Signal Processing 90 (2010) 2891–2897

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Fast communication

Impulse noise removal by a nonmonotone adaptive gradient method ☆

Gaohang Yu [a,b,*], Liqun Qi [b], Yimin Sun [c], Yi Zhou [d]

[a] Jiangxi Key Laboratory of Numerical Simulation Technology, School of Mathematics and Computer Sciences, GanNan Normal University, Ganzhou 341000, China
[b] Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong
[c] School of Mathematics and Computational Sciences, Sun Yat-Sen University, China
[d] Department of Biomedical Engineering, Zhongshan School of Medicine, Sun Yat-Sen University, China

ARTICLE INFO

Article history:
Received 6 April 2009
Received in revised form 10 April 2010
Accepted 17 April 2010
Available online 22 April 2010

Keywords:
Image denoising
Impulse noise
Optimization
Adaptive gradient method

ABSTRACT

Image denoising is a fundamental problem in image processing. This paper proposes a nonmonotone adaptive gradient method (NAGM) for impulse noise removal. The NAGM is a low-complexity method and its global convergence can be established. Numerical results illustrate the efficiency of the NAGM and indicate that such a nonmonotone method is more suitable to solve some large-scale signal processing problems.

1. Introduction

Images are often corrupted by impulse noise in which the noisy pixels are assumed to be randomly distributed in the image. An important characteristic of impulse noise is that only part of the pixels are contaminated by the noise and the rest are free. There are two common types of impulse noise: one is the salt-and-pepper noise and the other is the random-valued impulse noise. For images corrupted by salt-and-pepper noise (respectively, random-valued noise), the noisy pixels can take only the maximal and minimal pixel values (respectively, any random value) in the dynamic. The goal of noise removal is to suppress the noise while preserving image details. The median filter was once the main method for removing impulse noise [1]. Over the years, several improved methods for impulse noise removal with different noise detectors were proposed, for example, the adaptive median filter (AMF) [2] and adaptive center-weighted median filter (ACWMF) [3], etc. These nonlinear filters can detect the noisy pixels even at a high noise level. However, they cannot 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 two-phase method was proposed in [4,5]. The first phase is the detection of the noise pixels by using the adaptive median filter (AMF) [2] for salt-and-pepper noise while for random-valued noise, it is accomplished by using the adaptive center-weighted median filter (ACWMF) [3]. Let the true image denoted by X, and $\mathcal{A}=\{1,2,3,\ldots,M\}\times\{1,2,3,\ldots,N\}$ be the index set of X. Let $\mathcal{N}\subset\mathcal{A}$ denote the set of indices of the noise pixels detected in the first phase. Let $\mathcal{V}_{i,j}$ denote the set of the four closest neighbors of the pixel at position $(i,j)\in\mathcal{A}$ and $y_{i,j}$ be the observed pixel value of the image at position

☆ These authors contributed equally to the paper.

* Corresponding author at: Jiangxi Key Laboratory of Numerical Simulation Technology, School of Mathematics and Computer Sciences, GanNan Normal University, Ganzhou 341000, China.

E-mail addresses: maghyu@163.com (G. Yu), maqilq@polyu.edu.hk (L. Qi), sunyimin@amss.ac.cn (Y. Sun), zhouyi@mail.sysu.edu.cn (Y. Zhou).

0165-1684/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2010.04.017

(i,j), and $u=[u_{i,j}]_{(i,j)\in\mathcal{N}}$ denote a column vector of length c ordered lexicographically. Here c is the number of elements of $\mathcal{N}$. Let φ_α be an edge-preserving functional and set $S^1_{i,j}=\sum_{(m,n)\in\mathcal{V}_{i,j}\setminus\mathcal{N}}\varphi_\alpha(u_{i,j}-y_{m,n})$, $S^2_{i,j}=\sum_{(m,n)\in\mathcal{V}_{i,j}\cap\mathcal{N}}\varphi_\alpha(u_{i,j}-u_{m,n})$. Then, the second phase is the recovering of the noise pixels by minimizing the following functional:

$$\mathcal{G}_\alpha(u)=\sum_{(i,j)\in\mathcal{N}}|u_{i,j}-y_{i,j}|+\frac{\tau}{2}\sum_{(i,j)\in\mathcal{N}}(2\cdot S^1_{i,j}+S^2_{i,j}), \quad (1)$$

where first summation is a data-fitting term and the second summation is a regularization term, and $\tau>0$ is a parameter. An example of edge-preserving potential function is $\varphi_\alpha(t)=\sqrt{\alpha+t^2},\alpha>0$, which corresponds to the popular smoothly approximated total variation (TV) regularization term. The explanation of the extra factor "2" in the second summation in (1) can be seen in [7].

The two-phase method can restore large patches of noisy pixels because it introduces pertinent prior information via the regularization term. However, the functional to be minimized in the second phase is nonsmooth, and it is costly to get the minimizer. The relaxation method in [4,5] is convergent but slow. To improve the computational efficiency, it was proposed in [6] to drop the nonsmooth data-fitting term, as it is not needed in the 2-phase method, where only noisy pixels are restored in the minimization. Therefore, there are a lot of optimization methods can be extended to minimize smooth edge-preserving regularization (EPR) functional. A Newton method was proposed in [6], a quasi-Newton method was presented in [7] and a class of conjugate gradient methods were considered in [7,8] to minimize the following smooth functional:

$$\mathcal{G}_\alpha(u)=\sum_{(i,j)\in\mathcal{N}}(2\cdot S^1_{i,j}+S^2_{i,j}). \quad (2)$$

This paper proposes an effective nonmonotone method to solve the above minimization problem. Section 2 describes our globally convergent nonmonotone adaptive gradient method (NAGM) in detail. Section 3 gives numerical results to illustrate the convergence and efficiency of the proposed method. Finally we have a conclusion section.

2. Adaptive gradient method

Given a starting point u_0 and using the notation $g_k=\nabla f(u_k)$, the gradient methods for $\min_{u\in\Re^n}f(u)$ are defined by the iteration $u_{k+1}=u_k-t_kg_k$, $k=0,1,\ldots$, where the stepsize $t_k>0$ is determined through an appropriate selection rule. In the classical steepest descent (SD) method, the stepsize $t_k>0$ is obtained by minimizing the function $f(u)$ along the ray $\{u_k-tg_k : t>0\}$. In 1988, Barzilai and Borwein (BB) [9] developed an ingenious gradient method in which stepsize $t_k(k>0)$ is determined by

$$t_k^{BB_1}=\frac{s_{k-1}^Ts_{k-1}}{y_{k-1}^Ts_{k-1}} \quad \text{or} \quad t_k^{BB_2}=\frac{y_{k-1}^Ts_{k-1}}{y_{k-1}^Ty_{k-1}},$$

where $s_{k-1}=u_k-u_{k-1}$ and $y_{k-1}=g_k-g_{k-1}$. In fact, t_k is derived from an approximately secant equation: $t_k^{BB_1}=\text{argmin}_{t\in\Re}\|(1/t)s_{k-1}-y_{k-1}\|_2$ and $t_k^{BB_2}=\text{argmin}_{t\in\Re}\|s_{k-1}-ty_{k-1}\|_2$. It is clear that $t_k^{BB_1}\geq t_k^{BB_2}$. The BB method performs much better than the SD method in practice. Especially, when the objective function is convex quadratic function and n=2, the BB method converges R-superlinearly to the global minimizer [9]. For any dimension convex quadratic function, it is still globally convergent [10] but the convergence is R-linear [11]. In the last years, stepsize selection rules in gradient methods have received an increasing interest from both the theoretical and the practical point of view [12,13]. In [12], Zhou et al. proposed a gradient method with an adaptive stepsize:

$$t_k=\begin{cases} t_k^{BB_2} & \text{if } \dfrac{t_k^{BB_2}}{t_k^{BB_1}}<\kappa, \\ t_k^{BB_1} & \text{otherwise,}\end{cases} \quad (3)$$

where $\kappa\in(0,1)$ is a constant. They interpret (3) as follows: If the previous iterate u_k is a bad point (e.g. when $t_k^{BB_2}/t_k^{BB_1}<0.15$) for the minimal gradient method, and so that there is little reduction in $\|g(u)\|^2$, choose the smaller stepsize $t_k^{BB_2}$; otherwise, choose the larger stepsize $t_k^{BB_1}$. Their numerical results show that this adaptive stepsize can improve its practical performance.

In this paper, we consider a new gradient method which adaptively choose a small stepsize or a large stepsize at each iteration such as

$$t_k=\begin{cases} t_k^{BB_1} & \text{if } k \text{ is odd or } \dfrac{s_{k-1}^Ty_{k-1}}{\|s_{k-1}\|\|y_{k-1}\|}>\beta, \\ t_k^{BB_2} & \text{otherwise,}\end{cases} \quad (4)$$

where $\beta<1$ is close to 1. It is easy to derive that $t_k^{BB_2}/t_k^{BB_1}=(s_{k-1}^Ty_{k-1}/\|s_{k-1}\|\|y_{k-1}\|)^2$. Even then, the switch condition in (4) is different to that in (3). And we can interpret Eq. (4) in a different viewpoint. Let $H_k=\nabla^2f(u_k)$. We know that $s_{k-1}^Ty_{k-1}/\|s_{k-1}\|\|y_{k-1}\|\approx g_k^TH_kg_k/\|g_k\|\cdot\|H_kg_k\|$. When $g_k^TH_kg_k/\|g_k\|\cdot\|H_kg_k\|\approx1$, the gradient g_k can be regarded as a good approximation of an eigenvector of the Hessian matrix H_k. In this case, it is reasonable to choose $t_k=t_k^{BB_1}$ since $1/t_k^{BB_1}$ is a good approximation to the corresponding eigenvalue. In addition, this adaptive gradient method might overcome the drawbacks of the SD method in an efficient manner. Here the small stepsize $t_k^{BB_2}$ is primarily used to induce a favorable descent direction for the next iteration, while the large stepsize $t_k^{BB_1}$ is primarily used to produce a sufficient reduction (see for instance [12] for more mathematical analysis of this problem).

In order to ensure global convergence for the BB-like method, it is necessary to modify the BB-like method by incorporating some sort of line search [13,14]. In [14], Dai and Fletcher proposed an adaptive nonmonotone line search. The numerical results reported in [14] show that this kind of line search is particularly suitable for the BB method in the nonquadratic case. The method has a reference function value f_r, and each iteration must improve on the reference value such that

$$f(u_k+\lambda d_k)\leq f_r+\theta\lambda g_k^Td_k, \quad (5)$$

where $d_k=-t_k\,g_k$ denotes the current search direction, $\theta\in(0,1)$ is a given constant and $\lambda>0$ is the tried stepsize.