving potential function $\varphi_\alpha(t)$, see (2). Since the edges are preserved for the noise candidates and other pixels are left unchanged, the two-phase scheme out-performs previous methods by a great margin: salt-and-pepper noise with noise ratio as high as 90% can be cleaned quite efficiently, see [8] or Figures 3 and 4 in Section 5. With slight modification, the scheme also applies to random-valued impulse noise with noise level as high as 60% [10, 14]. Recently, it has been applied to problems in segmentation and inpainting, see [21].

The main drawback of the 2-phase method is its computational efficiency. The minimization of $\mathcal{G}$ is difficult as it contains a nonsmooth ℓ_1 data-fitting term. The relaxation method proposed in [8, 10, 24] is convergent but slow. Using probability arguments, we will show here that the relaxation method converges fast only if the noise level is low. To improve the computational efficiency, it was proposed in [9] to drop the non-smooth ℓ_1 data-fitting term in $\mathcal{G}$, as it is not needed in the 2-phase method, where only noisy pixels are restored in the minimization. The quality of the restored images is not affected by minimizing the new functional, see numerical results in [9, 6, 14] and the results in Section 5.

In this paper we first study the analytical properties of the new functional $\mathcal{F}$ which is obtained from $\mathcal{G}$ by dropping the ℓ_1 data-fitting term (see (9)). In particular, we prove that $\mathcal{F}$ inherits many nice and useful properties from the edge-preserving potential function φ_α, such as the first and second order Lipschitz continuity, convexity, strong convexity, and positive definiteness and uniform boundedness of the second order derivatives. Then we apply these results to show the global convergence of two methods for the minimization of $\mathcal{F}$: a conjugate gradient type method [6, 28] and a low-complexity

2

quasi-Newton method, recently introduced in [3, 15]. Finally, we show by numerical experiences that these methods can improve the efficiency of 2-phase denoising algorithms.

The outline of this paper is as follows. In Section 2, we give a brief review of the two-phase method in [8, 10], and show that the relaxation method used there is efficient only when the noise level is low. In Section 3, we prove that the new cost function $\mathcal{F}$ inherits many important regularity properties from φ_α. In Section 4, we use the results of Section 3 to establish the global convergence of two minimization methods applied to $\mathcal{F}$. In Section 5, we test the efficiency of the two methods when compared to the relaxation method in [8, 10]. Conclusions are given in the final section.

2   The Two-Phase Method

In [8], a two-phase scheme for detecting and removing salt-and-pepper noise is proposed. In the first phase, the noisy pixels are detected by the adaptive median filter (AMF) [19]. Then in the second phase, they are restored by the variational method in [24]. Let us recall the two-phase scheme here.

Let $\mathbf{x}$ be the true image with M-by-N pixels. For each $(i,j) \in \mathcal{A} := \{1, \ldots, M\} \times \{1, \ldots, N\}$, let $x_{i,j}$ be the gray level of $\mathbf{x}$ at pixel location (i,j) and $\mathcal{V}_{i,j}$ be the neighborhood of (i,j), i.e., $\mathcal{V}_{i,j} = \{(i, j-1), (i, j+1), (i-1, j), (i+1, j)\}$. Let $[s_{\min}, s_{\max}]$ be the dynamic range of $\mathbf{x}$, i.e. all $x_{i,j} \in [s_{\min}, s_{\max}]$. Denote by $\mathbf{y}$ the observed noisy image of $\mathbf{x}$ contaminated by the *salt-and-pepper noise*. Then the gray level of $\mathbf{y}$ at pixel (i,j) is

$$y_{i,j} = \begin{cases} x_{i,j}, & \text{with probability } 1 - r, \\ s_{\min}, & \text{with probability } p, \\ s_{\max}, & \text{with probability } q, \end{cases} \tag{1}$$

where $r = p + q$ is the *noise level*. The following is the two-phase scheme to get a denoised image $\mathbf{u}^*$:

1. *(Noise detection):* Denote by $\tilde{\mathbf{y}}$ the image obtained by applying AMF to the noisy image $\mathbf{y}$. Since noisy pixels either take the value $s_{\min}$ or $s_{\max}$, we define the *noise candidate set* as

$$\mathcal{N} := \{(i,j) \in \mathcal{A} : \tilde{y}_{i,j} \neq y_{i,j} \text{ and } y_{i,j} = s_{\min} \text{ or } s_{\max}\}.$$

2. *(Restoration):* If a pixel $(i,j) \notin \mathcal{N}$, it is detected as uncorrupted. Hence we naturally keep its original value, i.e., we set $u^*_{i,j} = y_{i,j}$. Otherwise, if $(i,j) \in \mathcal{N}$, then $y_{i,j}$ must be restored. In addition, if $(m,n) \in \mathcal{V}_{i,j} \setminus \mathcal{N}$, we set $u^*_{m,n} = y_{m,n}$; and if $(m,n) \in \mathcal{V}_{i,j} \cap \mathcal{N}$, then we restore $y_{m,n}$. The restoration of all $y_{i,j}$ is done by minimizing the functional:

$$\mathcal{G}(\mathbf{u}) = \sum_{(i,j) \in \mathcal{N}} \left\{ |u_{i,j} - y_{i,j}| + \frac{\beta}{2} \left[\sum_{(m,n) \in \mathcal{V}_{i,j} \setminus \mathcal{N}} 2 \cdot \varphi_\alpha(u_{i,j} - y_{m,n}) \right.\right.$$
$$\left.\left. + \sum_{(m,n) \in \mathcal{V}_{i,j} \cap \mathcal{N}} \varphi_\alpha(u_{i,j} - u_{m,n}) \right] \right\} \tag{2}$$

where β is the regularization parameter and φ_α is an even edge-preserving potential function with parameter α.

Since the right hand side of (2) involves only $u_{i,j}$, $(i,j) \in \mathcal{N}$, the vectors $\mathbf{u}$ and $\mathbf{u}^*$ are assumed to have entries $u_{i,j}$ and $u^*_{i,j}$ with (i,j) restricted to $\mathcal{N}$, i.e. $\mathbf{u}, \mathbf{u}^* \in \mathbb{R}^{|\mathcal{N}|}$. To explain the extra factor "2"

in the second summation in (2), consider a pair of neighbors (i,j) and (m,n), one of which belongs to $\mathcal{N}$, say $(i,j) \in \mathcal{N}$, while the other does not, say $(m,n) \notin \mathcal{N}$. Since φ_α is an even function and the summation is only over $(i,j) \in \mathcal{N}$, the factor "2" comes from

$$\varphi_\alpha(u_{i,j} - y_{m,n}) + \varphi_\alpha(y_{m,n} - u_{i,j}) = 2 \cdot \varphi_\alpha(u_{i,j} - y_{m,n}).$$

See also (16) in Section 3. Examples of edge-preserving potential functions are:

$$\varphi_\alpha(t) = \sqrt{\alpha + t^2}, \quad \alpha > 0, \tag{3}$$

$$\varphi_\alpha(t) = |t|^\alpha, \quad 1 < \alpha \leq 2, \tag{4}$$

$$\varphi_\alpha(t) = \log(\cosh(\alpha t)), \tag{5}$$

$$\varphi_\alpha(t) = |t|/\alpha - \log(1 + |t|/\alpha), \tag{6}$$

$$\varphi_\alpha(t) = \begin{cases} t^2/(2\alpha), & \text{if } |t| \leq \alpha, \\ |t| - \alpha/2, & \text{if } |t| > \alpha, \end{cases} \tag{7}$$

see [2, 4, 11, 17, 8]. The parameter α should be chosen such that $\varphi_\alpha(t)$ approximates $|t|$ for t away from zero, see [11]. In the following, we use φ'_α to denote the derivative of φ_α.

We note that $\mathcal{N}$ obtained in the noise detection phase in general is different from the *true noise set*:

$$\mathcal{N}_t := \{(i,j) \ : \ y_{i,j} \neq x_{i,j} \text{ in } (1)\}. \tag{8}$$

However, for salt-and-pepper noise, AMF is a very excellent noise detector and for images like those we tested in Section 5, the difference between $\mathcal{N}$ and $\mathcal{N}_t$ is usually less than 0.1% even for noise level r as high as 90%.

The two-phase schemes in [8, 10] both require the minimization of the functional (2) in the second phase, and it was done by a relaxation based method working on the residuals $z_{i,j} = u_{i,j} - y_{i,j}$, $(i,j) \in \mathcal{N}$. The use of such method was first proposed in [24]. It is sketched here below:

Algorithm 1 Relaxation method for minimizing (2)

1. Set $z_{ij} = 0$ for each $(i,j) \in \mathcal{N}$.

2. Iterate on $k = 1, 2, \cdots$ until convergence:
   For each $(i,j) \in \mathcal{N}$, consider $\mathcal{G}(\mathbf{u})$ as a 1D function of z_{ij} only, where $z_{i,j} = u_{i,j} - y_{i,j}$, and use the following three steps to compute its minimizer $z_{i,j}^{(k)}$:

   i. Calculate

   $$\xi_{i,j}^{(k)} = \beta \sum_{(m,n) \in \mathcal{V}_{i,j} \setminus \mathcal{N}} \varphi'_\alpha(y_{i,j} - y_{m,n}) + \beta \sum_{(m,n) \in \mathcal{V}_{i,j} \cap \mathcal{N}} \varphi'_\alpha(y_{i,j} - z_{m,n} - y_{m,n}),$$

   where $z_{m,n}$ are the latest updates.

   ii. If $|\xi_{i,j}^{(k)}| \leq 1$, set $z_{i,j}^{(k)} = 0$. Otherwise, solve for $z_{i,j}^{(k)}$ in the 1-D nonlinear equation:

   $$\text{sign}(\xi_{i,j}^{(k)}) = \beta \sum_{(m,n) \in \mathcal{V}_{i,j} \setminus \mathcal{N}} \varphi'_\alpha(z_{i,j}^{(k)} + y_{i,j} - y_{m,n})$$
   $$+ \beta \sum_{(m,n) \in \mathcal{V}_{i,j} \cap \mathcal{N}} \varphi'_\alpha(z_{i,j}^{(k)} + y_{i,j} - z_{m,n} - y_{m,n}).$$
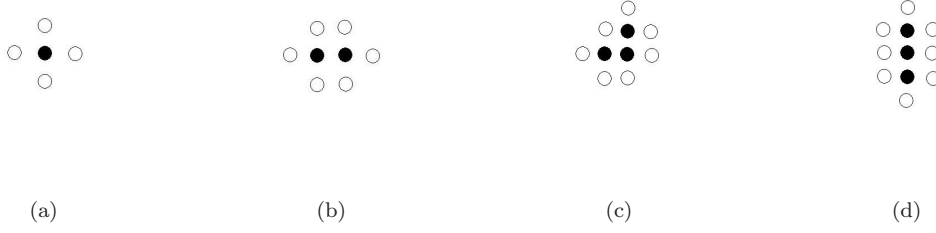
4

Figure 1: Pixels in $\mathcal{N}$ are labeled black while pixels not in $\mathcal{N}$ are labeled white.

    iii. Update $z_{i,j}$ to be $z_{i,j}^{(k)}$.

One can see that Algorithm 1 is a Gauss-Seidel type method where for each pixel $(i,j) \in \mathcal{N}$, we just minimize $\mathcal{G}$ with respect to the unknown variable $z_{i,j}$ and fix all other unknown variables to their latest computed values. It was shown in [24] that, under suitable assumptions on φ_α, the method converges. The 1-D nonlinear equations in Step 2(ii) can be solved by Newton's method. In [7], we have derived a formula for finding an initial guess for the Newton method that always guarantees convergence.

If a noise candidate $(i,j) \in \mathcal{N}$ is surrounded by pixels not in $\mathcal{N}$, see for example Figure 1(a), then the indeterminate $z_{i,j}$ appears in only one nonlinear equation. Hence in Step 2 above, we can obtain the required value of $z_{i,j}$, and hence of $u_{i,j}$, after 1 iteration. However, if noise candidates form a connected set, like those in Figure 1(b)–(d), then we need more iterations to find the required value of $z_{i,j}$ because the unknowns are coupled together. Clearly, the smaller such connected sets are, the faster the convergence of Algorithm 1 will be. In the extreme case when all the noise candidates are disconnected from each other, Algorithm 1 converges in one iteration.

Intuitively, if the noise level r increases, there will be more and larger connected sets of noise candidates. This is proven formally below. Since the noise candidate set $\mathcal{N}$ will change according to the noise detector used in phase 1, we prove the lemma for the true noise set $\mathcal{N}_t$ defined in (8).

Lemma 1 *Let the noise level be r and consider a noisy pixel $(i,j) \in \mathcal{N}_t$ far enough from the boundary. Let p_n be the probability that (i,j) is in a connected set consisting of n pixels in $\mathcal{N}_t$. Then*

$$p_n = nr^{n-1} \sum_{\mathcal{S}} (1-r)^{v_{\mathcal{S}}},$$

where the summation is taken over all possible shapes $\mathcal{S}$ of sets consisting of n connected pixels in $\mathcal{N}_t$, and $v_{\mathcal{S}}$ is the number of neighboring pixels of $\mathcal{S}$.

Proof: Assume that the noisy pixel $(i,j) \in \mathcal{N}_t$ is far enough from the boundary such that any shape $\mathcal{S}$ containing (i,j) does not touch the boundary. Then,

$$p_n = \sum_{\mathcal{S}} \text{prob}((i,j) \in \mathcal{S} | (i,j) \in \mathcal{N}_t) = \sum_{\mathcal{S}} \frac{\text{prob}((i,j) \in \mathcal{N}_t, (i,j) \in \mathcal{S})}{\text{prob}((i,j) \in \mathcal{N}_t)} = \sum_{\mathcal{S}} \frac{nr^n(1-r)^{v_{\mathcal{S}}}}{r},$$

since pixel (i,j) has n possible positions in a set consisting of n connected pixels in $\mathcal{N}_t$, and the probability of such set to appear is $r^n(1-r)^{v_{\mathcal{S}}}$. $\qquad\square$

We remark that for small n, it is not a serious requirement on (i,j) to be far from the boundary such that $\mathcal{S}$ containing (i,j) does not touch the boundary. This is because the number of pixels that
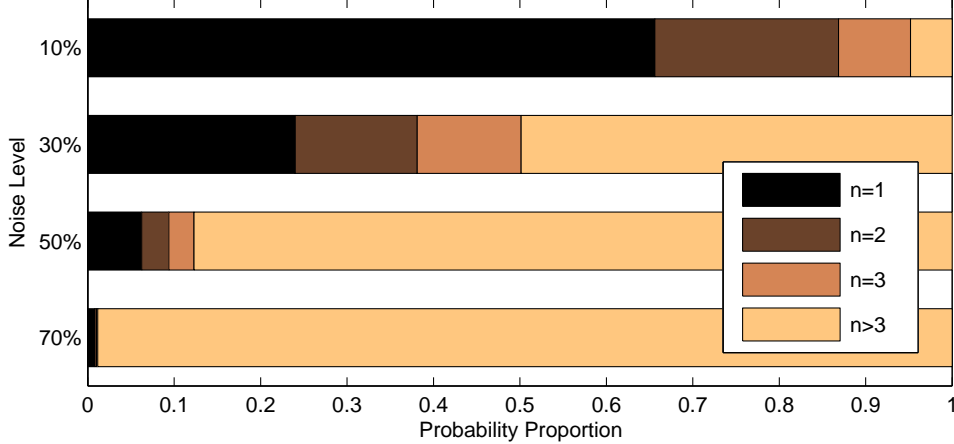
Figure 2: Probability for a noisy pixel to be in a connected set of n noisy pixels.

do not satisfy this is very small compared to the total number of pixels. By the lemma, we can easily get that

$$p_1 = (1-r)^4, \quad p_2 = 4r(1-r)^6, \quad p_3 = r^2(12(1-r)^7 + 6(1-r)^8).$$

We draw the proportion of $p_1, p_2, p_3, 1 - p_1 - p_2 - p_3$ in Figure 2 for different noise levels. As expected, the probability that a noisy pixel is in a small connected set is very large when the noise level is small. For example, when $r = 30\%$, the proportion of noisy pixels in connected subsets consisting only of $n = 1$ noisy pixel is about 24%, in those consisting of $n = 2$, $n = 3$ and $n > 3$ noisy pixels are about 14%, 12% and 50%, respectively. Since $\mathcal{N}$ approximates $\mathcal{N}_t$, Lemma 1 suggests that the relaxation method for the minimization of (2) converges fast when the noise level is low, but slow when the noise level becomes high. This fact has been verified numerically in [8, 10]. See also Section 5. There is a need to find better minimization algorithms for (2) for high noise level.

3   Properties of the New Functional

One main difficulties in minimizing (2) is that it contains the data-fitting term $|u_{i,j} - y_{i,j}|$ which is not continuously differentiable. It has already been observed in [9] that this term is not needed in the two-phase method. It is needed only in the original method in [24] where one wants to detect and restore noisy pixels simultaneously. Since the positions of the noisy pixels are not known a priori there, the term is required so that the restored values will not deviate too much from the observed values. Without this term, the minimizer will just be the zero vector.

However, in the two-phase method [8, 10], where the noise candidates are already detected at the first phase, there is no need to use the data-fitting term in the minimization in the second phase. This is because, in phase 2, the data-fitting is done exactly for uncorrupted pixels: $u_{i,j}^* = y_{i,j}$ if $(i, j) \notin \mathcal{N}$; and the restoration is done only on noise candidates. In fact, we observed already in [8, 10] that the minimum of (2) is independent of the regularization parameter β if it is large enough. Thus in [9],

6

we proposed to remove the data-fitting term and minimize the following new functional:

$$\mathcal{F}(\mathbf{u}) = \sum_{(i,j)\in\mathcal{N}} \left[\sum_{(m,n)\in\mathcal{V}_{i,j}\setminus\mathcal{N}} 2\,\varphi_\alpha(u_{i,j} - y_{m,n}) + \sum_{(m,n)\in\mathcal{V}_{i,j}\cap\mathcal{N}} \varphi_\alpha(u_{i,j} - u_{m,n}) \right] \qquad (9)$$

in the second phase of the two-phase method. The resulting restored images are basically the same as those obtained by using $\mathcal{G}$ in (2), see numerical results in [9, 6, 14] and also those in Section 5.

Clearly, the regularity of $\mathcal{F}$ now depends only on φ_α. If φ_α is smooth, then there are many methods for minimizing $\mathcal{F}$. For instance, in [9] and [6] we have used the Newton continuation method and a conjugate gradient type method, respectively. The aim of this section is to study some important analytical properties of $\mathcal{F}$. They will be used in Section 4 to establish the global convergence of CG-type [6, 28] and NSLQN [3, 15] optimization methods in minimizing $\mathcal{F}$. We remark that there is no convergence proof for the Newton continuation method used in [9]; and the convergence of the conjugate gradient type method in [6] was proved under the strong assumption $\varphi_\alpha''(t) > 0$. The results of this section will allow us to establish a convergence theorem for CG-type methods under weaker assumptions.

The following properties of $\mathcal{F}$ have already been proven in [6]. We recall that a function $f(\mathbf{x})$ is *coercive* if $f(\mathbf{x}) \to \infty$ when $\|\mathbf{x}\| \to \infty$.

Proposition 1 *If φ_α is respectively continuously differentiable, convex, strictly convex, or coercive, then the functional $\mathcal{F}$ is respectively continuously differentiable, convex, strictly convex, or coercive.*

In [6], we also proved that the global minimizer of $\mathcal{F}$ exists and satisfies the *maximum principle*:

Proposition 2 *If $\varphi_\alpha(t)$ is even, continuous and strictly increasing w.r.t. $|t|$, then the global minimum of $\mathcal{F}$ exists, and any global minimizer $\mathbf{u}^*$ is in the dynamic range, i.e., $u_{i,j}^* \in [s_{\min}, s_{\max}]$ for all $(i, j) \in \mathcal{N}$.*

Thus, the minimization of $\mathcal{F}$ is stable, and there is no need to restrict the iterants to be in the dynamic range during the minimization of $\mathcal{F}$. We remark that φ_α in (3)–(7) all satisfy the assumptions of the proposition.

In many optimization methods to minimize a functional f, the first (or second) order Lipschitz continuity or the strong convexity of f are sufficient conditions for global convergence [13, 25]. However, these properties, which involve the gradient or the Hessian of f, are generally difficult to verify. In the following two subsections, we show that for our functional $\mathcal{F}$, these properties are inherited from those of φ_α, which are easier to verify.

3.1 The Hessian of the New Functional

We first show that the boundedness and positive definiteness of the Hessian $\nabla^2\mathcal{F}(\mathbf{u})$ of $\mathcal{F}$ are inherited from the boundedness and positiveness of the second order derivative of φ_α.

Proposition 3 *Suppose that φ_α is even and φ_α'' exists.*

  (a) If φ_α'' is bounded, then $\nabla^2\mathcal{F}(\mathbf{u})$ is bounded independently of the dimension of the Hessian.

  (b) If $\varphi_\alpha''(t)$ is a nonnegative (resp. positive) function, then $\nabla^2\mathcal{F}(\mathbf{u})$ is positive semi-definite (resp. positive definite).

Proof: Let $H := \nabla^2 \mathcal{F}(\mathbf{u})$, i.e. $H_{[(i,j),(m,n)]} = \frac{\partial^2 \mathcal{F}(\mathbf{u})}{\partial u_{i,j} \partial u_{m,n}}$, $(i,j),(m,n) \in \mathcal{N}$. Then by (9), its entries are

$$H_{[(i,j),(i,j)]} = 2 \sum_{(m,n) \in \mathcal{V}_{i,j} \setminus \mathcal{N}} \varphi_\alpha''(u_{i,j} - y_{m,n}) + 2 \sum_{(m,n) \in \mathcal{V}_{i,j} \cap \mathcal{N}} \varphi_\alpha''(u_{i,j} - u_{m,n}), \tag{10}$$

and

$$H_{[(i,j),(m,n)]} = \begin{cases} -2\varphi_\alpha''(u_{i,j} - u_{m,n}), & \text{if } (m,n) \in \mathcal{V}_{i,j}, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Clearly H is a symmetric matrix. Define the matrix

$$W_{[(i,j),(m,n)]} = \begin{cases} 8, & \text{if } (m,n) = (i,j), \\ -2, & \text{if } (m,n) \in \mathcal{V}_{i,j}, \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$

i.e. W is obtained by letting $\varphi_\alpha''(u_{i,j} - y_{m,n}) = 1$ and $\varphi_\alpha''(u_{i,j} - u_{m,n}) = 1$ in (10) and (11). By (12), it is easy to see that W is a principal submatrix of the NM-by-NM matrix -2Δ, where Δ is the 2-D discrete Laplacian.

In order to verify (a), let M_α be a scalar such that $M_\alpha \geq \varphi_\alpha''(t) \geq -M_\alpha$ for all t. It is easy to show that the matrix $M_\alpha W - H$ has non-negative diagonals and non-positive off diagonals. Moreover, it is diagonally dominant. Hence it is positive semi-definite. Therefore

$$\mathbf{v}^T H \mathbf{v} \leq M_\alpha \mathbf{v}^T W \mathbf{v} \leq M_\alpha \lambda_{\max}(W) \leq 2M_\alpha \lambda_{\max}(-\Delta) \leq 16 M_\alpha,$$

where $\mathbf{v}$ is any normalized vector. Analogously, $\mathbf{v}^T H \mathbf{v} \geq -16 M_\alpha$. Therefore, $\|H\| \leq 16 M_\alpha$.

To verify (b), first let $\varphi_\alpha'' \geq m_\alpha \geq 0$. Proceeding as above, we observe that $H - m_\alpha W$ is positive semi-definite and hence

$$\lambda_{\min}(H) = \mathbf{w}^T H \mathbf{w} \geq m_\alpha \mathbf{w}^T W \mathbf{w} \geq m_\alpha \lambda_{\min}(W), \tag{13}$$

where $\mathbf{w}$ is the normalized eigenvector of H corresponding to its minimal eigenvalue $\lambda_{\min}(H)$. Thus H is positive semi-definite since $\lambda_{\min}(W) \geq 2\lambda_{\min}(-\Delta) \geq 0$.

Next consider the case $\varphi_\alpha'' > 0$ in $(-\infty, \infty)$. Let m_α be the minimal value of φ_α'' in (10) and (11) for all (i,j) and (m,n), then $m_\alpha > 0$ and (13) still holds. Therefore H is positive definite provided that $\lambda_{\min}(W) > 0$. This is the case if we impose zero, i.e. Dirichlet, boundary conditions on $\mathbf{u}$ in (9). In fact, with this imposition, Δ is the 2-D Laplacian with Dirichlet boundary conditions, hence $-\Delta$ is positive definite. Therefore, we have $\lambda_{\min}(W) > 0$ since W is a principal submatrix of -2Δ. Note that in our experiments in Section 5, we instead impose symmetric, i.e. Neumann, boundary conditions, for they usually give better restored images, see [22]. Then Δ is the 2D Laplacian with Neumann boundary conditions and hence $\lambda_{\min}(-\Delta) = 0$. However, since W is a principal submatrix of -2Δ, and the pixels detected as non-noise in the first step act as Dirichlet boundary conditions, W is in fact equal to -2 times a Laplacian with mixed Neumann and Dirichlet boundary conditions. It follows that W and hence H are still positive definite. In fact, we can show that rigorously. It suffices to show that $\mathbf{w}^T W \mathbf{w} \neq 0$ in (13). Since W is a principal submatrix of -2Δ, with a suitable permutation of the indices, we can write $\mathbf{w}^T W \mathbf{w} = -[\mathbf{w}^T, \mathbf{0}]\Delta[\mathbf{w}^T, \mathbf{0}]^T$. Note however that

$$\mathbf{x}^T W \mathbf{x} = -[\mathbf{x}^T, \mathbf{0}]\Delta[\mathbf{x}^T, \mathbf{0}]^T > 0 \tag{14}$$

for any nonzero vector $\mathbf{x}$, as the null vector of $-\Delta$ with Neumann boundary condition is the constant vector. Thus $\lambda_{\min}(H) \geq m_\alpha \mathbf{w}^T W \mathbf{w} > 0$. $\qquad \square$

Next we show that $\mathcal{F}$ is second order Lipschitz continuous if φ_α is. For this we need the following Lemma, where $\|\cdot\|$ denotes the 2-norm and $|\cdot|$ denotes taking the absolute value of the entries in the matrix or the vector.

8

Lemma 2 *Consider two square matrices A and B. If $|A| \leq |B|$ entry-wise, then $\|A\| \leq \||B|\|$.*

Proof: We can find a unit vector $\mathbf{w}_0$ such that $\|A\| = \|A\mathbf{w}_0\|$. Therefore,

$$\|A\| = \|A\mathbf{w}_0\| \leq \||A||\mathbf{w}_0|\| \leq \||B||\mathbf{w}_0|\| \leq \max_{\|\mathbf{w}\|=1} \||B|\mathbf{w}\| = \||B|\|.$$

$\square$

Proposition 4 *Suppose that φ_α is even and is second order Lipschitz continuous with Lipschitz constant ρ_1, i.e.,*

$$|\varphi_\alpha''(t_1) - \varphi_\alpha''(t_2)| \leq \rho_1 |t_1 - t_2|,$$

then $\mathcal{F}$ is also second order Lipschitz continuous with Lipschitz constant $32\rho_1$, i.e.,

$$\|\nabla^2 \mathcal{F}(\mathbf{u}) - \nabla^2 \mathcal{F}(\mathbf{v})\| \leq 32\rho_1 \|\mathbf{u} - \mathbf{v}\|.$$

Proof: Define $D := \nabla^2 \mathcal{F}(\mathbf{u}) - \nabla^2 \mathcal{F}(\mathbf{v})$. Then, for $(i,j) \in \mathcal{N}$,

$$
\begin{aligned}
|D_{[(i,j),(i,j)]}| &\leq 2 \sum_{(m,n) \in \mathcal{V}_{i,j} \setminus \mathcal{N}} |\varphi_\alpha''(u_{i,j} - y_{m,n}) - \varphi_\alpha''(v_{i,j} - y_{m,n})| \\
&\quad + 2 \sum_{(m,n) \in \mathcal{V}_{i,j} \cap \mathcal{N}} |\varphi_\alpha''(u_{i,j} - u_{m,n}) - \varphi_\alpha''(v_{i,j} - v_{m,n})| \\
&\leq 2\rho_1 \sum_{(m,n) \in \mathcal{V}_{i,j} \setminus \mathcal{N}} |u_{i,j} - v_{i,j}| + 2\rho_1 \sum_{(m,n) \in \mathcal{V}_{i,j} \cap \mathcal{N}} (|u_{i,j} - v_{i,j}| + |u_{m,n} - v_{m,n}|) \\
&\leq 2\rho_1 \|\mathbf{u} - \mathbf{v}\|_\infty |W_{[(i,j),(i,j)]}|.
\end{aligned}
$$

Similarly, if $(i,j), (m,n) \in \mathcal{N}$, $(i,j) \neq (m,n)$, then

$$|D_{[(i,j),(m,n)]}| \leq 2\rho_1 \|\mathbf{u} - \mathbf{v}\|_\infty |W_{[(i,j),(m,n)]}|.$$

Therefore, by Lemma 2

$$\|D\| \leq 2\rho_1 \||W|\| \|\mathbf{u} - \mathbf{v}\|_\infty \leq 32\rho_1 \|\mathbf{u} - \mathbf{v}\|_\infty \leq 32\rho_1 \|\mathbf{u} - \mathbf{v}\|.$$

$\square$

3.2 The Gradient of the New Functional

In this subsection, we show that the strong convexity and the first order Lipschitz continuity of $\mathcal{F}$ can be inherited from φ_α too. To this end, we first obtain some compact formulas for $\mathcal{F}(\mathbf{u})$, $\nabla\mathcal{F}(\mathbf{u})$ and $\nabla^2\mathcal{F}(\mathbf{u})$. Note that $\mathcal{F}(\mathbf{u})$ in (9) is the restriction of the following regularization functional onto the noise candidate set $\mathcal{N}$:

$$\sum_{(i,j) \in \mathcal{A}} \sum_{(m,n) \in \mathcal{V}_{i,j}} \varphi_\alpha(u_{i,j} - u_{m,n}),$$

where $\mathcal{A}$ is the set of all pixels, see [8]. Hence $\mathcal{F}(\mathbf{u})$ can be rewritten as

$$\mathcal{F}(\mathbf{u}) = \sum_{(i,j) \in \mathcal{A}} \sum_{(m,n) \in \mathcal{V}_{i,j}} \varphi_\alpha(z_{i,j} - z_{m,n}) - \sum_{(i,j) \notin \mathcal{N}} \sum_{(m,n) \in \mathcal{V}_{i,j} \setminus \mathcal{N}} \varphi_\alpha(y_{i,j} - y_{m,n}), \tag{15}$$

with

$$z_{k,l} = \begin{cases} u_{k,l}, & \text{if } (k,l) \in \mathcal{N}, \\ y_{k,l}, & \text{if } (k,l) \notin \mathcal{N}. \end{cases}$$

Recall that $\mathcal{V}_{i,j}$ is the set of the four neighboring pixels of (i,j). We are going to write the first summation in (15) as the sum of four numbers $\{f_\ell\}_{\ell=1}^4$, where each f_ℓ is the first summation in (15) summed over just one of the neighboring pixels of (i,j). More precisely, if f_1 corresponds to the pixel $(i+1,j)$, then

$$f_1 := \sum_{(i,j)\in\mathcal{A}} \varphi_\alpha(z_{i,j} - z_{i+1,j}) = \sum_{(i,j)\in\mathcal{A}} \begin{cases} \varphi_\alpha(u_{i,j} - u_{i+1,j}), & \text{if } (i,j) \in \mathcal{N} \text{ and } (i+1,j) \in \mathcal{N}, \\ \varphi_\alpha(u_{i,j} - y_{i+1,j}), & \text{if } (i,j) \in \mathcal{N} \text{ and } (i+1,j) \notin \mathcal{N}, \\ \varphi_\alpha(y_{i,j} - u_{i+1,j}), & \text{if } (i,j) \notin \mathcal{N} \text{ and } (i+1,j) \in \mathcal{N}, \\ \varphi_\alpha(y_{i,j} - y_{i+1,j}), & \text{if } (i,j) \notin \mathcal{N} \text{ and } (i+1,j) \notin \mathcal{N}. \end{cases} \quad (16)$$

Notice that if $\varphi(t)$ is an even function, then we have two terms of the form $\varphi(u_{\cdot,\cdot} - y_{\cdot,\cdot})$ in (16), which explains again why there is the extra factor "2" for such terms in (2) and (9).

If we define the matrix G_1 and the vector $\mathbf{b}_1$ by

$$(G_1)_{[(i,j),(m,n)]} = \begin{cases} 1, & \text{if } (m,n) = (i,j) \in \mathcal{N}, \\ -1, & \text{if } (i,j) \in \mathcal{N} \text{ and } (m,n) = (i+1,j) \in \mathcal{N}, \\ -1, & \text{if } (i,j) \notin \mathcal{N} \text{ and } (m,n) = (i+1,j) \in \mathcal{N}, \\ 0, & \text{otherwise}, \end{cases}$$

and

$$(\mathbf{b}_1)_{(i,j)} = \begin{cases} y_{i+1,j}, & \text{if } (i,j) \in \mathcal{N} \text{ and } (i+1,j) \notin \mathcal{N}, \\ -y_{i,j}, & \text{if } (i,j) \notin \mathcal{N} \text{ and } (i+1,j) \in \mathcal{N}, \\ 0, & \text{otherwise}; \end{cases}$$

for $(i,j) \in \mathcal{A}$ and $(m,n) \in \mathcal{N}$, then we can rewrite f_1 in (16) as

$$f_1 = \sum_{(i,j)\in\mathcal{A}} \varphi_\alpha([G_1\mathbf{u} - \mathbf{b}_1]_{i,j}) + \sum_{(i,j)\notin\mathcal{N}} \sum_{(i+1,j)\notin\mathcal{N}} \varphi_\alpha(y_{i,j} - y_{i+1,j}) - \sum_{(i,j)\notin\mathcal{N}} \sum_{(i+1,j)\notin\mathcal{N}} \varphi_\alpha(0).$$

We can define $\{G_\ell\}_{\ell=2}^4$ and $\{\mathbf{b}_\ell\}_{\ell=2}^4$ similarly to obtain $\{f_\ell\}_{\ell=2}^4$ for the other pixels in $\mathcal{V}_{i,j}$.

Putting f_ℓ back into (15) and using the notation $\varphi_\alpha([w_1, w_2, \cdots]^T) := [\varphi_\alpha(w_1), \varphi_\alpha(w_2), \cdots]^T$, we obtain a compact formula for $\mathcal{F}(\mathbf{u})$:

$$\begin{aligned} \mathcal{F}(\mathbf{u}) &= \sum_{\ell=1}^4 \varphi_\alpha(G_\ell\mathbf{u} - \mathbf{b}_\ell)^T \mathbf{1} - \sum_{(i,j)\notin\mathcal{N}} \sum_{(m,n)\in\mathcal{V}_{i,j}\backslash\mathcal{N}} \varphi_\alpha(0) \\ &= \varphi_\alpha(G\mathbf{u} - \mathbf{b})^T \mathbf{1} - \sum_{(i,j)\notin\mathcal{N}} \sum_{(m,n)\in\mathcal{V}_{i,j}\backslash\mathcal{N}} \varphi_\alpha(0), \end{aligned}$$

where $\mathbf{1}$ is the vector of all ones, $G = [G_1^T, G_2^T, G_3^T, G_4^T]^T$ and $\mathbf{b} = [\mathbf{b}_1^T, \mathbf{b}_2^T, \mathbf{b}_3^T, \mathbf{b}_4^T]^T$. Therefore

$$\nabla\mathcal{F}(\mathbf{u}) = G^T \varphi'_\alpha(G\mathbf{u} - \mathbf{b}), \quad (17)$$

and

$$\nabla^2\mathcal{F}(\mathbf{u}) = G^T \text{diag}(\varphi''_\alpha(G\mathbf{u} - \mathbf{b}))G.$$

Note that the above formula for $\nabla^2\mathcal{F}(\mathbf{u})$ and (12) imply the following interesting identity:

$$W = \nabla^2\mathcal{F}(\mathbf{u})\big|_{\varphi''_\alpha \equiv 1} = G^T G. \quad (18)$$

With all these preparations, we are now ready to establish the first order Lipschitz continuity and the strong convexity of $\mathcal{F}$.

10

Proposition 5 *Let φ_α be convex and first order Lipschitz continuous with Lipschitz constant ρ_2, i.e.,*

$$|\varphi_\alpha'(t_1) - \varphi_\alpha'(t_2)| \leq \rho_2 |t_1 - t_2|, \tag{19}$$

then

$$\|\nabla\mathcal{F}(\mathbf{u}) - \nabla\mathcal{F}(\mathbf{v})\|^2 \leq 16\rho_2 (\nabla\mathcal{F}(\mathbf{u}) - \nabla\mathcal{F}(\mathbf{v}))^T(\mathbf{u} - \mathbf{v}). \tag{20}$$

Proof: The representation of $\nabla\mathcal{F}$ in (17) implies

$$\|\nabla\mathcal{F}(\mathbf{u}) - \nabla\mathcal{F}(\mathbf{v})\|^2 = \|G^T\varphi_\alpha'(G\mathbf{u} - \mathbf{b}) - G^T\varphi_\alpha'(G\mathbf{v} - \mathbf{b})\|^2 = \Phi^T GG^T\Phi,$$

where we have defined $\Phi = \varphi_\alpha'(G\mathbf{u}-\mathbf{b}) - \varphi_\alpha'(G\mathbf{v}-\mathbf{b})$. Note that by (18), $\lambda_{\max}(GG^T) = \lambda_{\max}(G^TG) = \lambda_{\max}(W) \leq 16$. Combining the convexity condition $(\varphi_\alpha'(t_1)-\varphi_\alpha'(t_2))(t_1-t_2) \geq 0$ with (19), we obtain the inequality

$$(\varphi_\alpha'(t_1) - \varphi_\alpha'(t_2))^2 \leq \rho_2(\varphi_\alpha'(t_1) - \varphi_\alpha'(t_2))(t_1 - t_2)$$

which implies

$$\begin{aligned}
\|\nabla\mathcal{F}(\mathbf{u}) - \nabla\mathcal{F}(\mathbf{v})\|^2 &\leq 16\Phi^T\Phi \leq 16\rho_2\Phi^T G(\mathbf{u} - \mathbf{v}) = 16\rho_2(G^T\Phi)^T(\mathbf{u} - \mathbf{v}) \\
&= 16\rho_2(\nabla\mathcal{F}(\mathbf{u}) - \nabla\mathcal{F}(\mathbf{v}))^T(\mathbf{u} - \mathbf{v}).
\end{aligned}$$

$\square$

Since $\mathcal{F}$ is convex, (20) also follows if $\mathcal{F}$ has continuous second order derivatives which are bounded in a convex set containing $\mathbf{u}$ and $\mathbf{v}$ (see for example, Proposition 3.4 in [15]). Proposition 5 shows how our special $\mathcal{F}$ allows us to obtain the same result by the weaker condition (19). By the Cauchy-Schwartz inequality, we have the following corollary.

Corollary 1 *If φ_α is convex and first order Lipschitz continuous with Lipschitz constant ρ_2, i.e. (19) holds, then $\mathcal{F}$ is also first order Lipschitz continuous with Lipschitz constant $16\rho_2$, i.e.,*

$$\|\nabla\mathcal{F}(\mathbf{u}) - \nabla\mathcal{F}(\mathbf{v})\| \leq 16\rho_2\|\mathbf{u} - \mathbf{v}\|.$$

Proposition 6 *If φ_α is strongly convex, i.e., there exists a positive constant $c > 0$ such that*

$$(\varphi_\alpha'(t_1) - \varphi_\alpha'(t_2))(t_1 - t_2) \geq c(t_1 - t_2)^2,$$

then the functional $\mathcal{F}$ is also strongly convex, i.e., there exists a positive constant $c_1 > 0$ such that

$$(\nabla\mathcal{F}(\mathbf{u}) - \nabla\mathcal{F}(\mathbf{v}))^T(\mathbf{u} - \mathbf{v}) \geq c_1\|\mathbf{u} - \mathbf{v}\|^2.$$

Proof: By using (17) and the strong convexity of φ_α, we get

$$\begin{aligned}
(\nabla\mathcal{F}(\mathbf{u}) - \nabla\mathcal{F}(\mathbf{v}))^T(\mathbf{u} - \mathbf{v}) &= [\varphi_\alpha'(G\mathbf{u} - \mathbf{b}) - \varphi_\alpha'(G\mathbf{v} - \mathbf{b})]^T G(\mathbf{u} - \mathbf{v}) \\
&= [\varphi_\alpha'(G\mathbf{u} - \mathbf{b}) - \varphi_\alpha'(G\mathbf{v} - \mathbf{b})]^T[(G\mathbf{u} - \mathbf{b}) - (G\mathbf{v} - \mathbf{b})] \\
&\geq c(\mathbf{u} - \mathbf{v})^T G^T G(\mathbf{u} - \mathbf{v}).
\end{aligned}$$

Using (18) and (14), we then have

$$(\nabla\mathcal{F}(\mathbf{u}) - \nabla\mathcal{F}(\mathbf{v}))^T(\mathbf{u} - \mathbf{v}) \geq c(\mathbf{u} - \mathbf{v})^T W(\mathbf{u} - \mathbf{v}) \geq c\lambda_{\min}(W)\|\mathbf{u} - \mathbf{v}\|^2,$$

where $c\lambda_{\min}(W) > 0$. $\square$

In Table 1, we give the properties of $\mathcal{F}$ that are inherited from the potential functions φ_α given in (3)–(7). Strong convexity of $\mathcal{F}$ may not hold for many potential functions in the whole domain $\mathbb{R}^{|\mathcal{N}|}$, but it may hold if we restrict the domain onto a bounded set. In fact, many optimization methods work in a level set and the level set of our $\mathcal{F}$ is bounded due to the coercivity. Therefore, it is meaningful to test the strong convexity of $\mathcal{F}$ on bounded sets.

11

Table 1: Properties of $\mathcal{F}$.

	Convexity and Coercivity	Lipschitz Continuity		Strong Convexity in Bounded Sets	Uniformly Bounded Hessian	Positive Definite Hessian
		First	Second			
$\sqrt{\alpha + t^2}$	✓	✓	✓	✓	✓	✓
$|t|^\alpha$	✓	–	–	✓	–	–
$\log(\cosh(\alpha t))$	✓	✓	✓	✓	✓	✓
$|t|/\alpha - \log(1 + |t|/\alpha)$	✓	✓	✓	✓	✓	✓
$\begin{cases} t^2/(2\alpha), & \text{if } |t| \le \alpha, \\ |t| - \alpha/2, & \text{if } |t| > \alpha, \end{cases}$	✓	✓	–	–	–	–

4 Minimization of the New Functional

The results established in the previous section are useful in proving the convergence of optimization methods applied to $\mathcal{F}$. In this section, we show, in particular, that a conjugate gradient (CG) type method [6, 28] and a low-complexity quasi-Newton method [3, 15] are globally convergent.

4.1 A Conjugate Gradient Type Method

The minimization of $\mathcal{F}$ in [6] was done by an efficient CG method [28] where the line search rule is replaced by a predetermined step length formula. We briefly describe the method here.

Algorithm 2 Conjugate gradient (CG) method without line search to minimize $f(\mathbf{x})$.

1. Choose an initial guess $\mathbf{x}_0$ and set $\mathbf{d}_0 = -\mathbf{g}_0$ where $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$.

2. For $k = 0, 1, 2, \cdots$

    (a) $\mathbf{d}_k = -\mathbf{g}_k + \gamma_k \mathbf{d}_{k-1}$, where $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$,

    (b) $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$, where $\alpha_k = -\delta \mathbf{g}_k^T \mathbf{d}_k / \mathbf{d}_k^T Q_k \mathbf{d}_k$, Q_k is any symmetric positive definite matrix and δ is some fixed positive parameter depending on Q_k.

Some well-known formulae for γ_k are:

$$\gamma_k^{FR} = \frac{\|\mathbf{g}_k\|^2}{\|\mathbf{g}_{k-1}\|^2} \qquad \text{(Fletcher-Reeves)}, \tag{21}$$

$$\gamma_k^{PR} = \frac{\mathbf{g}_k^T(\mathbf{g}_k - \mathbf{g}_{k-1})}{\|\mathbf{g}_{k-1}\|^2} \qquad \text{(Polak-Ribière)}, \tag{22}$$

$$\gamma_k^{HS} = \frac{\mathbf{g}_k^T(\mathbf{g}_k - \mathbf{g}_{k-1})}{\mathbf{d}_{k-1}^T(\mathbf{g}_k - \mathbf{g}_{k-1})} \qquad \text{(Hestenes-Stiefel)}, \tag{23}$$

$$\gamma_k^{CD} = \frac{\|\mathbf{g}_k\|^2}{-\mathbf{d}_{k-1}^T \mathbf{g}_{k-1}} \qquad \text{(The Conjugate Descent Method)}, \tag{24}$$

$$\gamma_k^{DY} = \frac{\|\mathbf{g}_k\|^2}{\mathbf{d}_{k-1}^T(\mathbf{g}_k - \mathbf{g}_{k-1})} \qquad \text{(Dai-Yuan)}, \tag{25}$$

see [28] for references therein.

In [6], we proved the convergence of this CG method for minimizing $\mathcal{F}$ under the assumption that $\varphi_\alpha'' > 0$. Using the results in Section 3, we show that the assumption can be relaxed, and hence the method can be applied to more φ_α. We start with the convergence theorem stated in [28] for general functions $f(\mathbf{x})$.

Theorem 1 *Let $\{\mathbf{x}_k\}$ be the sequence generated by Algorithm 2. If there exists a neighborhood Ω of the bounded level set $\Upsilon = \{\mathbf{x} | f(\mathbf{x}) \le f(\mathbf{x}_0)\}$ such that ∇f is Lipschitz continuous on Ω, then for the choice of γ_k in (24) and (22), $\liminf_{k\to\infty} \|\nabla f(\mathbf{x}_k)\| \to 0$. If f is strongly convex in Ω, then the same conclusion holds for (21), (23), and (25).*

Applying the theorem to our $\mathcal{F}$, we have the following global convergence result.

Theorem 2 *Let φ_α be even, convex, continuously differentiable and strictly increasing in $|t|$. Let $\{\mathbf{u}_k\}$ be the sequence generated by applying Algorithm 2 to $\mathcal{F}$.*

1. *If φ_α is first order Lipschitz continuous, then there exists a subsequence of $\{\mathbf{u}_k\}$ converging to a global minimizer $\mathbf{u}^*$ of $\mathcal{F}$ for the choice of γ_k in (24) and (22).*

2. *If φ_α is strongly convex in every bounded set, then there exists a subsequence of $\{\mathbf{u}_k\}$ converging to a global minimizer $\mathbf{u}^*$ of $\mathcal{F}$ for the choice of γ_k in (21), (23) and (25).*

Moreover, the entries of $\mathbf{u}^*$ are all in the range $[s_{\min}, s_{\max}]$.

Proof: By Proposition 2, the global minimum exists. It is easy to verify that φ_α is coercive. Hence, by Proposition 1, $\mathcal{F}$ is is differentiable, convex and coercive. Therefore, any stationary point of $\mathcal{F}$ is a global minimizer of $\mathcal{F}$, see [25, Theorem 2.5]. Moreover, the level set $\Upsilon = \{\mathbf{u} | \mathcal{F}(\mathbf{u}) \le \mathcal{F}(\mathbf{u}_0)\}$ is a bounded set.

We construct the neighborhood Ω of Υ required in Theorem 1 as follows. Let $(u_0)_{i,j}$ be an arbitrary component of $\mathbf{u}_0$, and

$$z = \max \left\{ |(u_0)_{i,j}|, \max_{(m,n) \in \mathcal{V}_{i,j}} |(u_0)_{(m,n)}| \right\}.$$

Then we define a new vector $\mathbf{w}$ by replacing the entry $(u_0)_{i,j}$ by $w_{i,j} = 1 + 3z$. Then, for any neighbors v of $(u_0)_{i,j}$, we have

$$
\begin{aligned}
|(u_0)_{i,j} - v| &< 1 + (|v| - v) + |(u_0)_{i,j}| + |v| \\
&= 1 + |(u_0)_{i,j}| + 2|v| - v \le 1 + 3z - v = |w_{i,j} - v|,
\end{aligned}
$$

and consequently, $\mathcal{F}(\mathbf{u}_0) < \mathcal{F}(\mathbf{w})$. Therefore, $\Upsilon \subseteq \Omega := \{\mathbf{u} \mid \mathcal{F}(\mathbf{u}) < \mathcal{F}(\mathbf{w})\}$. By the continuity of φ_α, Ω is an open set, hence is a neighborhood of Υ. Moreover, Ω is bounded because of the coercivity.

If φ_α is first order Lipschitz continuous, then $\nabla \mathcal{F}$ is Lipschitz continuous by Corollary 1. If φ_α is strongly convex in every bounded set, then $\mathcal{F}$ is strongly convex in the bounded set Ω by Proposition 6. Therefore, in both cases, $\liminf_{k\to\infty} \|\nabla \mathcal{F}(\mathbf{u}_k)\| = 0$ by Theorem 1, which means that a subsequence of $\{\mathbf{u}_k\}$ converges to a stationary point $\mathbf{u}^*$ of $\mathcal{F}$, which is a global minimizer of $\mathcal{F}$. Moreover because of the maximum principle in Proposition 2, the entries of $\mathbf{u}^*$ are all in the range of $[s_{\min}, s_{\max}]$. $\square$

By Theorem 2, Algorithm 2 applied to $\mathcal{F}$ with (3), (5), (6) and (7) is globally convergent with γ_k given by (24) and (22); and when applied to $\mathcal{F}$ with (3), (4), (5) and (6), it is global convergent with γ_k given by (21), (23) and (25). In comparison, in [6], we can only get the convergence results for (3), (5) and (6) under the stronger assumption $\varphi_\alpha'' > 0$.

13

4.2 A Low-complexity Quasi-Newton Method

BFGS is one of the most efficient quasi-Newton methods [13, 25] for minimization. However, it is not practical for large scale problems because $O(n^2)$ operations and memories are required for each iteration, where n is the number of unknowns. In comparison, Algorithm 2 requires $O(n)$ operations per iteration. In [3, 15], a low complexity method, called LQN method, was proposed by projecting the BFGS Hessian onto a matrix algebra $\mathbf{L}_U$ which contains matrices that can be diagonalized by a given unitary matrix U. The non-secant version of LQN is given below.

Algorithm 3 `NSLQN: Non-Secant version of LQN method for minimizing` $f(\mathbf{x})$`:`

1. `Set` $\mathbf{x}_0$ `and an initial` H_0 `which has to be positive definite (f.i.` $H_0 = I$`).`

2. `For` $k = 0, 1, 2, \ldots$`, do while` $\nabla f(\mathbf{x}_k) \neq \mathbf{0}$`:`

   (a) `Compute the eigenvalues of the matrix` $\mathcal{P}(H_k)$`,` $\|\mathcal{P}(H_k) - H_k\|_F \leq \|X - H_k\|_F$`, for all` $X \in \mathbf{L}_U$`, which is the orthogonal projection in Frobenius norm of` H_k `onto` $\mathbf{L}_U$`.`

   (b) $\mathbf{d}_k = -\mathcal{P}(H_k)^{-1}\nabla f(\mathbf{x}_k)$`.`

   (c) $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$`, where` λ_k `is a step length defined by the Armijo-Goldstein line search` [13]`.`

   (d) $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$`,` $\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$`, and`

$$H_{k+1} = \mathcal{P}(H_k) + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} - \frac{\mathcal{P}(H_k)\mathbf{s}_k \mathbf{s}_k^T \mathcal{P}(H_k)}{\mathbf{s}_k^T \mathcal{P}(H_k)\mathbf{s}_k}.$$

We remark that if $\mathbf{z}_k$ is the vector of the eigenvalues of $\mathcal{P}(H_k)$, i.e. $\mathcal{P}(H_k) = Ud(\mathbf{z}_k)U^*$, where $d(\mathbf{v}) := \text{diag}(v_1, \ldots, v_n)$, then $\mathbf{z}_k$ satisfies the identity

$$\mathbf{z}_k = \mathbf{z}_{k-1} + \frac{1}{\mathbf{y}_{k-1}^T \mathbf{s}_{k-1}}|U^*\mathbf{y}_{k-1}|^2 - \frac{1}{\mathbf{z}_{k-1}^T|U^*\mathbf{s}_{k-1}|^2}d(\mathbf{z}_{k-1})^2|U^*\mathbf{s}_{k-1}|^2$$

where $|\mathbf{v}| := (|v_1|, \ldots, |v_n|)$. So, we need not compute the eigenvalues of $\mathcal{P}(H_k)$ from scratch. In fact, they can be obtained via the above identity from the eigenvalues of $\mathcal{P}(H_{k-1})$ with a number of operations proportional to the cost of matrix-vector products involving the matrix U. It is then clear that the computational complexity of the method depends on the choice of U. Some common choices are the Fourier, Hartley, cosine, and sine transform matrices [3, 15, 16], and Steps 2 (a) and (b) will then require $O(n \log n)$ operations.

The convergence of the algorithm has been established in [15]:

Theorem 3 *Let* $\{\mathbf{x}_k\}$ *be the sequence generated by Algorithm 3. If the level set* $\Upsilon = \{\mathbf{x}|f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$ *is bounded and*

$$\frac{\|\mathbf{y}_k\|^2}{\mathbf{y}_k^T \mathbf{s}_k} \leq c, \quad k = 0, 1, 2, \ldots \tag{26}$$

for some c, then $\liminf_{k \to \infty} \|\nabla f(\mathbf{x}_k)\| = 0$.

Using results in Section 3 and Theorem 3, we can easily prove the global convergence of Algorithm 3 when f is equal to the functional $\mathcal{F}$ in (9).

14

Theorem 4 *Let $\{\mathbf{u}_k\}$ be the sequence generated by Algorithm 3 applied to $\mathcal{F}$. If the potential function $\varphi_\alpha(t)$ is even, convex, first order Lipschitz continuous and strictly increasing w.r.t. $|t|$, then a subsequence of $\{\mathbf{u}_k\}$ converges to a global minimizer $\mathbf{u}^*$ of $\mathcal{F}$. Moreover, the entries of $\mathbf{u}^*$ are all in the range of $[s_{min}, s_{max}]$.*

Proof: As we have been seen in the proof of Theorem 2, the level set $\Upsilon = \{\mathbf{u}|\mathcal{F}(\mathbf{u}) \leq \mathcal{F}(\mathbf{u}_0)\}$ is bounded and any stationary point of $\mathcal{F}$ is a global minimizer of $\mathcal{F}$. By Proposition 5, (26) is verified. Therefore, $\liminf_{k\to\infty} \|\nabla\mathcal{F}(\mathbf{u}_k)\| = 0$, and a subsequence of $\{\mathbf{u}_k\}$ converges to a stationary point $\mathbf{u}^*$ of $\mathcal{F}$. By Proposition 2, the entries of $\mathbf{u}^*$ are all in $[s_{\min}, s_{\max}]$. $\square$

We remark that the potential functions (3), (5), (6) and (7) satisfy the conditions of Theorem 4, whereas (4) does not. Also the convergence results here are valid for all matrix algebra $\mathbf{L}_U$. Clearly how fast the method converges depends on how well $\mathcal{P}(H_k)$ approximates the Hessian $\nabla^2\mathcal{F}(\mathbf{u}_k)$, and this in turns depends on the choice of U again.

5 Numerical Experiments

In this section, we test the three algorithms (Algorithms 1–3) discussed in the paper. Note that Algorithm 1, the relaxation method, is applied to the old functional $\mathcal{G}$ in (2), whereas the CG and NSLQN methods are applied to the new functional $\mathcal{F}$ in (9). As pointed out in [8], Algorithm 1 is robust with respect to the regularization parameter β. Hence we fix $\beta = 10$ for Algorithm 1. In Algorithm 2, we define γ_k according to (22), and Q_k is chosen to be the identity matrix I. For Algorithm 3, we use the simplest matrix algebra—the algebra of all diagonal matrices, i.e. $U = I$. Hence $\mathcal{P}(H) = \mathrm{diag}(H)$ for all H, i.e. we approximate the Hessian by the diagonal part of H_k. The algorithm thus requires $O(n)$ operations per iteration.

We remark that both Algorithms 2 and 3 are globally convergent according to Theorems 2 and 4, so we use the output of AMF method (the noise detector in phase 1) as our initial guess for phase 2. In contrast, Algorithm 1 is only local convergent. Therefore we use the initial guess derived in [7] to guarantee convergence.

We test the following two potential functions in our experiments: $\varphi_\alpha(t) = \sqrt{t^2 + \alpha}$ with $\alpha = 100$ [6], and the Huber function

$$\varphi_\alpha(t) = \begin{cases} t^2/(2\alpha), & \text{if } |t| \leq \alpha, \\ |t| - \alpha/2, & \text{if } |t| > \alpha, \end{cases}$$

with $\alpha = 10$. The stopping criteria of the minimization are

$$\frac{\|\mathbf{u}^k - \mathbf{u}^{k-1}\|}{\|\mathbf{u}^k\|} \leq 10^{-4} \quad \text{and} \quad \frac{|\mathcal{F}(\mathbf{u}^k) - \mathcal{F}(\mathbf{u}^{k-1})|}{\mathcal{F}(\mathbf{u}^k)} \leq 10^{-4}.$$

We assess the restoration performance by using the peak signal to noise ratio (PSNR) defined as

$$\mathrm{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{MN}\sum_{i,j}(u^*_{i,j} - x_{i,j})^2},$$

where $u^*_{i,j}$ and $x_{i,j}$ are the pixel values of the restored image and of the original image, respectively (see [5]). In order to test the speed of the algorithms more fairly, the experiments are repeated for 10 different noise samples of each image and the average of the 10 results is given in the tables.

Tables 2–3 show the time required by the algorithms for restoring 512-by-512 images contaminated by salt-and-pepper noise with different noise levels. We can see from the tables that CG or NSLQN methods are faster than the relaxation method especially when the noise level is high; the relaxation

15

method has a comparable performance only when the noise level is low, a fact proven in Lemma 1. Moreover the restored results are basically the same either we minimize $\mathcal{G}$ or $\mathcal{F}$. Figure 3 shows the restoration results obtained by applying Algorithms 1–3 to the Lena image corrupted with 70% and 90% salt-and-pepper noise, and Figure 4 shows the results for four other images corrupted with 70% salt-and-pepper noise. The corresponding PSNR values are listed there.

Table 2: Performance of salt-and-pepper denoising algorithms with potential function $\sqrt{t^2 + \alpha}$.

Image	Noise Level	Relaxation		CG		NSLQN	
		Time	PSNR	Time	PSNR	Time	PSNR
Lena	30%	30.7	36.4	48.6	36.5	42.7	36.6
	50%	61.4	32.9	56.0	33.0	62.1	33.0
	70%	127	29.7	80.2	29.8	89.0	29.8
	90%	452	25.4	139	25.4	142	25.4
Bridge		246	24.9	86.1	24.9	90.6	24.9
Goldhill	70%	123	29.8	55.8	29.8	59.6	29.8
Barbara		127	24.6	72.8	24.7	75.7	24.7
Boat		116	27.9	73.9	28.0	79.5	28.0

Table 3: Performance of salt-and-pepper denoising algorithms with Huber's potential function.

Image	Noise Level	Relaxation		CG		NSLQN	
		Time	PSNR	Time	PSNR	Time	PSNR
Lena	30%	40.7	35.7	33.4	36.2	34.7	36.1
	50%	68.0	32.4	54.7	32.7	57.2	32.7
	70%	135	29.3	81.5	29.5	88.9	29.5
	90%	364	25.1	141	25.2	155	25.2
Bridge		207	24.8	111	24.8	113	24.8
Goldhill	70%	132	29.7	58.4	29.8	69.7	29.8
Barbara		118	24.5	79.7	24.6	86.3	24.6
Boat		119	27.8	86.8	27.9	90.9	27.9

6 Conclusions

In this paper, we have studied the analytical properties of the functional $\mathcal{F}$ given in (9), and we have used them to show the convergence of conjugate gradient-type and non-secant LQN methods when minimizing $\mathcal{F}$. The minimization of $\mathcal{F}$ is required by 2-phase impulse denoising algorithms, and has been implemented, until now, via a 1D relaxation method. Here, we have proved both theoretically and by experiments that such method becomes slow when the noise level is high. The numerical experiments in Section 5 show that, in this case, the above CG and NSLQN methods are good alternatives to the relaxation method.
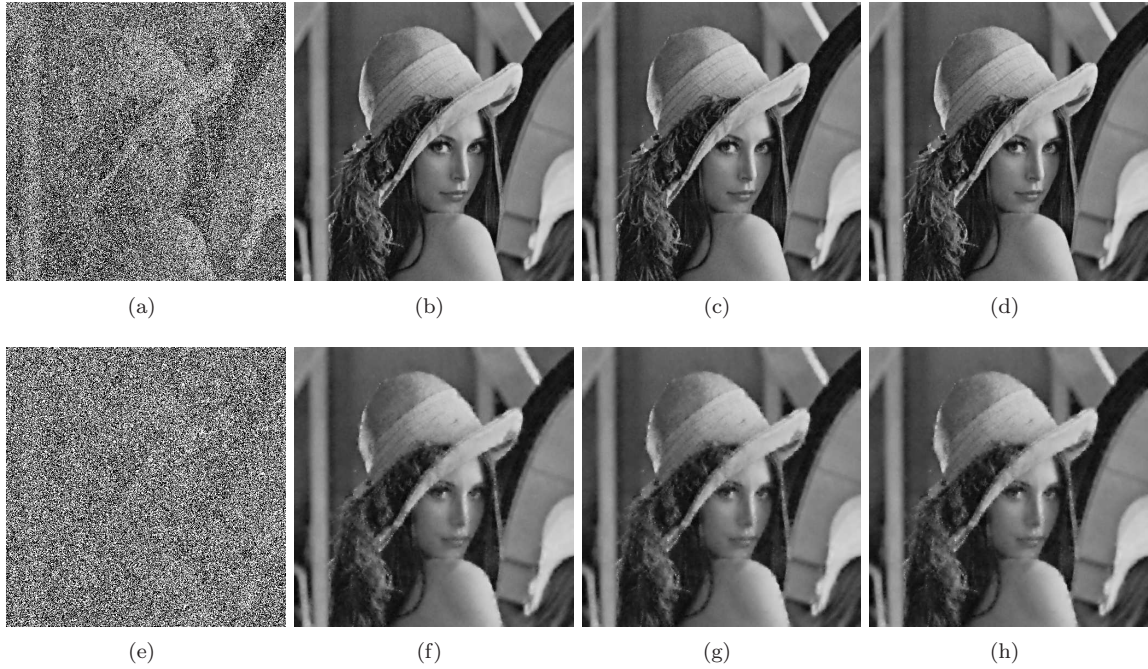
Figure 3: Restoration of the image Lena via Algorithms 1, 2 and 3, where $\varphi_\alpha(t) = \sqrt{t^2 + 100}$. From left to right: the noisy image and the restorations obtained by minimizing $\mathcal{G}$ with the relaxation method (Algorithm 1), minimizing $\mathcal{F}$ with the conjugant gradient method (Algorithms 2), and minimizing $\mathcal{F}$ with the quasi-Newton method (Algorithm 3) respectively. On the first row, the noise ratio is 70%, and the PSNRs are 6.7dB, 29.7dB, 29.8dB and 29.8dB. On the second row, the noise ratio is 90%, and the PSNRs are 5.6dB, 25.4dB, 25.4dB and 25.4dB.

References

[1] J. Astola and P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering.* Boca Raton, CRC, 1997.

[2] M. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications to early vision," *International Journal of Computer Vision*, 19 (1996), pp. 57–91.

[3] A. Bortoletti, C. Di Fiore, S. Fanelli, and P. Zellini, "A new class of quasi-Newtonian methods for optimal learning in MLP-networks," *IEEE Transactions on Neural Networks*, 14 (2003), pp. 263–273.

[4] C. Bouman and K. Sauer, "On discontinuity-adaptive smoothness priors in computer vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17 (1995), pp. 576–586.

[5] A. Bovik, *Handbook of Image and Video Processing*, Academic Press, 2000.

[6] J. F. Cai, R. H. Chan, and B. Morini, "Minimization of an edge-preserving regularization functional by conjugate gradient type methods", *Image Processing Based on Partial Differential*
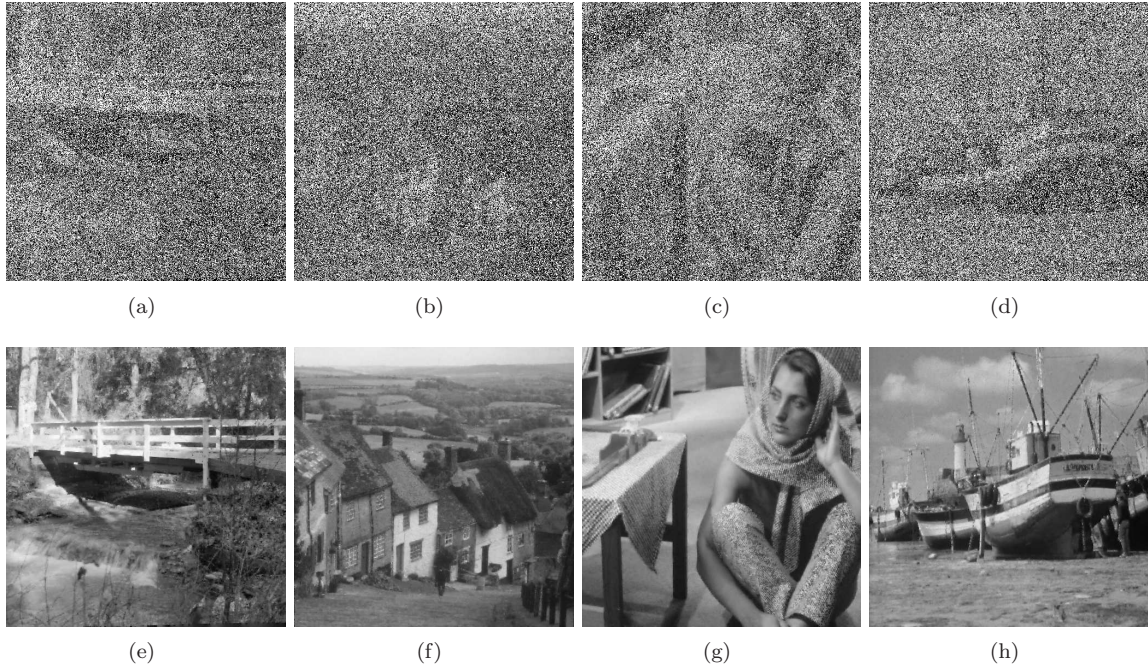
Figure 4: Restoration of the images (from left to right) *bridge*, *goldhill*, *barbara* and *boat* obtained by minimizing $\mathcal{F}$ with quasi-Newton method (Algorithm 3) and potential function $\sqrt{t^2 + 100}$. On the top row there are the noisy images corrupted by 70% salt-and-pepper noise, and on the bottom row there are the corresponding restorations. The PSNRs of the restored images are 24.9dB, 29.8dB, 24.7dB and 28.0dB respectively.

Equations: Proceedings of the International Conference on PDE-Based Image Processing and Related Inverse Problems, CMA, Oslo, August 8–12, 2005, Springer Berlin Heidelberg, 2007, pp. 109–122.

[7] R. H. Chan, C.-W. Ho, and M. Nikolova, "Convergence of Newton's method for a minimization problem in impulse noise removal", *Journal of Computational Mathematics*, 22 (2004), pp. 168–177.

[8] R. H. Chan, C. W. Ho, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detector and edge-preserving regularization," *IEEE Transactions on Image Processing*, 14 (2005), pp. 1479–1485.

[9] R. H. Chan, C. W. Ho, C. Y. Leung, and M. Nikolova, "Minimization of detail-preserving regularization functional by Newton's method with continuation," *Proceedings of IEEE International Conference on Image Processing*, Genova, Italy, 2005, pp. 125–128.

[10] R. H. Chan, C. Hu, and M. Nikolova, "An iterative procedure for removing random-valued impulse noise," *IEEE Signal Proc. Letters*, 11 (2004), pp. 921–924.

[11] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Transactions on Image Processing*, 6 (1997), pp. 298–311.

[12] T. Chen and H. R. Wu, "Space variant median filters for the restoration of impulse noise corrupted images," *IEEE Transactions on Circuits and Systems II*, 48 (2001), pp. 784–789.

[13] J. E. Dennis, and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations,* Prentice-Hall, Engle Cliffs: New Jersey, 1983.

[14] Y. Dong, R. Chan, and S. Xu, "A Detection statistic for random-valued impulse noise", *IEEE Trans. Image Proc.*, 16 (2007), 1112–1120.

[15] C. Di Fiore, S. Fanelli, F. Lepore, and P. Zellini, "Matrix algebras in quasi-Newton methods for unconstrained minimization," *Numerische Mathematik*, 94 (2003), pp. 479–500.

[16] C. Di Fiore, F. Lepore, and P. Zellini, "Hartley-type algebras in displacement and optimization strategies," *Linear Algebra and its Applications*, 366 (2003), pp. 215–232.

[17] P. J. Green, "Bayesian reconstructions from emission tomography data using a modified EM algorithm," *IEEE Transactions on Medical Imaging*, MI-9 (1990), pp. 84–93.

[18] T. S. Huang, G. J. Yang, and G. Y. Tang, "Fast two-dimensional median filtering algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1 (1979), pp. 13–18.

[19] H. Hwang and R. A. Haddad, "Adaptive median filters: new algorithms and results," *IEEE Transactions on Image Processing*, 4 (1995), pp. 499–502.

[20] S.-J. Ko and Y. H. Lee, "Center weighted median filters and their applications to image enhancement," *IEEE Transactions on Circuits and Systems*, 38 (1991), pp. 984–993.

[21] S. Leung and S. Osher, "Global minimization of the active contour model with TV-inpainting and two-phase denoising," *Proceeding of the 3rd IEEE workshop on Variational, Geometric and Level Set Methods in Computer Vision*, 2005, pp. 149–160.

[22] M. K. Ng, R. H. Chan, and W. C. Tang, "A fast algorithm for deblurring models with Neumann boundary conditions," *SIAM Journal on Scientific Computing*, 21 (2000), pp. 851–866.

[23] M. Nikolova, "Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers," *SIAM Journal on Numerical Analysis*, 40 (2002), pp. 965–994.

[24] M. Nikolova, "A variational approach to remove outliers and impulse noise," *Journal of Mathematical Imaging and Vision*, 20 (2004), pp. 99–120.

[25] J. Nocedal and S. J. Wright, *Numerical Optimization,* Springer, 1999.

[26] T. A. Nodes and N. C. Gallagher, Jr. "The output distribution of median type filters," *IEEE Transactions on Communications*, COM-32, 1984.

[27] G. Pok, J.-C. Liu, and A. S. Nair, "Selective removal of impulse noise based on homogeneity level information," *IEEE Transactions on Image Processing*, 12 (2003), pp. 85–92.

[28] J. Sun and J. Zhang, "Global convergence of conjugate gradient methods without line search," *Annals of Operations Research*, vol. 103 (2001), pp. 161–173

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