

# RESTORATION OF IMAGES CORRUPTED BY IMPULSE NOISE AND MIXED GAUSSIAN IMPULSE NOISE USING BLIND INPAINTING

MING YAN\*

**Abstract.** This article studies the problem of image restoration of observed images corrupted by impulse noise and mixed Gaussian impulse noise. Since the pixels damaged by impulse noise contain no information about the true image, how to find this set correctly is a very important problem. We propose two methods based on blind inpainting and  $\ell_0$  minimization that can simultaneously find the damaged pixels and restore the image. By iteratively restoring the image and updating the set of damaged pixels, these methods have better performance than other methods, as shown in the experiments. In addition, we provide convergence analysis for these methods, these algorithms will converge to coordinatewise minimum points. In addition, they will converge to local minimum points (or with probability one) with some modifications in the algorithms.

**Key words.** impulse noise, mixed Gaussian impulse noise, total variation, blind inpainting, image restoration,  $\ell_0$  minimization

**1. Introduction.** Observed images are often corrupted by impulse noise during image acquisition and transmission, caused by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or bit errors in transmission [1]. There are two common types of impulse noise: salt-and-pepper impulse noise and random-valued impulse noise. Assume that the dynamic range of an image is  $[d_{\min}, d_{\max}]$ . For images corrupted by salt-and-pepper impulse noise, the noisy pixels can take only two values  $d_{\min}$  and  $d_{\max}$ , while for images corrupted by random-valued impulse noise, the noisy pixels can take any random value between  $d_{\min}$  and  $d_{\max}$ .

In this work, the original unknown  $M \times N$  image  $u$  is defined on a domain  $\Omega = \{(i, j) : i = 1, \dots, M, j = 1, \dots, N\}$ , and the observed  $M \times N$  image  $f$  is modeled as

$$f_{i,j} = \begin{cases} (Hu)_{i,j} + (n_1)_{i,j}, & (i, j) \in \Omega_1, \\ (n_2)_{i,j}, & (i, j) \in \Omega_1^c := \Omega \setminus \Omega_1. \end{cases} \quad (1.1)$$

Here,  $n_2$  is the impulse noise, and  $n_1$  is the additive zero-mean Gaussian white noise.  $H$  is the identity or a blurring operator, which is assumed to be continuous. The subset  $\Omega_1^c$  of  $\Omega$  denotes the region where the information of  $Hu$  is missing. The problem is to find the true image  $u$  from observed image  $f$  given the operator  $H$ .

If  $\Omega_1^c$  is empty, there is no impulse noise, then we have  $f = Hu + n_1$ , which is an image denoising (and deblurring) problem, and it has been extensively studied by both signal processing researchers and mathematicians. If  $\Omega_1^c$  is not empty and known, this can be considered as an image inpainting (and deblurring) problem.

Here, we will consider the last and most difficult case where  $\Omega_1^c$  is not empty and unknown. The challenge of this problem is to restore the lost details, and remove the impulse noise simultaneously. If  $n_1 = 0$ , this problem is an impulse noise removal (and deblurring) problem and if  $n_1 \neq 0$  it becomes a mixed Gaussian impulse noise removal (and deblurring) problem. There are already several types of approaches for solving these problems.

The first type of approaches treats  $n_2$  as outliers and uses the  $\ell_1$  norm in the fidelity term to increase the robustness of inpainting to outliers [2, 3, 4, 5], and the

---

\*Department of Mathematics, University of California, Los Angeles, CA, 90095 USA. E-mail: yanm@math.ucla.edu.

problem is to solve

$$\underset{u}{\text{minimize}} \sum_{i,j} |(Hu)_{i,j} - f_{i,j}| + \lambda_1 J(u), \quad (1.2)$$

where  $J(u)$  is a regularization on the true image  $u$ . There are many candidates for the regularization  $J(u)$ , and some examples are Tikhonov regularization [6], Geman and Reynolds' half quadratic variational models [7], Rudin, Osher and Fatemi's total variation models [8, 9], and framelet based models [10, 11]. This approach does not need to find the damaged pixels and performs well in impulse noise removal. However, for the case of images corrupted by mixed Gaussian impulse noise, the Gaussian noise is not treated properly.

The second type of approaches is the two-stage approach [12, 13, 14, 15, 11, 16, 17], which estimates the inpainting region  $\Omega_1^c$  before estimating  $u$ . In these approaches, the second stage becomes a regular image inpainting (and deblurring) problem [18, 19, 20, 21]

$$\underset{u}{\text{minimize}} \frac{1}{2} \sum_{(i,j) \in \Omega_1} ((Hu)_{i,j} - f_{i,j})^2 + \lambda_1 J(u). \quad (1.3)$$

The success of these two-stage approaches relies on the accurate detection of  $\Omega_1^c$ , e.g. adaptive median filter (AMF) [22] is used to detect salt-and-pepper impulse noise, while adaptive center-weighted median filter (ACWMF) [23] and rank-ordered logarithmic difference (ROLD) [24] are utilized to detect random-valued impulse noise.

Though adaptive median filter can detect most pixels damaged by salt-and-pepper impulse noise, it is more difficult to detect pixels corrupted by random-valued impulse noise than salt-and-pepper impulse noise. Recently, by considering two different types of noise, Dong et al. [25] proposed a new method using framelet to remove random-valued impulse noise plus Gaussian noise by solving

$$\underset{u,v}{\text{minimize}} \frac{1}{2} \sum_{i,j} ((Hu)_{i,j} + v_{i,j} - f_{i,j})^2 + \lambda_1 \|Wu\|_1 + \lambda_2 \sum_{i,j} |v_{i,j}|, \quad (1.4)$$

where  $W$  is a transformation from the image to the framelet coefficients. Two unknowns  $u$  (restored image) and  $v$  (noise) are introduced into this variational model, and their methods can simultaneously find  $u$  and  $v$  using split Bregman iterations [26].

Dong et al.'s method uses  $\ell_1$  norm as a convex approximation of  $\ell_0$  term to make the result  $v$  sparse, and keep the problem convex in the meantime. However, using non-convex optimization ( $\ell_p$  when  $p < 1$ ) has better performance than convex optimization in dealing with the sparsity, as shown in compressive sensing [27]. Even  $\ell_0$  minimization and smoothed  $\ell_0$  minimization are used in many algorithms [28, 29, 30, 31, 32, 33]. In this paper, we will use  $\ell_0$  minimization instead of  $\ell_1$  minimization in the problem, and by using  $\ell_0$  minimization, the problem of finding  $u$  can be solved by considering a problem of finding  $u$  and  $\Omega_1$ . In addition, using alternating minimization algorithm, it can be solved easily by alternately solving the image inpainting problem and finding the damaged pixels.

The work is organized as follows. In section 2 and 3, we introduce our general methods for removing impulse noise using two different treatments for the  $\ell_0$  term: I) the  $\ell_0$  term is put in the objective function, II) the  $\ell_0$  term is in the constraint. The algorithms for these two models are similar. The convergence analysis of these two

algorithms is shown in section 4. These algorithms will converge to coordinatewise minimum points. In addition, they will converge to local minimum points (or with probability one) with some modifications in the algorithms. Some experiments are given in section 5 to show the efficiency of the proposed methods for removing impulse noise and mixed Gaussian impulse noise. We will end this work by a short conclusion section.

## 2. Blind Inpainting Models using $\ell_0$ Term.

**2.1. Formulation.** For an  $M \times N$  image,  $\Lambda \in \{0, 1\}^{M \times N}$  is a binary matrix representing a subset  $\Omega_s$  of the pixels as follows:

$$\Lambda_{i,j} = \begin{cases} 1, & \text{if pixel } (i,j) \in \Omega_s, \\ 0, & \text{otherwise.} \end{cases} \quad (2.1)$$

The connection between binary matrix  $\Lambda$  and subset  $\Omega_s$  will be used many times in the follow.

Given a degraded image  $f$ , our objective is to estimate the damaged (or missing) pixels and restore them. We propose the following model using  $\ell_0$  minimization to solve this problem:

$$\underset{u,v}{\text{minimize}} F^P(u, v) \equiv \frac{1}{2} \sum_{i,j} ((Hu)_{i,j} + v_{i,j} - f_{i,j})^2 + \lambda_1 J(u) + \lambda_2 \|v\|_0, \quad (2.2)$$

where  $J(u)$  is the regularization term on the image,  $\lambda_1$  and  $\lambda_2$  are two positive parameters. Here  $P$  means that  $\ell_0$  term is used as a penalty term in the objective function. The parameter  $\lambda_1$  is dependent on the noise level of  $n_1$ . The higher the noise level, the larger the parameter should be. The parameter  $\lambda_2$  is dependent on the noise level of impulse noise. The difference from Dong et al.'s method is that  $\ell_1$  norm is replaced by  $\ell_0$  term. It is difficult to solve this problem because of the  $\ell_0$  term in the function.  $\ell_0$  term makes the problem non-convex and the objective function is non-continuous. Because what we need to find is just  $u$ , we can eliminate  $v$  from problem (2.2) by defining  $E_0^P(u)$  as  $\min_v F^P(u, v)$ , and the problem becomes

$$\underset{u}{\text{minimize}} E_0^P(u) = \min_v F^P(u, v). \quad (2.3)$$

However,  $E_0^P(u)$  is still non-convex and difficult to solve. In order to solve this problem, we will transform the problem into a continuous and multi-convex problem of  $u$  and  $\Lambda$  by introducing a new variable  $\Lambda$ , and by solving the new problem of  $u$  and  $\Lambda$ , we can obtain a local optimal solution for the original problem (2.3) of  $u$  only.

First of all, we provide the intuition behind choosing  $\ell_0$  minimization instead of  $\ell_1$  minimization for  $v$ . Because of the speciality of  $\ell_0$  minimizations,  $v$  can be easily eliminated from  $\min_v F^P(u, v)$  and we can obtain the function of  $u$  only as follows:

$$E_0^P(u) = \frac{1}{2} \sum_{i,j} R_0((Hu)_{i,j} - f_{i,j}) + \lambda_1 J(u), \quad (2.4)$$

where  $R_0(x) = \min(|x|^2, 2\lambda_2)$ . Similarly, we can obtain the function of  $u$  only when  $\ell_1$  term is used instead of  $\ell_0$  term as follows:

$$E_1^P(u) \equiv \frac{1}{2} \sum_{i,j} R_1((Hu)_{i,j} - f_{i,j}) + \lambda_1 J(u), \quad (2.5)$$

where  $R_1(x) = \begin{cases} |x|^2, & \text{if } |x| \leq \lambda_2, \\ 2\lambda_2|x| - \lambda_2^2, & \text{otherwise.} \end{cases}$

The data fidelity terms  $\frac{1}{2} \sum_{i,j} R_0((Hu)_{i,j} - f_{i,j})$  and  $\frac{1}{2} \sum_{i,j} R_1((Hu)_{i,j} - f_{i,j})$  in problems (2.4) and (2.5) are used to approximate the negative log-likelihood resulting from the mixed Gaussian impulse noise model, with each  $R_0$  or  $R_1$  describing the negative log-likelihood for each pixel, because the noise is independently distributed at all pixels. What we are trying to do is finding better and simpler model for mixed Gaussian impulse noise. We can simulate the probability distribution of the pixel values when it is corrupted by both additive Gaussian and random-valued impulse noise. For a fixed pixel value (128 in Fig. 2.1), a value is added as a Gaussian noise, and it is replaced by any random value between  $d_{\min}$  and  $d_{\max}$  (0 and 255 in Fig. 2.1) with some probability related to the impulse noise level. This is run for  $10^8$  times and the approximated negative log-likelihood function is shown in Fig. 2.1. We can see that it is a constant when the value is far from the true pixel value. In this figure we also show  $R_0(x - 128)$  and  $R_1(x - 128)$  with some scaling and lifting.

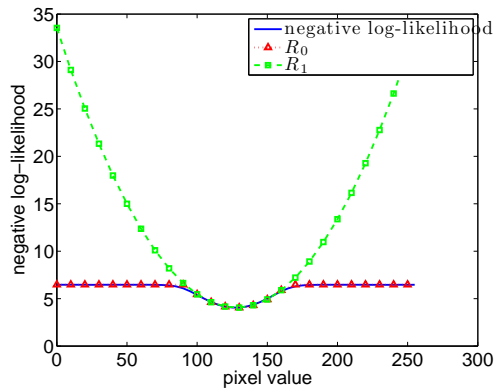


Fig. 2.1: Negative log-likelihood value for the mixed Gaussian impulse noise, and comparison with  $R_0$  and  $R_1$

From Fig. 2.1, we can find that it is more reasonable to use  $\ell_0$  term instead of  $\ell_1$  term. However,  $\ell_1$  minimization makes the problem convex and it is easier to find the solution, while the problem is non-convex and difficult to solve if  $\ell_0$  minimization is used. Next, we will introduce an auxiliary variable  $\Lambda$  or  $\Omega_1$  and we can find a local minimizer of  $E_0^P(u)$  by solving the new problem of  $u$  and  $\Lambda$ .

For any fixed  $\bar{u}$ , we can find the optimal  $v$  by solving the following optimization problem:

$$\underset{v}{\text{minimize}} \frac{1}{2} \sum_{i,j} (H\bar{u} + v - f)^2 + \lambda_2 \|v\|_0.$$

The solution is

$$v_{i,j} = \begin{cases} 0, & \text{if } |f_{i,j} - (H\bar{u})_{i,j}|^2 < 2\lambda_2, \\ f_{i,j} - (H\bar{u})_{i,j}, & \text{if } |f_{i,j} - (H\bar{u})_{i,j}|^2 > 2\lambda_2, \\ 0 \text{ or } f_{i,j} - (H\bar{u})_{i,j}, & \text{if } |f_{i,j} - (H\bar{u})_{i,j}|^2 = 2\lambda_2. \end{cases}$$

When  $v_{i,j} \neq 0$ , we have  $v_{i,j} = f_{i,j} - (H\bar{u})_{i,j}$ . Therefore if we denote

$$\Lambda_{i,j} = \begin{cases} 0, & \text{if } v_{i,j} \neq 0, \\ 1, & \text{if } v_{i,j} = 0, \end{cases} \quad (2.6)$$

then we have a new problem with  $u$  and  $\Lambda$  as follows:

$$\underset{u, \Lambda \in \{0,1\}^{M \times N}}{\text{minimize}} F_1(u, \Lambda) \equiv \frac{1}{2} \sum_{i,j} \Lambda_{i,j} ((Hu)_{i,j} - f_{i,j})^2 + \lambda_1 J(u) + \lambda_2 \sum_{i,j} (1 - \Lambda_{i,j}). \quad (2.7)$$

Problem (2.7) can be solved easily by alternating minimization method, and the algorithm for solving (2.7) is described in section 2.2. For a general alternative minimization procedure for convex and non-convex problems, please see [34].

**Remark:** In fact, the constraint of  $\Lambda \in \{0, 1\}^{M \times N}$  can further be relaxed into  $\Lambda \in [0, 1]^{M \times N}$ , and we have the following multi-convex problem:

$$\underset{u, \Lambda \in [0,1]^{M \times N}}{\text{minimize}} F_1(u, \Lambda) = \frac{1}{2} \sum_{i,j} \Lambda_{i,j} ((Hu)_{i,j} - f_{i,j})^2 + \lambda_1 J(u) + \lambda_2 \sum_{i,j} (1 - \Lambda_{i,j}). \quad (2.8)$$

If  $u$  is fixed,  $F_1(u, \Lambda)$  is a function of  $\Lambda$  only and it is separable. The optimal solution for  $\Lambda$  with fixed  $u$  is

$$\Lambda_{i,j} = \begin{cases} 0, & \text{if } |f_{i,j} - (Hu)_{i,j}|^2 > 2\lambda_2, \\ 1, & \text{if } |f_{i,j} - (Hu)_{i,j}|^2 < 2\lambda_2, \\ t, & \text{if } |f_{i,j} - (Hu)_{i,j}|^2 = 2\lambda_2, \end{cases}$$

where  $t$  is 0 or 1 for unrelaxed problem (2.7), and  $t$  is any number in  $[0, 1]$  for relaxed problem (2.8). If we eliminate  $\Lambda$  as before,  $\min_{\Lambda \in \{0,1\}^{M \times N}} F_1(u, \Lambda)$  and  $\min_{\Lambda \in [0,1]^{M \times N}} F_1(u, \Lambda)$  are functions with respect to  $u$  only and same as  $E_0^P(u)$ . Because all the problems are non-convex, there may exist many local optimal solutions which may not be global optimal solutions. We will show in section 4 that by solving problem (2.8), we can obtain a local minimizer of function  $E_0^P(u)$ .

**2.2. Algorithm.** The objective function defined in (2.8) is non-convex. It is still difficult to solve it in the pair  $(u, \Lambda)$ , but we can use alternating minimization method, which separates the energy minimization over  $u$  and  $\Lambda$  into two steps. For solving the problem in  $u$  with  $\Lambda$  fixed, it is a convex optimization problem for image inpainting and the problem of finding  $\Lambda$  with  $u$  fixed can be solved in one step. These two subproblems are

1) Finding  $u$ : Given an estimate of the support matrix  $\Lambda$ , the minimization over  $u$  is just an image inpainting (and deblurring) problem [35]:

$$\underset{u}{\text{minimize}} \frac{1}{2} \sum_{(i,j) \in \Omega_1} ((Hu)_{i,j} - f_{i,j})^2 + \lambda_1 J(u). \quad (2.9)$$

There are many existing methods for solving this problem.

2) Finding  $\Lambda$ : Given an estimate of the image  $u$ , the minimization over  $\Lambda$  becomes:

$$\underset{\Lambda \in [0,1]^{M \times N}}{\text{minimize}} \frac{1}{2} \sum_{i,j} \Lambda_{i,j} ((Hu)_{i,j} - f_{i,j})^2 - \lambda_2 \sum_{i,j} \Lambda_{i,j}. \quad (2.10)$$

Since this minimization problem of  $\Lambda$  is separable, it can be solved exactly in only one step:

$$\Lambda_{i,j} = \begin{cases} 0 & \text{if } ((Hu)_{i,j} - f_{i,j})^2/2 > \lambda_2, \\ 0 \text{ or } 1 & \text{if } ((Hu)_{i,j} - f_{i,j})^2/2 = \lambda_2, \\ 1 & \text{if } ((Hu)_{i,j} - f_{i,j})^2/2 < \lambda_2. \end{cases} \quad (2.11)$$

Therefore, the proposed algorithm for blind inpainting with  $\ell_0$  minimization is iteratively finding  $u$  and  $\Lambda$ . As mentioned in section 2.1,  $\min_{\Lambda \in \{0,1\}^{M \times N}} F_1(u, \Lambda) = \min_{\Lambda \in \{0,1\}^{M \times N}} F_1(u, \Lambda)$  for all fixed  $u$ , thus we can force  $\Lambda_{i,j} \in \{0,1\}$  during the algorithm. When  $((Hu)_{i,j} - f_{i,j})^2/2 = \lambda_2$ , we can randomly choose  $\Lambda_{i,j}$  to be 0 or 1.

The detailed algorithm for blind inpainting is described below, the initial  $\Lambda^0$  is chosen by the methods for detecting the impulse noise (AMF for salt-and-pepper impulse noise and ACWMF for random-valued impulse noise). Usually three iterations are sufficient, as shown in the experiments.

---

**Algorithm 1** Proposed blind inpainting algorithm.

---

**Input:**  $f, \lambda_1, \lambda_2, \Lambda^0, \epsilon$   
**Initialization:**  $k = 1$ .  
**while**  $k < 2$  or  $F_1(u^k, \Lambda^k) - F_1(u^{k-1}, \Lambda^{k-1}) > \epsilon$  **do**  
    Obtain  $u^k$  by solving (2.9).  
    Obtain  $\Lambda^k$  by (2.11).  
     $k = k + 1$ .  
**end while**

---

Here  $\epsilon$  is chosen to be small and served as a stopping criteria to stop the algorithm when the difference in function values between two iterations is too small.  $\lambda_1$  and  $\lambda_2$  are two parameters depending on the noise levels of  $n_1$  and  $n_2$ .

**Remark:** This algorithm and the algorithm in next section depend on the initial  $\Lambda^0$ , and choosing a better  $\Lambda^0$  will reduce the total number of iterations and the restoration result (because of the non-convexity of the problem). Therefore, we can choose the result of AMF and ACWMF for salt-and-pepper and random-valued impulse noise respectively as initial  $\Lambda^0$ . For salt-and-pepper impulse noise, AMF will provide a very accurate initial guess for  $\Lambda$  for most cases and the improvement from more iterations is not too much, while for random-valued impulse noise, the output of ACWMF is not accurate, and more iterations are needed to improve the detection of corrupted pixels and the recovery result.

**3. Blind Inpainting Using Adaptive Outlier Pursuit.** In the previous section, we proposed a method for blind inpainting by putting a  $\ell_0$  term in the objective function, which can be solved by iteratively updating the set of pixels damaged by impulse noise (or the binary matrix  $\Lambda \in \{0,1\}^{M \times N}$ ) and restoring the image. Instead of putting the  $\ell_0$  term in the objective function, we can also put a constraint on  $\|v\|_0$ , which will be equivalent to a constraint on  $\sum_{i,j} \Lambda_{i,j}$ . This technique has been applied to robust 1-bit compressive sensing where there are sign flips in the binary measurements belonging to  $\{-1,1\}$  [36] and robust matrix completion [37]. We proposed an algorithm, named adaptive outlier pursuit (AOP), which can adaptively find the sign flips (outliers) and reconstruct the signal by using other measurements assumed to be correct. Since images corrupted by impulse noise can also be considered as sparsely

corrupted measurements, the same idea can be applied in impulse noise (and mixed Gaussian impulse noise) removal by iteratively finding the pixels corrupted by impulse noise and recovering the image using other pixels.

Let us assume that the number of pixels corrupted by impulse noise is bounded above by a integer  $L$ , this can be obtained from the noise level of impulse noise. Therefore, the new problem is

$$\begin{aligned} & \underset{u,v}{\text{minimize}} && \frac{1}{2} \sum_{i,j} ((Hu)_{i,j} + v_{i,j} - f_{i,j})^2 + \lambda_1 J(u), \\ & \text{subject to} && \|v\|_0 \leq L, \end{aligned} \quad (3.1)$$

which can be written as

$$\underset{u,v}{\text{minimize}} F^C(u, v) \equiv \frac{1}{2} \sum_{i,j} ((Hu)_{i,j} + v_{i,j} - f_{i,j})^2 + \lambda_1 J(u) + \iota_{\{v: \|v\|_0 \leq L\}}, \quad (3.2)$$

where  $\iota_{\{v: \|v\|_0 \leq L\}}$  is the indicator function equals to zero when  $\|v\|_0 \leq L$  and  $+\infty$  otherwise. Here  $C$  means that the  $\ell_0$  term is put in the constraint. We can further eliminate  $v$ , and the problem of  $u$  only is:

$$\underset{u}{\text{minimize}} E_0^C(u) \equiv \min_v F^C(u, v). \quad (3.3)$$

Similarly, we can introduce new variation  $\Lambda$  and the corresponding problem of  $u$  and  $\Lambda$  is

$$\begin{aligned} & \underset{u,\Lambda}{\text{minimize}} && \sum_{i,j} \frac{1}{2} \Lambda_{i,j} ((Hu)_{i,j} - f_{i,j})^2 + \lambda_1 J(u), \\ & \text{subject to} && \sum_{i,j} (1 - \Lambda_{i,j}) \leq L, \quad \Lambda_{i,j} \in \{0, 1\}, \end{aligned} \quad (3.4)$$

which can be described in another way as

$$\underset{u,\Lambda \in \{0,1\}^{M \times N}}{\text{minimize}} F_2(u, \Lambda) \equiv \sum_{i,j} \frac{1}{2} \Lambda_{i,j} ((Hu)_{i,j} - f_{i,j})^2 + \lambda_1 J(u) + \iota_{\{\Lambda: \sum_{i,j} (1 - \Lambda_{i,j}) \leq L\}}, \quad (3.5)$$

where  $\iota_{\{\Lambda: \sum_{i,j} (1 - \Lambda_{i,j}) \leq L\}}$  is the indicator function equals to zero when  $\sum_{i,j} (1 - \Lambda_{i,j}) \leq L$  and  $+\infty$  otherwise. Note that we can also relax the constraint  $\Lambda \in \{0, 1\}^{M \times N}$  to  $\Lambda \in [0, 1]^{M \times N}$  and the problem becomes a multi-convex problem, as done in the previous section. In addition we have  $\min_{\Lambda \in \{0,1\}^{M \times N}} F_2(u, \Lambda) = \min_{\Lambda \in [0,1]^{M \times N}} F_2(u, \Lambda) = E_0^C(u)$ .

This problem can also be solved iteratively as in the previous section. The  $u$ -subproblem is same as the previous one, and the  $\Lambda$ -subproblem is slightly different. In order to update  $\Lambda$ , we have to solve

$$\begin{aligned} & \underset{\Lambda}{\text{minimize}} && \sum_{i,j} \Lambda_{i,j} ((Hu)_{i,j} - f_{i,j})^2 / 2, \\ & \text{subject to} && \sum_{i,j} (1 - \Lambda_{i,j}) \leq L, \quad \Lambda_{i,j} \in \{0, 1\}. \end{aligned} \quad (3.6)$$

This problem is to choose  $M \times N - L$  elements with least sum from  $M \times N$  elements  $\{((Hu)_{i,j} - f_{i,j})^2 / 2\}_{i=1, j=1}^{M, N}$ . Given a  $u$  estimated from (2.9), we can update  $\Lambda$  in one step:

$$\Lambda_{i,j} = \begin{cases} 0, & \text{if } ((Hu)_{i,j} - f_{i,j})^2 / 2 \geq \lambda_2, \\ 1, & \text{if } ((Hu)_{i,j} - f_{i,j})^2 / 2 < \lambda_2, \end{cases} \quad (3.7)$$

where  $\lambda_2$  is the  $L^{\text{th}}$  largest term of  $\{((Hu)_{i,j} - f_{i,j})^2/2\}_{i=1,j=1}^{M,N}$ . If the  $L^{\text{th}}$  and  $(L+1)^{\text{th}}$  largest terms are equal, then we can choose any binary matrix  $\Lambda$  such that  $\sum_{i,j} \Lambda_{i,j} = M \times N - L$  and

$$\min_{i,j,\Lambda_{i,j}=0} ((Hu)_{i,j} - f_{i,j})^2/2 \geq \max_{i,j,\Lambda_{i,j}=1} ((Hu)_{i,j} - f_{i,j})^2/2. \quad (3.8)$$

The algorithm for blind inpainting using AOP is described below.

---

**Algorithm 2** Proposed blind inpainting using AOP.

---

**Input:**  $f, \lambda_1, L, \Lambda^0, \epsilon$   
**Initialization:**  $k = 1$ .  
**while**  $k < 2$  or  $F_2(u^k, \Lambda^k) - F_2(u^{k-1}, \Lambda^{k-1}) > \epsilon$  **do**  
    Obtain  $u^k$  by solving (2.9).  
    Obtain  $\Lambda^k$  by (3.7).  
     $k = k + 1$ .  
**end while**

---

Here  $\epsilon, \Lambda^0$  and  $\lambda_1$  are chosen in the same way as algorithm 1. The integer  $L$  is an estimation of the number of corrupted pixels, which is easy to obtain from the noise level of the impulse noise.

The difference between these two algorithms is the  $\Lambda$ -subproblem, the threshold  $\lambda_2$  is fixed for algorithm 1, while  $\lambda_2$  is changing for AOP. AOP can be considered as one special case of algorithm 1 with changing  $\lambda_2$ . However the performance of these two algorithms is similar, and the parameter  $L$  is easier to obtain than  $\lambda_2$ . So we will only use algorithm 2 for numerical experiments.

**4. Convergence Analysis.** In this section, we establish some convergence results for these two algorithms. We will show that these two algorithms will stop in finite steps, and the output is a coordinatewise minimum point of  $F_1(u, \Lambda)$  (or  $F_2(u, \Lambda)$ ) with relaxed constraint  $\Lambda_{i,j} \in [0, 1]$ . A point  $(\tilde{u}, \tilde{\Lambda})$  is a coordinatewise minimum point of  $F_1(u, \Lambda)$  (or  $F_2(u, \Lambda)$ ) means that  $\tilde{u}$  is a minimizer of  $F_1(u, \tilde{\Lambda})$  (or  $F_2(u, \tilde{\Lambda})$ ) and  $\tilde{\Lambda}$  is a minimizer of  $F_1(\tilde{u}, \Lambda)$  (or  $F_2(\tilde{u}, \Lambda)$ ). In addition, we can modify a little bit in the algorithm and the output will be a local minimizer of  $E_0^P(u)$  (or  $E_0^C(u)$ ) (or with probability one).

Since the convergence analysis is similar for both algorithms, let  $F(u, \Lambda)$  stand for  $F_1(u, \Lambda)$  in the penalty problem (2.7) and  $F_2(u, \Lambda)$  in the constraint problem (3.5), and  $E_0(u)$  stand for  $E_0^P(u)$  and  $E_0^C(u)$  as  $\min_{\Lambda \in \{0,1\}^{M \times N}} F(u, \Lambda)$ .

Before deriving the convergence results of the algorithms, two theorems are introduced to show that we can solve problems (2.7) and (3.5) of  $u$  and  $\Lambda$  to find a local minimizer for  $E_0(u)$ .

**THEOREM 4.1.** *If  $u^*$  is a local minimizer of  $E_0(u)$ , then for any  $\Lambda^* \in \{0,1\}^{M \times N}$  minimizing  $F(u^*, \Lambda)$ ,  $(u^*, \Lambda^*)$  is a local minimizer of  $F(u, \Lambda)$ .*

*Proof.* Since  $u^*$  is a local minimizer of  $E_0(u)$ , we can find  $\epsilon > 0$  such that for all  $u$  satisfying  $\|u - u^*\| < \epsilon$ , we have  $E_0(u) \geq E_0(u^*)$ . Therefore for all  $(u, \Lambda)$  satisfying  $\|(u, \Lambda) - (u^*, \Lambda^*)\| < \epsilon$ , we have  $\|u - u^*\| < \epsilon$ .

$$F(u, \Lambda) \geq E_0(u) \geq E_0(u^*) = F(u^*, \Lambda^*).$$

Thus  $(u^*, \Lambda^*)$  is a local minimizer of  $F(u, \Lambda)$ .  $\square$



To obtain a sufficient condition for  $u^*$  being a local minimizer of  $E_0(u)$ , we have the following theorem.

**THEOREM 4.2.** *Given fixed  $u^*$ , if for all  $\bar{\Lambda} \in \{0, 1\}^{M \times N}$  minimizing  $F(u^*, \Lambda)$ , we also have that  $u^*$  minimizing  $F(u, \bar{\Lambda})$ , then  $u^*$  is a local minimizer of  $E_0(u)$ .*

*Proof.* We will prove it for the penalty problem (2.7) with  $F(u, \Lambda) = F_1(u, \Lambda)$  first. Let  $\Omega_+ = \{(i, j) : |f_{i,j} - (Hu^*)_{i,j}|^2 > 2\lambda_2\}$  and  $\Omega_- = \{(i, j) : |f_{i,j} - (Hu^*)_{i,j}|^2 < 2\lambda_2\}$ , then from the continuity of  $H$ , we can find  $\epsilon > 0$  such that when  $\|u - u^*\| < \epsilon$ , we have  $|f_{i,j} - (Hu)_{i,j}|^2 > 2\lambda_2$  for all  $(i, j) \in \Omega_+$  and  $|f_{i,j} - (Hu)_{i,j}|^2 < 2\lambda_2$  for all  $(i, j) \in \Omega_-$ . Notice that  $E_0(u) = \min_{\Lambda \in \{0, 1\}^{M \times N}} F(u, \Lambda)$ , then there exists  $\bar{\Lambda} \in \{0, 1\}^{M \times N}$  such that  $E_0(u) = F(u, \bar{\Lambda})$ . We have  $\bar{\Lambda}_{i,j} = 0$  when  $(i, j) \in \Omega_+$  and  $\bar{\Lambda}_{i,j} = 1$  when  $(i, j) \in \Omega_-$ . Thus  $\bar{\Lambda}$  is also a minimizer of  $F_1(u^*, \Lambda)$ . In addition, we have  $E_0(u^*) = F(u^*, \bar{\Lambda}) \leq F(u, \bar{\Lambda}) = F(u)$ . Therefore,  $u^*$  is a local minimizer of  $E_0(u)$ .

For the constraint problem (3.5) with  $F(u, \Lambda) = F_2(u, \Lambda)$ , we have to just replace  $\lambda_2$  with the  $L^{\text{th}}$  largest term of  $\{((Hu)_{i,j} - f_{i,j})^2\}_{i=1, j=1}^{M, N}$ , and the result follows.  $\square$

**Remark:** We can replace the  $\{0, 1\}^{M \times N}$  with the relaxed version  $[0, 1]^{M \times N}$  and both theorems are still valid.

With these theorems, we are ready to shown the convergence results of the two algorithms.

**THEOREM 4.3.** *Both algorithms will converge in finite steps and the output  $(u^*, \Lambda^*)$  is a coordinatewise minimum point of  $F(u, \Lambda)$ .*

*Proof.* As explained before, though we have  $\Lambda_{i,j} \in [0, 1]$  for the relaxed problem, we can always force  $\Lambda_{i,j} \in \{0, 1\}$  during the iterations because for every fixed  $\bar{u}$ , we can also find a minimizer of  $F(\bar{u}, \Lambda)$  satisfying  $\Lambda_{i,j} \in \{0, 1\}$ . Since  $\Lambda_{i,j} \in \{0, 1\}$ , there are only finite number of  $\Lambda$ 's (the total number of different  $\Lambda$ 's with constraint  $\Lambda_{i,j} \in \{0, 1\}$  is  $2^{M \times N}$ ) and the algorithm will stop in finite steps if the  $u$ -subproblem and  $\Lambda$ -subproblem are solved exactly. Assume that at step  $k$ , the function  $F(u, \Lambda)$  stops decreasing, which means

$$F(u^k, \Lambda^k) = F(u^{k+1}, \Lambda^{k+1}). \quad (4.1)$$

Together with nonincreasing property of the algorithm

$$F(u^k, \Lambda^k) \geq F(u^{k+1}, \Lambda^k) \geq F(u^{k+1}, \Lambda^{k+1}), \quad (4.2)$$

we have

$$F(u^k, \Lambda^k) = F(u^{k+1}, \Lambda^k) = F(u^{k+1}, \Lambda^{k+1}). \quad (4.3)$$

Thus

$$F(u^k, \Lambda^k) = F(u^{k+1}, \Lambda^k) = \min_u F(u, \Lambda^k), \quad (4.4)$$

$$F(u^k, \Lambda^k) = \min_{\Lambda} F(u^k, \Lambda). \quad (4.5)$$

Then  $(u^*, \Lambda^*) = (u^k, \Lambda^k)$  is a coordinatewise minimum point of  $F(u, \Lambda)$ .  $\square$

However, **coordinatewise minimum point**  $(\tilde{u}, \tilde{\Lambda})$  may not be a local minimum point of  $F(u, \Lambda)$ . As shown in the next theorem,  $\tilde{\Lambda}$  being the unique minimum point of  $F(\tilde{u}, \Lambda)$  is a sufficient condition for  $(\tilde{u}, \tilde{\Lambda})$  to be a local minimum point.

**THEOREM 4.4.** *For a coordinatewise minimum point  $(u^*, \Lambda^*)$  of  $F(u, \Lambda)$ , if  $\Lambda^*$  is the unique minimum point of  $F(u^*, \Lambda)$ , then  $(u^*, \Lambda^*)$  is a local minimum point of  $F(u, \Lambda)$ . Furthermore,  $u^*$  is a local minimum point of  $E_0(u)$ .*

*Proof.* Since  $\Lambda^*$  is the unique minimum point of  $F(u^*, \Lambda)$  and  $u^*$  minimizes  $F(u, \Lambda^*)$ , we have  $u^*$  being a local minimum point of  $E_0(u)$  from theorem 4.2. Then  $(u^*, \Lambda^*)$  being a local minimum point of  $F(u, \Lambda)$  follows from theorem 4.1.  $\square$

Let  $(u^*, \Lambda^*)$  be a coordinatewise minimum point of  $F(u, \Lambda)$ . From theorem 4.4, if  $u^*$  is not a local minimum point of  $E_0(u)$ , then there are many minimum points for  $F(u^*, \Lambda)$ . In addition, from theorem 4.2, there exists another minimum point  $\bar{\Lambda}$  of  $F(u^*, \Lambda)$ , such that  $F(u^*, \bar{\Lambda}) > \min_u F(u, \bar{\Lambda})$ .

Based on theorems 4.2 and 4.4, we have the following two corollaries.

**COROLLARY 4.5.** *When solving the subproblem of finding  $\Lambda^k$ , if there are many minimum points for  $F(u^k, \Lambda)$ , we can choose the best  $\Lambda$  with lowest  $\min_u F(u, \Lambda)$ , then the algorithm will stop at a local minimum point of  $E_0(u)$ .*

*Proof.* When the algorithm stops at  $(u^*, \Lambda^*)$ , we have  $F(u^*, \bar{\Lambda}) = \min_u F(u, \bar{\Lambda})$  for all  $\bar{\Lambda} \in \{0, 1\}^{M \times N}$  minimizing  $F(u^*, \Lambda)$ , otherwise, we can find a  $\Lambda$  giving lower  $\min_u F(u, \Lambda)$  and the algorithm continues. Then from theorem 4.2, we know that  $u^*$  is a local minimum point of  $E_0(u)$ .  $\square$

**Remark:** This strategy for choosing  $\Lambda$  may not be very useful for practical computation because there could be a large number of candidates for such  $\Lambda$ . However, in the numerical experiments, this does not happen, because of the rounding error in the calculation and the probability for two values to equal is 0.

Instead of having to choose the best candidates from many candidates, we can modify the function  $F(u, \Lambda)$  to avoid this case, as Wang et al. did in [38].

**COROLLARY 4.6.** *If we can modify the objective function  $F(u, \Lambda)$  by adding  $\tau \sum_{i,j} \Lambda_{i,j} r_{i,j}$ , where  $\{r_{i,j}\}$  are random values uniformly distributed in  $[0, 1]$  and  $\tau$  is a small number, then the algorithm will stop at a local minimum of  $\tilde{E}_0(u) \equiv \min_{\Lambda \in \{0,1\}^{M \times N}} (F(u, \Lambda) + \tau \sum_{i,j} \Lambda_{i,j} r_{i,j})$  with probability one.*

*Proof.* In this case, the subproblem for updating  $\Lambda$  is

$$\Lambda_{i,j} = \begin{cases} 0 & \text{if } ((Hu)_{i,j} - f_{i,j})^2 / 2 + \tau r_{i,j} > \lambda_2, \\ 0 \text{ or } 1 & \text{if } ((Hu)_{i,j} - f_{i,j})^2 / 2 + \tau r_{i,j} = \lambda_2, \\ 1 & \text{if } ((Hu)_{i,j} - f_{i,j})^2 / 2 + \tau r_{i,j} < \lambda_2. \end{cases} \quad (4.6)$$

The probability of getting  $((Hu)_{i,j} - f_{i,j})^2 / 2 + \tau r_{i,j} = \lambda_2$  is 0 because of the randomness of  $r_{i,j}$ . Then the algorithm will converge to a local minimum of  $\tilde{E}_0(u)$  with probability one. Similarly for AOP, the probability of  $L^{\text{th}}$  and  $(L+1)^{\text{th}}$  largest term being equal is 0.  $\square$

**Remark:** Besides adding additional term onto  $F(u, \Lambda)$  we can also add a small random variable  $\tau \Lambda_{i,j} r_{i,j}$  onto  $f_{i,j}$ .

**5. Numerical Experiments.** Because the performance of these two algorithms is similar and it is easier to obtain an approximate value for the number of damaged pixels, in this section, we apply the blind inpainting algorithm using AOP to remove impulse noise and mixed Gaussian impulse noise.

As mentioned in the introduction, there are many different choices for  $J(x)$ , and the performance of this algorithm depends on which  $J(x)$  is chosen. In order to make the comparison with other methods fair, we choose  $J(x)$  to be total variation for all methods used in the numerical experiments. Therefore the optimization problem will be different from the ones that proposed in the literature. For example, we implement the algorithm of paper [25] by replacing the wavelet frame with total variation [39].

If the regularization for image  $u$  is total variation, and  $H$  is the identity operator,

the step for finding  $u$  is

$$\underset{u}{\text{minimize}} \sum_{i,j} \frac{1}{2} \Lambda_{i,j} (u_{i,j} - f_{i,j})^2 + \lambda_1 \text{TV}(u), \quad (5.1)$$

which is the famous TV inpainting model [40, 41]. Numerous algorithms proposed for solving TV denoising problem can be adopted to solve this TV inpainting problem with some necessary modifications. Some examples are algorithms based on duality [42, 43], augmented Lagrangian methods [44, 45], and split Bregman iterations [26, 46]. In the numerical experiments, we choose the split Bregman iteration to solve the subproblem.

To evaluate the quality of the restoration results, peak signal to noise ratio (PSNR) is employed. Given an image  $u \in [0, 255]^{M \times N}$ , the PSNR of the restoration result  $\hat{u}$  is defined as follows:

$$\text{PSNR}(\hat{u}, u) = 10 \log_{10} \frac{255^2}{\frac{1}{MN} \sum_{i,j} (\hat{u}_{i,j} - u_{i,j})^2}. \quad (5.2)$$

There are two important types of impulse noise: salt-and-pepper impulse noise and random-valued impulse noise. The pixels damaged by salt-and-pepper impulse noise are much easier to find since the values are either  $d_{\min}$  or  $d_{\max}$ . The adaptive median filter (AMF) has been widely used to accurately identify most pixels damaged by salt-and-pepper impulse noise (See e.g. [47, 22]). The detection of pixels corrupted by random-valued impulse noise is much harder than salt-and-pepper impulse noise because the value of damaged pixels can be any number between  $d_{\min}$  and  $d_{\max}$ . ACWMMF was proposed to detect pixels damaged by random-valued impulse noise.

For the first experiment, salt-and-pepper impulse noise is considered. Because the pixels corrupted by this kind of impulse noise can only take two values, the detection of damaged pixels is easy. As an efficient method for detecting the damaged pixels, AMF is used widely in salt-and-pepper impulse noise removal. We will compare total variation blind inpainting using AOP with AMF and TVL1, where TVL1 is the result of solving the following problem,

$$\underset{u}{\text{minimize}} \sum_{i,j} |u_{i,j} - f_{i,j}| + \lambda \text{TV}(u), \quad (5.3)$$

using split Bregman [46]. The parameter  $\lambda$  is tuned to achieve the best quality of the restoration images.

Four test images are corrupted by Gaussian noise of zero mean and standard deviations  $\sigma = 5, 10, 15$ , then we add salt-and-pepper impulse noise with different levels ( $s = 30\%, 50\%, 70\%$ ) on the test images, with or without the Gaussian noise. The PSNR values of the results from three methods are summarized in Table 5.1.

From Table 5.1, we can see that for salt-and-pepper impulse noise, the results from total variation blind inpainting using AOP are better than those by AMF and TVL1 for all noise levels. The visual comparison of some results is shown in Fig. 5.1. We can see noisy artifacts in the background of the images obtained by AMF, and the images obtained by TVL1 are blurred with some lost details. Images restored by total variation blind inpainting using AOP are smooth in flat regions of the background and the details are kept.

Salt-and-Pepper Impulse Noise								
$\sigma + s$	"Lena"				"House"			
	Noisy	AMF	TVL1	AOP	Noisy	AMF	TVL1	AOP
0+30%	10.68	33.80	30.97	<b>37.75</b>	10.42	38.97	36.53	<b>47.14</b>
5+30%	10.66	31.47	30.32	<b>34.56</b>	10.40	33.69	34.49	<b>39.09</b>
10+30%	10.62	27.93	29.40	<b>32.25</b>	10.39	28.90	32.79	<b>35.73</b>
15+30%	10.54	25.14	28.59	<b>30.41</b>	10.30	25.66	31.41	<b>33.49</b>
0+50%	8.44	30.35	27.98	<b>33.98</b>	8.22	34.60	31.70	<b>42.50</b>
5+50%	8.45	29.00	27.58	<b>32.61</b>	8.19	31.73	31.36	<b>37.43</b>
10+50%	8.42	26.54	27.25	<b>30.88</b>	8.18	27.84	30.37	<b>34.60</b>
15+50%	8.40	24.15	26.46	<b>29.50</b>	8.14	24.96	29.60	<b>32.50</b>
0+70%	7.00	26.85	24.90	<b>30.61</b>	6.75	30.05	26.80	<b>36.84</b>
5+70%	6.97	26.11	24.65	<b>29.94</b>	6.73	28.80	26.81	<b>34.44</b>
10+70%	6.98	24.62	24.57	<b>29.05</b>	6.74	26.15	26.36	<b>32.28</b>
15+70%	6.97	22.83	24.20	<b>28.11</b>	6.72	23.86	25.85	<b>31.23</b>
	"Cameraman"				"Boat"			
	Noisy	AMF	TVL1	AOP	Noisy	AMF	TVL1	AOP
0+30%	10.32	33.62	30.43	<b>38.43</b>	10.70	30.16	27.64	<b>33.32</b>
5+30%	10.28	31.34	30.09	<b>35.33</b>	10.69	30.27	27.84	<b>33.39</b>
10+30%	10.25	28.15	29.35	<b>32.47</b>	10.69	30.34	27.70	<b>33.06</b>
15+30%	10.21	25.39	28.40	<b>30.45</b>	10.68	30.24	27.70	<b>32.96</b>
0+50%	8.08	29.78	26.80	<b>34.58</b>	8.49	27.27	25.00	<b>30.54</b>
5+50%	8.08	28.35	26.55	<b>32.78</b>	8.51	27.20	25.02	<b>30.49</b>
10+50%	8.09	26.38	26.16	<b>30.57</b>	8.48	27.24	25.24	<b>30.19</b>
15+50%	8.04	24.29	26.09	<b>29.08</b>	8.49	27.12	25.02	<b>30.12</b>
0+70%	6.62	25.73	23.20	<b>29.85</b>	7.02	24.33	22.42	<b>27.20</b>
5+70%	6.62	25.22	23.21	<b>29.06</b>	7.02	24.19	22.35	<b>27.14</b>
10+70%	6.60	24.02	23.01	<b>28.04</b>	7.02	24.18	22.42	<b>27.08</b>
15+70%	6.60	22.48	22.66	<b>26.96</b>	7.01	24.23	22.37	<b>26.97</b>

Table 5.1: PSNR(dB) for denoising results of different algorithms for noisy images corrupted by salt-and-pepper impulse noise and mixed Gaussian impulse noise.  $\sigma$  is the standard deviation for the Gaussian noise and  $s$  is the level of salt-and-pepper impulse noise.

We do not compare AOP with two-stage approaches because the detection of damaged pixels by salt-and-pepper impulse noise using AMF is very accurate, and  $\Lambda$  will not change too much during the iterations, thus the performance of AOP will be similar to the two-stage approach by first detecting the damaged pixels by AMF and then solve the total variation image inpainting problem, which is just the first iteration of AOP. Therefore, for the cases with salt-and-pepper impulse noise, using blind inpainting does not improve too much by adaptively updating  $\Lambda$ , and our focus will be on random-valued impulse noise.

For random-valued impulse noise removal, it is more difficult to detect the corrupted pixels because they can take any value between  $d_{min}$  and  $d_{max}$ . ACWMF and ROLD are used in two-stage approaches for detecting the damaged pixels [23, 24]. In this experiment, we will compare AOP with ACWMF, TVL1, and two-stage approaches for random-valued impulse noise removal. The two-stage approach we used here is just one step of AOP, and the parameter for second stage (total variation image inpainting) is also tuned to achieve the best quality of the restoration images. In addition, we will compare with two other methods, one is Dong et al.'s method [25]



Fig. 5.1: Denoising results of images contaminated by both Gaussian noise and salt-and-pepper impulse noise with  $\sigma = 10$  and  $s = 30\%$ . Top row: noisy images; Second row: the results restored by AMF; Third row: the results restored by TVL1; Bottom row: the results restored by total variation blind inpainting using AOP.

for solving problem

$$\underset{u,v}{\text{minimize}} \frac{1}{2} \sum_{i,j} ((Hu)_{i,j} + v_{i,j} - f_{i,j})^2 + \lambda_1 TV(u) + \lambda_2 \sum_{i,j} |v_{i,j}|. \quad (5.4)$$

The parameters  $\lambda_1$  and  $\lambda_2$  are chosen to achieve the best quality. The other is the penalty decomposition method (PD) by Lu and Zhang [33] for the problem

$$\begin{aligned} & \underset{u,v}{\text{minimize}} \quad \frac{1}{2} \sum_{i,j} ((Hu)_{i,j} + v_{i,j} - f_{i,j})^2 + \lambda_1 TV(u), \\ & \text{subject to} \quad \|v\|_0 \leq L. \end{aligned} \quad (5.5)$$

$\lambda_1$  is tuned to achieve the best quality of the restoration images.

Again four test images are corrupted by Gaussian noise of zero mean and standard deviation ( $\sigma = 10, 25$ ), then we add random-valued impulse noise with different levels



( $s = 25\%, 40\%$ ) onto the test images, with or without Gaussian noise. The PSNR values of the results from these six methods are summarized in Table 5.2.

Random-Valued Impulse Noise								
	$\sigma + s$	Noisy	ACWMF	TVL1	Two-Stage	[25]	PD	AOP
"Lena"	0+25%	15.25	30.53	31.75	32.65	31.09	33.42	<b>33.74</b>
	10+25%	15.26	30.44	31.87	32.65	31.21	33.42	<b>33.66</b>
	25+25%	15.11	29.23	30.37	31.32	30.13	32.36	<b>32.64</b>
	0+40%	13.27	24.62	29.22	28.44	28.61	30.39	<b>30.77</b>
	10+40%	13.22	24.31	28.94	28.15	28.41	30.14	<b>30.34</b>
	25+40%	13.05	23.42	28.02	27.23	27.40	29.23	<b>29.54</b>
"House"	0+25%	14.71	31.50	36.88	35.55	35.04	41.38	<b>42.11</b>
	10+25%	14.71	31.38	36.36	35.55	35.20	41.36	<b>41.61</b>
	25+25%	14.49	30.48	35.51	34.62	34.33	40.54	<b>40.85</b>
	0+40%	12.65	23.90	32.77	30.07	31.45	35.53	<b>37.39</b>
	10+40%	12.62	23.82	32.44	29.97	31.24	35.24	<b>36.71</b>
	25+40%	12.34	22.93	31.52	29.02	30.38	34.32	<b>35.86</b>
"Cameraman"	0+25%	14.48	28.93	31.24	31.75	30.54	32.72	<b>33.16</b>
	10+25%	14.52	28.99	31.32	31.86	30.54	32.72	<b>33.26</b>
	25+25%	14.10	27.98	30.43	30.93	29.63	31.80	<b>32.55</b>
	0+40%	12.37	22.26	27.36	26.51	27.00	28.39	<b>29.16</b>
	10+40%	12.39	22.42	27.75	26.74	27.37	28.42	<b>29.21</b>
	25+40%	12.02	21.32	26.69	25.56	26.30	27.43	<b>28.11</b>
"Boat"	0+25%	15.29	28.18	28.63	29.37	28.26	29.48	<b>29.60</b>
	10+15%	15.35	28.16	28.76	29.48	28.36	29.61	<b>29.78</b>
	25+25%	14.01	27.03	27.73	28.58	27.43	28.62	<b>28.81</b>
	0+40%	13.30	23.56	26.16	26.03	25.88	26.88	<b>27.12</b>
	10+40%	13.31	23.42	26.22	25.96	25.90	26.88	<b>26.99</b>
	25+40%	13.12	22.37	25.27	25.00	24.79	25.93	<b>26.02</b>

Table 5.2: PSNR(dB) for denoising results of different algorithms for noisy images corrupted by random-valued impulse noise and mixed Gaussian impulse noise.  $\sigma$  is the standard deviation for the Gaussian noise and  $s$  is the level of random-valued impulse noise.

From Table 5.2, we can see that for random-valued impulse noise, the results from total variation blind inpainting using AOP are better than those by other methods for all noise levels. The comparison of ACWMF and TVL1 shows that TVL1 outperforms ACWMF for all noise levels tested, because ACWMF misses quite a lot of real noise and false-hits some noise-free pixels. TVL1 has better performance than two-stage approach for the cases when noise level is high ( $s = 40\%$  in the numerical experiments), because the accuracy of detecting corrupted pixels by random-valued impulse noise using ACWMF is very low when the noise level is high. The accuracy of detecting corrupted pixels can be improved by our method via iteratively updating the binary matrix  $\Lambda$ , as shown in the comparison. PD outperforms other methods except AOP because  $\ell_0$  term is used in the problem. The problem PD solves is a non-continuous problem and it will stop at a local minimum ( $u^*, v^*$ ). However,  $u^*$  may not be a local minimum of the problem with  $u$  only. While AOP will converge to a local minimum of the problem with  $u$  only.

The visual comparison of some results is shown in Fig. 5.2. We can see noisy artifacts in the background of the images obtained by ACWMF, and the images obtained by TVL1 are blurred with some lost details. Images restored by total variation blind inpainting are smooth in flat regions of the background and the details are kept.

Both experiments show that our method by iteratively updating the inpainting re-



Fig. 5.2: Denoising results of images contaminated by both Gaussian noise and random-valued impulse noise with  $\sigma = 10$  and  $s = 25\%$ . Top row: noisy images; Second row: the results restored by ACWMF; Third row: the results restored by TVL1; Bottom row: the results restored by total variation blind inpainting using AOP.

gion and performing image inpainting provides better results in identifying the outliers and recovering damaged pixels. For salt-and-pepper impulse noise, because there are very accurate methods for detecting the corrupted pixels such as AMF, our method has similar performance as two-stage approaches. However, for random-valued impulse noise, there is no method can detect corrupted pixels accurately, especially when the noise level is high. Our method by iteratively updating the corrupted pixels is a better choice.

At the end of this section, we compare the damaged pixels detected by ACWMF and obtained from AOP in Fig. 5.3 for the cameraman image. The damaged pixels are chosen randomly ( $s = 40\%$ ). The pixels with black color are detected as damaged. The set obtained from AOP is also random and does not contain any information from

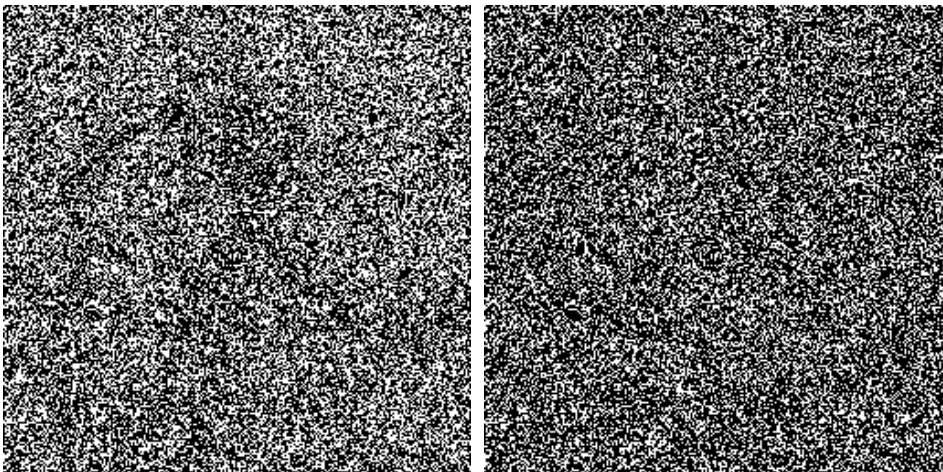


Fig. 5.3: The damaged pixels detected by ACWMF (left column) and AOP (right column).

the image, while the set detected by ACWMF still has some features from cameraman image.

**6. Conclusion.** This paper presents two general algorithms based on blind inpainting and  $\ell_0$  minimization for removing impulse noise. The difference is in the treatment for the  $\ell_0$  term: I) the  $\ell_0$  term is put in the objective function, II) the  $\ell_0$  term is in the constraint. Both problems can be solved by iteratively restoring the images and identifying the damaged pixels. The performance of these two methods is similar, and the connection between these two methods is shown. It is also shown in the experiments that the proposed methods perform better than other methods. This simple idea can also be applied to other cases where the noise model is not Gaussian.

**Acknowledgment.** This work was supported by NSF Grant DMS-0714945 and Center for Domain-Specific Computing (CDSC) under the NSF Expeditions in Computing Award CCF-0926127. We would like to thank the anonymous referees for making several very helpful suggestions.

#### REFERENCES

- [1] A. C. Bovik, *Handbook of Image and Video Processing (Communications, Networking and Multimedia)*. Orlando, FL, USA: Academic Press, Inc., 2005. **1**
- [2] M. Nikolova, “A variational approach to remove outliers and impulse noise,” *Journal of Mathematical Imaging and Vision*, vol. 20, pp. 99–120, 2004. **1**
- [3] L. Bar, N. A. Sochen, and N. Kiryati, “Image deblurring in the presence of salt-and-pepper noise,” in *Scale-Space*, ser. Lecture Notes in Computer Science, R. Kimmel, N. A. Sochen, and J. Weickert, Eds., vol. 3459. Springer, 2005, pp. 107–118. **1**
- [4] L. Bar, N. Kiryati, and N. Sochen, “Image deblurring in the presence of impulsive noise,” *International Journal of Computer Vision*, vol. 70, pp. 279–298, 2006. **1**
- [5] G. Gilboa and S. Osher, “Nonlocal operators with applications to image processing,” *Multiscale Modeling & Simulation*, vol. 7, no. 3, pp. 1005–1028, 2008. **1**
- [6] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*. V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York,, 1977. **2**
- [7] D. Geman and G. Reynolds, “Constrained restoration and the recovery of discontinuities,”



- IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 3, pp. 367–383, 1992. [2](#)
- [8] L. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physics D*, vol. 60, pp. 259–268, 1992. [2](#)
- [9] L. I. Rudin and S. Osher, “Total variation based image restoration with free local constraints,” in *Proceedings of 1st International Conference on Image Processing*, vol. 1. IEEE Comput. Soc. Press, 1994, pp. 31–35. [2](#)
- [10] J. Cai, R. H. Chan, and Z. Shen, “A framelet-based image inpainting algorithm,” *Applied and Computational Harmonic Analysis*, vol. 24, pp. 131–149, 2008. [2](#)
- [11] Y. Li, L. Shen, D. Dai, and B. Suter, “Framelet algorithms for de-blurring images corrupted by impulse plus Gaussian noise,” *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1822–1837, 2011. [2](#)
- [12] R. Chan, C. Hu, and M. Nikolova, “An iterative procedure for removing random-valued impulse noise,” *IEEE Signal Processing Letters*, vol. 11, no. 12, pp. 921–924, 2004. [2](#)
- [13] R. H. Chan, C.-W. Ho, and M. Nikolova, “Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization,” *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1479–1485, 2005. [2](#)
- [14] J. Cai, R. H. Chan, and M. Nikolova, “Two-phase approach for deblurring images corrupted by impulse plus Gaussian noise,” *Inverse Problem and Imaging*, vol. 2, pp. 187–204, 2008. [2](#)
- [15] —, “Fast two-phase image deblurring under impulse noise,” *Journal of Mathematical Imaging and Vision*, vol. 36, pp. 46–53, 2010. [2](#)
- [16] Y. Xiao, T. Zeng, J. Yu, and M. Ng, “Restoration of images corrupted by mixed Gaussian-impulse noise via  $l_1$ - $l_0$  minimization,” *Pattern Recognition*, vol. 44, pp. 1708–1720, 2011. [2](#)
- [17] P. Rodríguez, R. Rojas, and B. Wohlberg, “Mixed gaussian-impulse noise image restoration via total variation,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 1077–1080. [2](#)
- [18] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '00, 2000, pp. 417–424. [2](#)
- [19] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, “Simultaneous structure and texture image inpainting,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2003, p. 707. [2](#)
- [20] M. Bertalmio, L. A. Vese, G. Sapiro, and S. Osher, “Simultaneous structure and texture image inpainting,” *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 882–889, 2003. [2](#)
- [21] T. F. Chan, J. Shen, and H. Zhou, “Total variation wavelet inpainting,” *Journal of Mathematical Imaging and Vision*, vol. 25, pp. 107–125, 2006. [2](#)
- [22] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001. [2](#), [11](#)
- [23] T. Chen and H. R. Wu, “Adaptive impulse detection using center-weighted median filters,” *IEEE Signal Processing Letters*, vol. 8, no. 1, pp. 1–3, 2001. [2](#), [12](#)
- [24] Y. Dong, R. H. Chan, and S. Xu, “A detection statistic for random-valued impulse noise,” *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 1112–1120, 2007. [2](#), [12](#)
- [25] B. Dong, H. Ji, J. Li, Z. Shen, and Y. Xu, “Wavelet frame based blind image inpainting,” *Applied and Computational Harmonic Analysis*, vol. 32, pp. 268–279, 2012. [2](#), [10](#), [12](#), [14](#)
- [26] T. Goldstein and S. Osher, “The split Bregman method for L1-regularized problems,” *SIAM Journal on Imaging Sciences*, vol. 2, pp. 323–343, 2009. [2](#), [11](#)
- [27] R. Chartrand and V. Staneva, “Restricted isometry properties and nonconvex compressive sensing,” *Inverse Problems*, vol. 24, no. 3, p. 035020, 2008. [2](#)
- [28] H. Mohimani, M. Babaie-Zadeh, I. Gorodnitsky, and C. Jutten, “Sparse recovery using smoothed  $l_0$  (sl0): convergence analysis,” *ArXiv preprint arXiv:1001.5073*, 2010. [2](#)
- [29] Z. Lu and Y. Zhang, “Penalty Decomposition Methods for  $l_0$ -Norm Minimization,” *ArXiv preprint arXiv:1008.5372*, 2010. [2](#)
- [30] T. Blumensath and M. Davies, “Iterative Thresholding for Sparse Approximations,” *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 629–654, Dec. 2008. [2](#)
- [31] Y. Zhang, B. Dong, and Z. Lu, “ $l_0$  minimization of wavelet frame based image restoration,” *Mathematics of Computation*, vol. accepted, 2011. [2](#)
- [32] B. Dong and Y. Zhang, “An efficient algorithm for  $l_0$  minimization in wavelet frame based image restoration,” *Journal of Scientific Computing*, vol. 54, no. 2-3, pp. 350–368, 2013. [2](#)
- [33] Z. Lu and Y. Zhang, “Sparse Approximation via Penalty Decomposition Methods,” *ArXiv preprint arXiv:1205.2334*, 2012. [2](#), [13](#)

- [34] P. Tseng, “Convergence of a block coordinate descent method for nondifferentiable minimization,” *J. Optim Theory Appl*, pp. 475–494, 2001. [5](#)
- [35] T. Chan and J. Shen, “Variational image inpainting,” *Communications on Pure and Applied Mathematics*, vol. 58, pp. 579–619, 2005. [5](#)
- [36] M. Yan, Y. Yang, and S. Osher, “Robust 1-bit compressive sensing using adaptive outlier pursuit,” *IEEE Transactions on Signal Processing*, pp. 3868–3875, 2012. [6](#)
- [37] —, “Exact low-rank matrix completion from sparsely corrupted entries via adaptive outlier pursuit,” *Journal of Scientific Computing*, p. to appear, 2013. [6](#)
- [38] Y. Wang, A. Szlam, and G. Lerman, “Robust locally linear analysis with applications to image denoising and blind inpainting,” *SIAM Journal on Imaging Sciences*, to appear. [10](#)
- [39] J. Yang, “A tv-based approach to blind image inpainting,” in *Image and Signal Processing (CISP), 2011 4th International Congress on*, vol. 2, 2011, pp. 779–781. [10](#)
- [40] T. F. Chan and J. Shen, “Mathematical models for local nontexture inpaintings,” *SIAM Journal of Applied Mathematics*, vol. 62, no. 3, pp. 1019–1043, 2002. [11](#)
- [41] T. Chan and J. Shen, *Image Processing And Analysis: Variational, PDE, Wavelet, And Stochastic Methods*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2005. [11](#)
- [42] A. Chambolle, “An algorithm for total variation minimization and applications,” *Journal of Mathematical Imaging and Vision*, vol. 20, pp. 89–97, 2004. [11](#)
- [43] M. Zhu, S. J. Wright, and T. F. Chan, “Duality-based algorithms for total-variation-regularized image restoration,” *Computational Optimization and Applications*, vol. 47, pp. 377–400, 2010. [11](#)
- [44] X.-C. Tai and C. Wu, “Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model,” in *Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision*, 2009, pp. 502–513. [11](#)
- [45] C. Wu and X.-C. Tai, “Augmented lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models,” *SIAM Journal on Imaging Sciences*, vol. 3, pp. 300–339, 2010. [11](#)
- [46] P. Getreuer, “tvreg v2: Variational imaging methods for denoising, deconvolution, inpainting, and segmentation,” 2010. [11](#)
- [47] H. Hwang and R. A. Haddad, “Adaptive median filters: new algorithms and results,” *IEEE Transactions on Image Processing*, vol. 4, no. 4, pp. 499–502, 1995. [11](#)