A sufficient descent conjugate gradient method for impulse noise removal

Peiting Gao, Yongfei Wu*

College of Data Science, Taiyuan University of Technology, Shanxi 030024 China

*Corresponding author, e-mail: wuyongfei@tyut.edu.cn

ABSTRACT: In this paper, we introduce a modified PRP-type conjugate gradient (CG) method for impulse noise removal in the second phase of the two-phase method. A nice property of the scheme is that the search direction at each iteration satisfies the sufficient descent condition independent of any line search. Under the Armijo-type line search, its global convergence result is proved. Numerical comparison is given to illustrate that the proposed method for removing impulse noise is promising.

KEYWORDS: image processing, variational method, two-phase method, conjugate gradient method, global convergence

INTRODUCTION

Image processing is a highly diverse field which includes subfields such as image recognition, image segmentation, and image denoising [1, 2]. Image denoising is a typical inverse problem and is hard to be solved. A common type of noise in images is the impulse noise which can be further categorized the two types: salt-and-pepper noise, for which the noisy pixels can take only the maximal and minimal pixel values, and the random-valued noise, for which the noisy pixels can take any random values between the maximal and minimal pixel values.

Numerous methods for restoring images corrupted by impulse noise have been proposed in past years. Among these methods, two methods are the most popular among research activities. One is the median filter and its several remedies [3–5] which are based on nonlinear digital filters [6]. However, its performance is not good when the noise level is high since they always fail to obtain local image features such as the possible presence of edges. Another is variational method [7], which is depended on variational framework and is capable of preserving the details and the edges. But, these methods change the gray level of each pixel including uncorrupted ones.

In order to avoid the disadvantages of the methods mentioned above, Chan et al [8] proposed a two-phase method based on the adaptive median filter method (AMF) [3] and the variational method [7, 9, 10]. More precisely, the noise pixels are first detected by using AMF method and then they are restored by minimizing an objective function $\mathcal{G}_u$ with an $\ell_1$ data-fitting term and a regularization term involving an edge-preserving potential function $\phi_u(t)$ [8]. The objective function $\mathcal{G}_u$ is computed as follows:

$$
\mathcal{G}_u(u) = \sum_{(i,j) \in \mathcal{N}} \left\{ |u_{i,j} - y_{i,j}| + \frac{\beta}{2} \sum_{(m,n) \in \mathcal{N}_{i,j},\cdot} 2\phi_u(u_{i,j} - y_{m,n}) \right\} + \frac{\beta}{2} \sum_{(i,j) \in \mathcal{N}} \left\{ \sum_{(m,n) \in \mathcal{N}_{i,j},\cdot} \phi_u(u_{i,j} - u_{m,n}) \right\},
$$

(1)

where $\mathcal{N} \subset \mathcal{A}$ is the set of the noisy candidates, which are detected by AMF method in the first phase, $\mathcal{A} = \{1, 2, 3, \ldots, M\} \times \{1, 2, 3, \ldots, N\}$ is the index set of $X$ denoting the original image with $M$-by-$N$ pixels, $\mathcal{N}_{i,j}$ is the set of the four closest neighbors of the pixels at the position $(i, j) \in \mathcal{A}$, $y_{i,j}$ is the observed pixel value of the image at the position $(i, j)$, $\beta$ is the regularization parameter, $u = [u_{i,j}]_{i,j \in \mathcal{N}}$ is a column vector of length $c$ ordered lexicographically with $c$ denoting the number of elements of $\mathcal{N}$, and $\phi_u$ is an edge-preserving function.

Because of introducing the regularization term involving pertinent prior information, the two-phase method can preserve the details and the edges of the image and unchange the uncorrupted pixels. However, the objective function $\mathcal{G}_u$ to be minimized is nonsmooth as it includes a nonsmooth $\ell_1$ data-fitting term, and so it is destined to the high cost of getting the minimizer. In order to overcome the drawbacks, Chan et al [11] proved that the nonsmooth $\ell_1$ data-fitting term can be dropped because it is useless in the second phase. The objective function $\mathcal{G}_u(1)$ is converted to $\mathcal{F}_u(2)$ as follows:

$$
\mathcal{F}_u(u) = \sum_{(i,j) \in \mathcal{N}} \left\{ \sum_{(m,n) \in \mathcal{N}_{i,j},\cdot} \phi_u(u_{i,j} - y_{m,n}) \right\} + \frac{1}{2} \sum_{(i,j) \in \mathcal{N}} \left\{ \sum_{(m,n) \in \mathcal{N}_{i,j},\cdot} \phi_u(u_{i,j} - u_{m,n}) \right\}.
$$

(2)
Despite minimizing $\mathcal{F}_a$ instead of $\mathcal{G}_a$ in the second phase, the quality of the restored images is not affected. Some related results are showed in [12, 13]. Thus, the 2-phase method for impulse noise removal can be viewed as a minimization unconstrained optimization problem of the new objective function $\mathcal{F}_a$. Numerous methods have been proposed by authors to minimize $\mathcal{F}_a$. Besides, Yin et al. [14] recently proposed a generalized hybrid conjugate gradient projection method-based algorithm to restore image by solving large-scale convex constrained equations.

In this section, we first review a general formula of CG method outperforms other competitors to remove salt-and-pepper noise. Preliminary numerical results show that the proposed modified PRP-type CG method for solving unconstrained optimization problems involving the modified PRP CG method as follows:

$$d_k = -g_k + \beta_k^mPRP d_{k-1} + \theta_k y_{k-1},$$

(4)

where

$$\beta_k^mPRP = \frac{1}{\|y_{k-1}\|^2} \left( y_{k-1} - \frac{\|y_{k-1}\|^2 d_{k-1}}{\|y_{k-1}\|^2} \right)^\top g_k,$$

$$\theta_k = -\frac{g_k^\top d_{k-1}}{\|y_{k-1}\|^2},$$

and $r > 0$. They proved that this modified CG method satisfies the sufficient descent condition with $r > 0$. Observe that if $r < 0$, this search direction may generate an ascent direction.

Motivated by the above articles, we consider a modified PRP CG method such that the search direction is generated by the following way:

$$d_k = -g_k + \beta_k^mPRP d_{k-1} - \beta_k^mPRP g_k^\top d_{k-1} \|g_k\|^2 g_k,$$

(5)

Notice that such defined direction $d_k$ satisfies the sufficient descent condition at each iteration without $r > 0$. Obviously,

$$g_k^\top d_k = -\|g_k\|^2.$$  

(6)

Furthermore, we obtain that

$$\|g_k\| \leq \|d_k\|.$$  

Next, we list the detailed process of Algorithm 1 based on the above analysis.

**Algorithm 1**

**Step 0.** Given an initial point $x_0 \in \mathbb{R}^n$, and $0 < \rho < 1$, $\sigma > 0$, and $\tau > 0$. Set $k = 0$.

**Step 1.** If the stopping criteria holds, stop. Otherwise go to Step 2.

**Step 2.** Compute the direction $d_k$ as follows:
If $k = 0$, then $d_k = -g_k$.

If $k > 0$, then

$$d_k = -g_k + \beta_k^{\text{mPP}} d_{k-1} - \beta_k^{\text{mPP}} g_k^T g_k^{-1} g_k,$$

where

$$\beta_k^{\text{mPP}} = \frac{1}{\|g_k\|} \left( y_{k-1} - \frac{r\|y_{k-1}\|^2 d_{k-1}}{\|g_k\|^2} \right)^T g_k,$$ (7)

and $r$ is a constant.

Step 3. Set $x_{k+1} = x_k + \alpha_k d_k$, when the step-size $\alpha_k$ is computed by using the Armijo line search, that is $\alpha_k = t_k \rho_k$, with $t_k$ being the smallest nonnegative integer such that

$$-f(x_k + \alpha_k d_k) \leq f(x_k) - \sigma \alpha_k^2 \|d_k\|^2,$$ (8)

where $t_k = \min\{\tau \|g_k\|^2 / \|d_k\|^2, 1\}$.

Step 4. Set $k := k + 1$, go to Step 1.

GLOBAL CONVERGENCE ANALYSIS

In this section, the global convergence of Algorithm 1 is established and the following assumption is needed.

Assumption 1

(i) The level set $\Lambda = \{X \mid f(X) \leq f(X_0)\}$ is bounded.

(ii) In some neighborhood $\tilde{N}$ of $O$, the objective function $f$ is continuously differentiable and its gradient $g$ is Lipschitz continuous in $\tilde{N}$, i.e., there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L \|x - y\|, \quad \forall x, y \in \tilde{N},$$

which imply that there exists a positive constant $\gamma$ so that

$$\|g(x)\| \leq \gamma, \quad \forall x \in \tilde{N}.$$

Lemma 1 Let Assumption 1 hold, and the sequence $\{d_k\}$ be generated by Algorithm 1, then the sequence $\{\|d_k\|\}$ is bounded.

Proof: By Eq. (7), we have

$$\|d_k\| \leq \|\|d_k\| + 2\|\beta_k^{\text{mPP}}\| \|d_{k-1}\|$$

By (9) and the Assumption 1, we obtain

$$\|d_k\| \leq \|g_k\| + 2\|\beta_k^{\text{mPP}}\| \|d_{k-1}\|$$

and $r$ is a constant.

Step 3. Set $x_{k+1} = x_k + \alpha_k d_k$, when the step-size $\alpha_k$ is computed by using the Armijo line search, that is $\alpha_k = t_k \rho_k$, with $t_k$ being the smallest nonnegative integer such that

$$-f(x_k + \alpha_k d_k) \leq f(x_k) - \sigma \alpha_k^2 \|d_k\|^2,$$ (8)

where $t_k = \min\{\tau \|g_k\|^2 / \|d_k\|^2, 1\}$.

Step 4. Set $k := k + 1$, go to Step 1.

GLOBAL CONVERGENCE ANALYSIS

In this section, the global convergence of Algorithm 1 is established and the following assumption is needed.

Assumption 1

(i) The level set $\Lambda = \{X \mid f(X) \leq f(X_0)\}$ is bounded.

(ii) In some neighborhood $\tilde{N}$ of $O$, the objective function $f$ is continuously differentiable and its gradient $g$ is Lipschitz continuous in $\tilde{N}$, i.e., there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L \|x - y\|, \quad \forall x, y \in \tilde{N},$$

which imply that there exists a positive constant $\gamma$ so that

$$\|g(x)\| \leq \gamma, \quad \forall x \in \tilde{N}.$$

Lemma 1 Let Assumption 1 hold, and the sequence $\{d_k\}$ be generated by Algorithm 1, then the sequence $\{\|d_k\|\}$ is bounded.

Proof: By Eq. (7), we have

$$\|d_k\| \leq \|\|d_k\| + 2\|\beta_k^{\text{mPP}}\| \|d_{k-1}\|$$

By (9) and the Assumption 1, we obtain

$$\|d_k\| \leq \|g_k\| + 2\|\beta_k^{\text{mPP}}\| \|d_{k-1}\|$$

and $r$ is a constant.

Step 3. Set $x_{k+1} = x_k + \alpha_k d_k$, when the step-size $\alpha_k$ is computed by using the Armijo line search, that is $\alpha_k = t_k \rho_k$, with $t_k$ being the smallest nonnegative integer such that

$$-f(x_k + \alpha_k d_k) \leq f(x_k) - \sigma \alpha_k^2 \|d_k\|^2,$$ (8)

where $t_k = \min\{\tau \|g_k\|^2 / \|d_k\|^2, 1\}$.

Step 4. Set $k := k + 1$, go to Step 1.

GLOBAL CONVERGENCE ANALYSIS

In this section, the global convergence of Algorithm 1 is established and the following assumption is needed.

Assumption 1

(i) The level set $\Lambda = \{X \mid f(X) \leq f(X_0)\}$ is bounded.

(ii) In some neighborhood $\tilde{N}$ of $O$, the objective function $f$ is continuously differentiable and its gradient $g$ is Lipschitz continuous in $\tilde{N}$, i.e., there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L \|x - y\|, \quad \forall x, y \in \tilde{N},$$

which imply that there exists a positive constant $\gamma$ so that

$$\|g(x)\| \leq \gamma, \quad \forall x \in \tilde{N}.$$

Lemma 1 Let Assumption 1 hold, and the sequence $\{d_k\}$ be generated by Algorithm 1, then the sequence $\{\|d_k\|\}$ is bounded.

Proof: By Eq. (7), we have

$$\|d_k\| \leq \|\|d_k\| + 2\|\beta_k^{\text{mPP}}\| \|d_{k-1}\|$$

By (9) and the Assumption 1, we obtain

$$\|d_k\| \leq \|g_k\| + 2\|\beta_k^{\text{mPP}}\| \|d_{k-1}\|$$

and $r$ is a constant.
From the definition of $a_k$, we can find that $\rho^{-1}a_k$ does not satisfy the line search and the inequality holds as follows:

$$f(x_k + \rho^{-1}a_k d_k) > f(x_k) - \delta \rho^{-2}a_k^2\|d_k\|^2.$$  \hspace{1cm} (11)

By Assumption 1, we get that

$$f(x_k + \rho^{-1}a_k d_k) - f(x_k) = \int_0^1 g(x_k + t\rho^{-1}a_k d_k)^\top (\rho^{-1}a_k d_k) dt$$

$$= \int_0^1 [g(x_k + t\rho^{-1}a_k d_k) - g(x_k)]^\top (\rho^{-1}a_k d_k) dt$$

$$+ \rho^{-1}a_k g_k^\top d_k$$

$$\leq \frac{1}{2} \rho^{-2}a_k^2 \|d_k\|^2 + \rho^{-1}a_k g_k^\top d_k$$

$$= \frac{1}{2} \rho^{-2}a_k^2 \|d_k\|^2 - \rho^{-1}a_k \|g_k\|^2.$$  \hspace{1cm} (12)

Combining (12) with (6) and (11), we obtain that

$$\|g_k\|^2 \leq \rho^{-1}\left(\frac{1}{2} \rho^{-2} + \delta\right) a_k \|d_k\|^2.$$  

Due to Lemma 1, Lemma 2 and

$$\lim_{k \to \infty} \alpha_k = 0,$$

we have

$$\lim_{k \to \infty} \|g_k\| = 0.$$  

To sum up, the conclusion holds.

**Remark 1** According to [13, Proposition 1], the $F_\alpha$ is smooth when $\varphi_\alpha$ is smooth. In this paper, the edge-preserving potential function is defined as the Huber function:

$$\varphi_\alpha = \begin{cases} \frac{\alpha^2}{2 \alpha}, & \text{if } |t| \leq \alpha, \\ |t| - \frac{\alpha^2}{2}, & \text{otherwise}, \end{cases}$$  \hspace{1cm} (13)

where $\alpha > 0$. Obviously, $F_\alpha$ is continuously differentiable, convex, and its gradient $\nabla F_\alpha$ is first order Lipschitz continuous. Furthermore, the global minimum of $F_\alpha$ exists. Interested readers may refer to [9, Propositions 1–3]. Hence, Algorithm 1 can be used for restoring the images corrupted by impulse noise, which has some nice features, such as simplicity, sufficient descent direction without any restriction, and global convergence.

**Remark 2** To minimize $F_\alpha$ by Algorithm 1, we view $u$ as a column vector of length $c$ lexicographically, in which $c$ is the number of elements of $sN$. In fact, $F_\alpha$ can be viewed as a special case of the problem (3), thus, Algorithm 1 proposed for the problem (3) is reasonable.

**NUMERICAL PERFORMANCE**

In this section, we presented the results of a series of experiments to illustrate the performance of Algorithm 1 for removing the salt-and-pepper impulse noise and compare its performance with existing state-of-the-art algorithm, including the three-term PRP conjugate method (referred to as TPRP) [9]. The edge-preserving potential function $\varphi(s)$ is the Huber function (13) with $\alpha = 10$. All experiments were implemented under MATLAB (Version 2017b) environment and run on a PC with 2.30 GHz CPU processor and 8.0 GB memory.

In the experiments, we choose Lena (256 × 256), Cameraman (256 × 256), Barbara (512 × 512) and Baboon (512 × 512) as the test images. Herein, the peak signal noise ratio (PSNR) [20] is calculated to quantify image quality upgradation after denoising procedure, and the formula is defined as

$$\text{PSNR} = 10 \log_{10} \frac{(N - 1)^2}{\sum_{i,j} (u_{ij} - x_{ij})^2}.$$  

In order to compare the effectiveness of the test method fairly, the line search and the parameters involved for competing method is from the original paper. In this paper, the parameter in Algorithm 1 are chosen by testing the best PSNR values as follows: $\sigma = 0.5$, $\rho = 0.5$ and $\tau = \frac{\sqrt{2}}{8}$. The stopping criteria of the minimization are

$$\frac{|F(u_k) - F(u_{k-1})|}{F(u_k)} \leq \epsilon \quad \text{or} \quad \frac{\|u_k - u_{k-1}\|}{\|u_k\|} \leq \epsilon,$$

where $\epsilon = 10^{-4}$.

In this section, the numerical comparison were conducted from two aspects. Firstly, we discuss the parameter $r$ to Algorithm 1 to obtain an acceptable image restoration results. Therefore, we choose different $r$ to test the effectiveness of Algorithm 1. The detailed numerical results are summarized in Table 1, which is presented in the form “NI/CPU/PSNR”. “Niter” stands for the number of iterations, “CPU” stands for the CPU time required for the whole image denoising process, and “NL” stands for the noise level. Table 1 shows that Algorithm 1 performs best when the parameter $r = 0$. Secondly, we investigate the computational efficiency of TPRP method. The experiments are showed for the average results from five different noise samples of each image at each noise level. The results are listed in Table 1. From Table 1, it is not difficult to obtain that Algorithm 1 can successfully remove noise with $(r = -1)$, and TPRP method $(r \geq 0)$ cannot be used to deal with it. Then from Table 1, data format in Table 2 is consistent with that in Table 1. “Niter”, “CPU” and “PSNR” all stand for the average of the corresponding data in Table 2. Table 2 indicates that Algorithm 1 (r = 0) has absolute advantages in the
Table 1  The results of the salt-and pepper noise removal via Algorithm 1 with different $r$.

<table>
<thead>
<tr>
<th>Image</th>
<th>NL</th>
<th>$r = -1$</th>
<th>$r = 0$</th>
<th>$r = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NI/CPU/PSNR</td>
<td>NI/CPU/PSNR</td>
<td>NI/CPU/PSNR</td>
<td></td>
</tr>
<tr>
<td>Lena</td>
<td>0.3</td>
<td>38/0.9442/39.5604</td>
<td>33/0.86/39.4508</td>
<td>37/0.9340/39.3387</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>53/1.681/36.2405</td>
<td>50/1.5253/36.1649</td>
<td>53/1.6006/36.0527</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>72/2.1859/32.9788</td>
<td>69/2.1123/33.1216</td>
<td>80/2.1132/33.0879</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>163/3.1796/28.7250</td>
<td>148/2.8831/28.0709</td>
<td>163/3.1784/28.9564</td>
</tr>
<tr>
<td>Cameraman</td>
<td>0.3</td>
<td>58/1.3671/36.7560</td>
<td>54/1.2829/36.6279</td>
<td>56/1.3159/36.4710</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>70/2.0705/33.4324</td>
<td>68/2.0121/33.3550</td>
<td>68/2.0124/33.0973</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>94/2.5934/30.6907</td>
<td>93/2.4613/31.0721</td>
<td>96/2.5073/30.6286</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>179/3.6216/27.2220</td>
<td>178/3.6011/27.3443</td>
<td>188/3.5604/27.2268</td>
</tr>
<tr>
<td>Barbara</td>
<td>0.3</td>
<td>38/3.7640/35.0806</td>
<td>37/3.6570/35.1309</td>
<td>38/3.7061/35.1333</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>49/6.8907/32.4663</td>
<td>46/6.5241/32.4430</td>
<td>47/6.5971/32.4321</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>65/9.2749/30.6253</td>
<td>62/8.6864/30.6137</td>
<td>64/8.8865/30.5832</td>
</tr>
<tr>
<td>Baboon</td>
<td>0.3</td>
<td>41/4.0800/33.4618</td>
<td>41/3.9914/33.4610</td>
<td>41/4.1958/33.4452</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>52/7.2834/30.6219</td>
<td>51/7.2434/30.6342</td>
<td>53/7.4044/30.6004</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>124/19.9708/26.3612</td>
<td>120/18.2438/26.3470</td>
<td>127/18.9068/26.3780</td>
</tr>
</tbody>
</table>

Table 2  The results of the salt-and pepper noise removal via TPRP method Algorithm 1 ($r = 0$).

<table>
<thead>
<tr>
<th>Image</th>
<th>NL</th>
<th>TPRP method</th>
<th>Algorithm 1 ($r = 0$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NI/CPU/PSNR</td>
<td>NI/CPU/PSNR</td>
<td>NI/CPU/PSNR</td>
</tr>
<tr>
<td>Lena</td>
<td>0.3</td>
<td>44/1.0767/39.1855</td>
<td>33/0.8764/39.2185</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>50/1.5559/36.1226</td>
<td>46/1.4669/36.0134</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>84/2.2515/33.0647</td>
<td>73/1.9696/33.1343</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>153/3.0341/28.4516</td>
<td>139/2.8404/28.7287</td>
</tr>
<tr>
<td>Cameraman</td>
<td>0.3</td>
<td>62/1.4605/36.9490</td>
<td>52/1.2778/36.5752</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>72/2.1133/33.6533</td>
<td>68/2.0466/33.5211</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>97/2.5047/30.4425</td>
<td>88/2.5140/30.7138</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>176/3.3631/27.1365</td>
<td>147/3.0963/27.2260</td>
</tr>
<tr>
<td>Barbara</td>
<td>0.3</td>
<td>39/4.1143/35.1078</td>
<td>37/3.9403/35.0755</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>49/7.7791/32.4582</td>
<td>47/7.5141/32.4067</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>64/10.0530/30.6405</td>
<td>62/10.1970/30.6096</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>125/18.5569/28.3923</td>
<td>120/17.3208/28.5258</td>
</tr>
<tr>
<td>Baboon</td>
<td>0.3</td>
<td>42/4.6022/33.4330</td>
<td>40/4.2608/33.4442</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>52/7.9431/30.5748</td>
<td>49/7.1642/30.6334</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>71/11.3374/28.4237</td>
<td>66/9.3323/28.4001</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>125/18.2859/26.3308</td>
<td>120/18.6282/26.3493</td>
</tr>
</tbody>
</table>

number of iterates and the CPU time, and it performs best in removing the salt-and-pepper impulse noise for most cases. Fig. 1 gives the flow chart of our experiments. Fig. 2 gives that the restoration results via TPRP method and Algorithm 1 ($r = 0$) for the test images corrupted with 0.7 salt-and-pepper noise.

CONCLUSION

In this paper, we propose an efficient conjugate gradient method for impulse noise removal. An attractive feature of the proposed method is that the search direction obtained satisfies the sufficient descent condition regardless of any restriction. And its global convergence is proved under Armijo-type line search. Numerical comparison is given to illustrate that the proposed method ($r = 0$) for removing impulse noise is promising and robust. Furthermore, in the Numerical comparison section, we successfully use Algorithm 1 to remove noise with ($r = -1$), and TPRP method ($r \geq 0$) cannot be used to deal with it. In this paper, $r$ is a constant and we try to construct different $r$ to satisfy different conditions to improve computational efficiency in future.

Acknowledgements: This research was supported by the National Science Foundation of Shanxi Province, China (No. 202203021212255) and the National Natural Foundation of China (No. 61901292).

REFERENCES

8. Chan RH, Ho CW, Nikolova M (2005) Salt-and-pepper noise removal by median-type noise detector and edge-
Fig. 1 Flow chart of our experiments.

Fig. 2 From top to bottom: original images, noisy images with 0.7 the salt-and-pepper noise, the restorations obtained by minimizing $F_a$ with TPRP method and Algorithm 1, respectively.


