# Wavelet Frame Based Blind Image Inpainting

Bin Dong[b], Hui Ji[a,*], Jia Li[a], Zuowei Shen[a], Yuhong Xu[c]

[a] *Department of Mathematics, National University of Singapore, Singapore, 117542*
[b] *Department of Mathematics, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA, 92093-0112*
[c] *Center for Wavelets, Approx. and Info. Proc., National University of Singapore, Singapore, 117542*

**Abstract**

Image inpainting has been widely used in practice to repair damaged/missing pixels of given images. Most of the existing inpainting techniques require knowing beforehand where those damaged pixels are, either given as a priori or detected by some pre-processing. However, in certain applications, such information neither is available nor can be reliably pre-detected, *e.g.* removing random-valued impulse noise from images or removing certain scratches from archived photographs. This paper introduces a blind inpainting model to solve this type of problems, i.e., a model of simultaneously identifying and recovering damaged pixels of the given image. A tight frame based regularization approach is developed in this paper for such blind inpainting problems, and the resulted minimization problem is solved by the split Bregman algorithm first proposed by [1]. The proposed blind inpainting method is applied to various challenging image restoration tasks, including recovering images that are blurry and damaged by scratches and removing image noise mixed with both Gaussian and random-valued impulse noise. The experiments show that our method is compared favorably against many available two-staged methods in these applications.

*Key words:* image inpainting, sparse approximation, split Bregman algorithm, wavelet frame

## 1. Introduction

The word "inpainting" has been used by museum restoration artists for quite a while before the concept was applied to digital image inpainting first by [2]. Image inpainting problem occurs when image pixels are missing, over-written or damaged by some other means. This arises for example in restoring ancient

*Corresponding author. Tel.: +65-65168845, Fax: +65-67795452
*Email addresses:* dongbin725@gmail.com (Bin Dong), matjh@nus.edu.sg (Hui Ji), lijia@nus.edu.sg (Jia Li), matzuows@nus.edu.sg (Zuowei Shen), tslxyh@nus.edu.sg (Yuhong Xu)

drawings or old videos where a portion of a picture or frame is missing or damaged due to aging or scratching; or when an image is corrupted by impulse noise due to noisy sensors or channel transmission error. Thus, image inpainting is about recovering the missing information within the damaged regions from the incomplete and noisy observation of the image. An ideal recovery of an image in the corrupted regions should possess image features, like edges and textures, that are consistent to the features observed. In recent years, there have been great progresses on image inpainting, see [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14] and the references therein.

The model for general image inpainting problem, accompanied by other image degradation effects such as blurring, can be formulated as follows. Let an image be represented as a column vector in $\mathbb{R}^n$, where $n$ is the total number of pixels. Then the mathematical formulation of imaging inpainting can be expressed as

$$f(i) = \begin{cases} (Hu)(i) + \epsilon(i), & i \in \Lambda \\ v(i), & i \in \Lambda^c, \end{cases} \tag{1.1}$$

where $f$ is the observed corrupted image, $u$ is the original image that we are trying to recover, $\epsilon$ represents i.i.d. additive Gaussian white noise ([15]), $H$ is some degradation operator (e.g. convolution for image blurring), and $\Lambda$ is the index set of certain pixels of the image. The random-valued vector $v$ represents the values of all other image pixels corrupted by other factors (besides Gaussian white noise). The sub-vector $v|_{\Lambda^c}$ of $v$ defined on $\Lambda^c$ can represent various types of degradations to the original image, including impulse noise (e.g. salt-and-pepper noise and random-valued impulse noise) and random-patterned scratching with unknown intensities. The goal of image inpainting is then to estimate the original image $u$ from the observation $f$.

The index set $\Lambda^c$ is called the inpainting region/domain which is assumed to be known or estimated beforehand in most literatures. However, in some applications, the inpainting domain may not be readily available, or it can not be accurately detected by some separate process, *e.g.* when the vector $v$ in (1.1) contains random-valued impulse noise mixed with Gaussian white noise. We call such a inpainting problem a *blind inpainting* problem, that is, both the inpainting domain $\Lambda^c$ (or equivalently the projection $P_\Lambda$) and the original image $u$ are unknown. In contrast to regular image inpainting problems in which $\Lambda$ is known, both $\Lambda$ and $u$ are unknown in blind inpainting problem. Thus, blind inpainting problem is a highly ill-posed inverse problem.

In recent years, there have been a few works on solving such blind inpainting problems. For video restoration, the blind inpainting problem is first discussed in [16]. Based on the same sparsity prior of damaged pixels in image space, a patch-based approach is proposed in [16] to do blind video inpainting. For image restoration, the blind inpainting problem for repairing damaged pixels is tackled by [17] and [18], both of which use the $\ell_1$ norm as the fidelity measurement to suppress the outlier effect of damaged pixels. The differences between them is one is using non-local total variation (TV) regularization ([17]) and the other is using $\ell_1$ norm of tight frame coefficients ([18]). Recently, a TV-based approach

is proposed [19] to simultaneously identify occlusions and estimating optical flow of a video sequence, which also can be viewed as a blind inpainting problem.

The goal of this paper is to develop computational models and corresponding efficient algorithms for solving such blind image inpainting problems. In this paper, we take a Lagrangian regularization approach to tackle such an ill-posed inverse problem. In order to overcome the ill-posedness of the problem, appropriate regularization terms on both the original image $u$ and the inpainting region $\Lambda^c$ have to be enforced in the minimization model. The basic idea of our approach is to utilize the sparsity priors of images and random-valued vector $v$ in different domains. Motivated by the impressive performance of using sparsity prior of images under tight wavelet frames in many image restoration tasks ([20, 21, 22, 23, 24, 25, 26, 27, 11, 12, 13]), we also use the $\ell_1$ norm of wavelet tight frame coefficients of images as the regularization term for images in our approach. Since the values of $v|_\Lambda$ can be arbitrary, we seek the solution to $v$ whose elements defined on the index $\Lambda$ are zeros. Since the number of the inpainting domain $\Lambda^c$ is usually much less than that of the whole image domain, we assume that $v$ is sparse in spatial domain. Such a assumption is utilized in our approach by using the $\ell_1$ norm of $v$ in spatial domain as the regularization term on $v$. Moreover, the $\ell_1$ related minimization resulting from our proposed models can be efficient solved using the so-called split Bregman method. The split Bregman iteration is first proposed in [1] with many successful applications in imaging sciences (see *e.g.* [1, 28, 29]).

The rest of this paper will be organized as follows. In Section 2, we will propose two sparsity-based regularization models for blind image inpainting. In Section 3, the split Bregman iteration based algorithms will be applied to solve the minimization problems resulted from the proposed models. Numerical experiments will be conducted in Section 4 for three image restoration tasks:

1. image denoising for random-valued impluse noise mixed with Gaussian noise.

2. image deblurring in the presence of both random-valued impulse noise and Gaussian white noise.

3. blind image inpainting for images corrupted by multiple sources including random-valued impulse noise, Gaussian white noise and scratching.

## 2. Frame Based Blind Inpainting Models

Before presenting our regularization models for blind image inpainting, we briefly review a few facts of discrete tight wavelet frame decomposition and reconstruction. Interesting readers should consult [30, 31, 32] for theories of frames and framelets, [12] for a short survey on theory and applications of frames, and [13] for a more detailed survey. In the discrete setting, a 2-dimensional image is a 2-dimensional array that can be understood as a vector living in $\mathbb{R}^n$, with $n$ the total number of pixels in the image. Then the discrete framelet decomposition and reconstruction can be represented as matrix multiplications $Wu$

and $W^\top v$ respectively. Here $W \in \mathbb{R}^{k\times n}$ satisfies $W^\top W = I$, i.e. $u = W^\top W u$, $\forall u \in \mathbb{R}^n$, where $W$ is derived by the filters of framelets obtained by the unitary extension principle [31]. The matrix multiplications by $W$ and $W^\top v$ are only for the notational convenience. In our numerical implementations, these two matrix multiplications are done by using the fast tensor product tight wavelet frame decomposition and reconstruction algorithms instead, which are essentially just the convolution of images by a set of filters. Interested readers can refer to [13, 27] for more details.

*2.1. Blind Inpainting Model: Single System*

For notational convenience, we denote the projection matrix $P_\Lambda$ associated to each $\Lambda$ as an $n \times n$ diagonal matrix with the diagonal entries 1 for the indices in $\Lambda$ and 0 otherwise. Under this notation, (1.1) can be written equivalently as

$$P_\Lambda f = P_\Lambda(Hu + \epsilon) \quad \text{and} \quad P_{\Lambda^c} f = P_{\Lambda^c} v. \tag{2.1}$$

We propose the following model to solve the blind image inpainting problem (2.1)

$$\min_{u,v} \ \frac{1}{2}\|Hu + v - f\|_2^2 + \lambda_1 \|Wu\|_1 + \lambda_2 \|v\|_1, \tag{2.2}$$

where $u$ is the ideal image that we are trying to recover, $v$ is the random-valued vector in the observed image $f$, $H$ is some degradation matrix, and $W$ is the decomposition matrix associated to some tight framelet system. The basic idea of (2.2) is as follows. Besides the variable $u$ that represents the unknown true image, we introduce a new variable $v$ in the fidelity term. The role of this variable is to explicitly represents the outliers (pixels damaged by impulse noise) existing in $f$. Certain regularizations on both variable $u$ and $v$ are needed to solve the ill-posed linear system $Hu + v = f$. The proposed regularization on the true image $u$ is based on the assumption that a clear noise-free image $u$ should have a sparse approximation under wavelet tight frame domain. The proposed regularization on the variable $v$ is based on the assumption that the percentage of pixels damaged by impulse noise is small, which is equivalent to say that the vector $v$ is sparse with only a small percentage of non-zero elements. As a convex relaxation of $\ell_0$-norm that measures the exact sparsity of the signal, $\ell_1$-norm is used on both $Wu$ and $v$ in (2.2) to measure their sparsities. For the case $H$ is the identity, a similar model already appeared in [17] with non-local total variation (TV) regularization.

In our proposed approach, the vector $v$ is explicitly defined as an unknown to be estimated in the optimization model. There are a few alternative approaches to handle the random-valued vector $v$. One of them appears in [11, 18] is the two-stage approach that estimates the inpainting region $\Lambda$ before estimating $u$. Then the problem (2.1) is reduced to a regular inpainting problem:

$$\min_{u} \ \frac{1}{2}\|P_\Lambda(Hu - f)\|_2^2 + \lambda \|Wu\|_1. \tag{2.3}$$

Such a two-stage approach works well when an accurate detection of $\Lambda$ is possible, *e.g.* detecting salt-and-pepper noise using adaptive median filter ([33]).

However, it is much harder to accurately detect general random-valued impulse noise in images. Such unavoidable detection errors of $\Lambda$ could seriously hamper the performance of the inpainting if using (2.3), as we will see in our experiments.

The other approach is first proposed by [18, 34, 35] that treats $v$ as the outliers and uses the $\ell_1$ norm in the fidelity term to increase the robustness of inpainting to outliers:

$$\min_u \quad \|Hu-f\|_1+\lambda\|Wu\|_1. \tag{2.4}$$

This model can also be applied to the blind image inpainting problems. When there exists only random-valued impulsive noise, the performance of the regularization model (2.4) is similar to the proposed model (2.2) as we observed in the experiments. In practice, however, image noise is hardly from a single source. Five major sources of image noise with different statistical distributions have been identified in [36] including amplifier noise modeled by Gaussian noise. In the presence of noises from multiple sources such as the mixture of impulse noise and Gaussian noise, the model (2.2) is better than (2.4). The reason is as follows. It is known that the $\ell_2$ norm based fidelity term yields the optimal estimate in the presence of only Gaussian noise. Although the $\ell_1$ norm based fidelity term used in (2.4) suppresses the negative impact caused by the outliers, the adverse effect is the less optimal usage of the other data polluted by mostly Gaussian noise. On the contrary, the proposed model (2.2) uses those data damaged by mostly Gaussian noise in a optimal way while avoiding the outlier effect of other data. The experimental evaluation in Section 4 will also justify the advantage of the regularization (2.2) over (2.4).

### *2.2. Blind Inpainting Model: Two Systems*

The success of the model (2.2) largely relies on the validity of two sparsity assumptions: one is the sparsity of images in wavelet tight frame domain and the sparsity of $v$ in image domain. Thus, the model (2.2) is suitable for images that are piecewise smooth. When there exists rich texture information in images, these texture features are no longer piecewise smooth and they are relatively sparse in image domain ([37, 38, 39]). As a result, it is possible in (2.2) that these texture features are wrongly identified as the elements of $v$ instead of being preserved in $u$. Thus, the model (2.2) needs to be modified to prevent the texture features from being absorbed into the vector $v$.

Our proposed approach is based on the observation that many types of textures can be sparsely approximated by local discrete cosine transform (local DCT), which has been successfully used in some image restoration tasks ([25, 10, 22]). Motivated by these approaches, we propose the following model for blind inpainting problem:

$$\min_{u_1,u_2,v} \quad \frac{1}{2}\|H(u_1+u_2)+v-f\|_2^2+\lambda_1\|Wu_1\|_1+\lambda_2\|v\|_1+\lambda_3\|Du_2\|_1. \tag{2.5}$$

Here $D$ represents the local DCT, $u_1$ is the cartoon component and $u_2$ is the texture component, and the desired recovery $u = u_1 + u_2$. The model (2.5) provided a more accurate estimate of $u$ than (2.2) does for images of rich texture information. However, the complexity of the resulting minimization problem is also increased, which leads to a higher computational cost. Thus, both models have their own merits and the choice of using either one of them should depends on the image content. The model (2.2) is more suitable for images without rich textures and the model (2.5) is more suitable for images with rich textures.

## 3. Numerical Algorithms

In this section, we will present numerical algorithms that solves our proposed models (2.2) and (2.5), as well as (2.4) and (2.3) used for experimental evaluation. All of these algorithms are built upon the split Bregman iteration. The split Bregman algorithm was first proposed in [1] which showed its efficiency applied to various PDE based image restoration models , e.g., ROF and nonlocal variational models ([1, 40]). Convergence analysis of the split Bregman algorithm, as well as its applications in various wavelet frame based image restoration algorithms, were given in [22]. For the completeness, we give a brief introduction of the basic idea of split Bregman algorithm. Interesting readers are referred to [1, 22] for more details.

Consider the following minimization problem

$$\min_u \quad E(u) + \lambda \|Lu\|_1, \tag{3.1}$$

where $E(u)$ is a smooth convex functional and $L$ is some linear operator. Let $d = Lu$. Then (3.1) can be rewritten as

$$\min_{u, d=Lu} \quad E(u) + \lambda \|d\|_1. \tag{3.2}$$

Note that both $u$ and $d$ are variables now. The derivation of splitting Bregman iteration for solving (3.2) is based on Bregman distance ([1, 22]). It was recently shown (see e.g. [41, 42]) that the split Bregman algorithm can also be derived by applying augmented Lagrangian method (see e.g. [43]) to (3.2). The connection between split Bregman algorithm and Douglas-Rachford splitting was addressed by [44]. We shall skip the detailed derivations and directly describe the split Bregman algorithm that solves (3.1) through (3.2) as follows,

$$\begin{cases} u^{k+1} = \arg\min_u E(u) + \frac{\mu}{2}\|Lu - d^k + b^k\|_2^2, \\ d^{k+1} = \arg\min_d \lambda\|d\|_1 + \frac{\mu}{2}\|d - Lu^{k+1} - b^k\|_2^2, \\ b^{k+1} = b^k + Lu^{k+1} - d^{k+1}. \end{cases} \tag{3.3}$$

By [45, 46], the second subproblem has a simple analytical solution based on soft-thresholding operator. Therefore, (3.3) can be written equivalently as

$$\begin{cases} u^{k+1} = \arg\min_u E(u) + \frac{\mu}{2}\|Lu - d^k + b^k\|_2^2, \\ d^{k+1} = \mathcal{T}_{\lambda/\mu}(Lu^{k+1} + b^k), \\ b^{k+1} = b^k + (Lu^{k+1} - d^{k+1}), \end{cases} \tag{3.4}$$

where $\mathcal{T}_\theta$ is the soft-thresholding operator defined by

$$\mathcal{T}_\theta : x = [x_1, x_2, \cdots, x_M] \to \mathcal{T}_\theta(x) = [t_{\theta_1}(x_1), t_{\theta_2}(x_2), \cdots, t_{\theta_M}(x_M)],$$

with

$$t_{\theta_i}(x_i) = \mathrm{sgn}(x_i)\max\{0, |x_i| - \theta_i\}.$$

Note that the last two steps of (3.4) are straightforward and very efficient to compute, while the computation cost of the first step is usually more expensive as it involves the procedure of solving some linear system.

*3.1. Algorithms Solving Our Models* (2.2) *and* (2.5)

The minimization (2.2) can be efficiently solved split Bregman iteration. Let $d = Wu$ and rewrite (2.2) as

$$\min_{u,v,d=Wu} \quad \frac{1}{2}\|Hu + v - f\|_2^2 + \lambda_1\|d\|_1 + \lambda_2\|v\|_1.$$

Then we have the following iterative schemes that solves the above optimization problem:

$$\begin{cases} u^{k+1} = \arg\min_u \frac{1}{2}\|Hu + v^k - f\|_2^2 + \frac{\mu}{2}\|Wu - d^k + b^k\|_2^2, \\ v^{k+1} = \arg\min_v \lambda_2\|v\|_1 + \frac{1}{2}\|v - (f - Hu^{k+1})\|_2^2, \\ d^{k+1} = \arg\min_d \lambda_1\|d\|_1 + \frac{\mu}{2}\|d - (Wu^{k+1} + b^k)\|_2^2, \\ b^{k+1} = b^k + (Wu^{k+1} - d^{k+1}). \end{cases}$$

The complete description of the algorithm for solving (2.2) is provided in Algorithm 1.

**Algorithm 1** Numerical algorithm for solving (2.2)

(i) Set initial guesses $u^0$, $v^0$, $d^0$, $b^0$. Choose an appropriate set of parameters $(\lambda_1, \lambda_2, \mu)$.

(ii) For $k = 0, 1, \ldots$, perform the following iterations until convergence

$$\begin{cases} u^{k+1} = (H^\top H + \mu W^\top W)^{-1}\left(H^\top(f - v^k) + \mu W^\top(d^k - b^k)\right), \\ v^{k+1} = \mathcal{T}_{\lambda_2}(f - Hu^{k+1}), \\ d^{k+1} = \mathcal{T}_{\lambda_1/\mu}(Wu^{k+1} + b^k), \\ b^{k+1} = b^k + (Wu^{k+1} - d^{k+1}). \end{cases} \tag{3.5}$$

In our numerical experiments, the initializations are $u^0 = v^0 = d^0 = b^0 = 0$. The stopping criteria is

$$\|d^k - Wu^k\|_2 \le \epsilon.$$

Because of the linear system of the first equation in (3.5) is positive definite and sparse, we will use conjugate gradient (CG) method to solve the linear

**Algorithm 2** Fast algorithm for solving (2.5)

(i) Set initial guesses $u_1^0$, $u_2^0$, $v^0$, $d_1^0$, $b_1^0$, $d_2^0$, $b_2^0$. Choose an appropriate set of parameters $(\lambda_1,\lambda_2,\lambda_3,\mu_1,\mu_2)$.

(ii) For $k=0,1,\ldots$, perform the following iterations until convergence

$$\begin{cases} {u_1}^{k+1} = (H^\top H+\mu_1 W^\top W)^{-1}\left(H^\top(f-Hu_2^k-v^k)+\mu_1 W^\top(d_1^k-b_1^k)\right), \\ {u_2}^{k+1} = (H^\top H+\mu_2 D^\top D)^{-1}\left(H^\top(f-Hu_1^{k+1}-v^k)+\mu_2 D^\top(d_2^k-b_2^k)\right), \\ v^{k+1} = \mathcal{T}_{\lambda_2}(f-H({u_1}^{k+1}+{u_2}^{k+1})), \\ d_1^{k+1} = \mathcal{T}_{\lambda_1/\mu_1}(W{u_1}^{k+1}+b_1^k), \\ b_1^{k+1} = b_1^k+(W{u_1}^{k+1}-d_1^{k+1}), \\ d_2^{k+1} = \mathcal{T}_{\lambda_3/\mu_2}(D{u_2}^{k+1}+b_2^k), \\ b_2^{k+1} = b_2^k+(D{u_2}^{k+1}-d_2^{k+1}). \end{cases} \tag{3.6}$$

equations. In practice, we will not solve the first equation of (3.5) accurately but only run a few iterations of CG method.

The algorithm for solving the two-systems model (2.5) is similar to that for the single-system model (2.2). We will skip the details and directly present the detailed algorithm in Algorithm 2. In our numerical experiments, the initialization of Algorithm 2 is set to be $u_i^0=d_i^0=b_i^0=v^0=0$ for $i=1,2$. The stopping criteria is

$$\|{d_1}^k-W{u_1}^k\|_2+\|{d_2}^k-D{u_2}^k\|_2\le\epsilon.$$

Conjugate gradient method is also used to solve $u_1$ and $u_2$ in each iteration of (3.6). Similarly, only a few iterations of CG method is carried on when solving the linear systems.

*3.2. Algorithms Solving* (2.3) *and* (2.4)

The split Bregman iteration can be directly applied to solve (2.3) by rewritting (2.3) as follows:

$$\min_{u,d=Wu}\quad \frac{1}{2}\|P_\Lambda(Hu-f)\|_2^2+\lambda\|d\|_1.$$

Let $E(u)=\frac{1}{2}\|P_\Lambda(Hu-f)\|_2^2$ and let $L=W$ in (3.4). The detailed algorithm for solving (2.3) is given in Algorithm 3. In our implementation, $u^0=0$ and $d^0=b^0=0$ in the initialization. The stopping criteria is the same as Algorithm 1 and Algorithm 2:

$$\|d^k-Wu^k\|\le\epsilon.$$

The idea of split Bregman algorithm can also applied to (2.4) with small modifications. Notice that $E(u)=\|Hu-f\|_1$ in (2.4) is not differentiable. Thus,

**Algorithm 3** Fast algorithm for solving (2.3)

(i) Set initial guesses $u^0$, $d^0$, $b^0$. Choose an appropriate set of parameters $(\lambda,\mu)$.

(ii) For $k=0,1,\ldots$, perform the following iterations until convergence

$$\begin{cases} u^{k+1} := (H^T P_\Lambda H + \mu W^T W)u = H^T P_\Lambda f + \mu W^T (d^k - b^k), \\ d^{k+1} := \mathcal{T}_{\lambda/\mu}(Wu^{k+1} + b^k), \\ b^{k+1} := b^k + (Wu^{k+1} - d^{k+1}). \end{cases} \tag{3.7}$$

we need to introduce an additional splitting step (See also [47, 41]). First, rewrite the problem (2.4)as follows:

$$\min_u \ \|Hu - f\|_1 + \lambda \|Wu\|_1.$$

Let $d_1 = Wu$ and $d_2 = Hu - f$. Then (2.4) can be re-written as

$$\min_{u,d_1,d_2} \left\{ \|d_1\|_1 + \lambda \|d_2\|_1 : \ d_1 = Hu - f, \ d_2 = Wu \right\}. \tag{3.8}$$

Now we have the following split Bregman iterations that solves (3.8) and hence solves (2.4):

$$\begin{cases} u^{k+1} = \arg\min_u \frac{\mu_1}{2} \|Hu - f - d_1^k + b_1^k\|_2^2 + \frac{\mu_2}{2} \|Wu - d_2^k + b_2^k\|_2^2, \\ d_1^{k+1} = \arg\min_{d_1} \|d_1\|_1 + \frac{\mu_1}{2} \|d_1 - (Wu^{k+1} - f + b_1^k)\|_2^2, \\ d_2^{k+1} = \arg\min_{d_2} \lambda \|d_2\|_1 + \frac{\mu_2}{2} \|d_2 - (Wu^{k+1} + b_2^k)\|_2^2, \\ b_1^{k+1} = b_1^k + (Hu^{k+1} - f - d_1^{k+1}), \\ b_2^{k+1} = b_2^k + (Wu^{k+1} - d_2^{k+1}). \end{cases}$$

See Algorithm 4 for the complete description. In our implementation, the variables are initialized as follows: $u^0 = 0$, $d_1^0 = b_1^0 = 0$ and $d_2^0 = b_2^0 = 0$ . Similar to other algorithms, the stopping criteria is

$$\|d_1^k - Hu^k + f\|_2 + \|d_2^k - Wu^k\|_2 \le \epsilon.$$

## 4. Related Applications and Experimental Evaluation

As we discussed in Section 2, both (2.2) and (2.5) are capable of solving blind inpainting problem by simultaneously detecting inpainting region and recovering damaged pixels of images. The model (2.2) is a good choice for recovering images with less texture content and the model (2.5) works better for image with rich textures due to its additional term that protects textures. In this section, we will apply our blind inpainting algorithms in the following three applications:

**Algorithm 4** Fast algorithm for solving (2.4)

(i) Set initial guesses $u^0$, $d_1^0$, $b_1^0$, $d_2^0$, $b_2^0$. Choose an appropriate set of parameters $(\lambda,\mu_1,\mu_2)$.

(ii) For $k=0,1,\ldots$, perform the following iterations until convergence

$$\begin{cases} u^{k+1}=(\mu_1 H^\top H+\mu_2 W^\top W)^{-1}\left(\mu_1 H^\top(d_1^k-b_1^k+f)+\mu_2 W^\top(d_2^k-b_2^k)\right),\\ d_1^{k+1}=\mathcal{T}_{1/\mu_1}(Hu^{k+1}-f+b_1^k),\\ d_2^{k+1}=\mathcal{T}_{\lambda/\mu_2}(Wu^{k+1}+b_2^k),\\ b_1^{k+1}=b_1^k+(Hu^{k+1}-f-d_1^{k+1}),\\ b_2^{k+1}=b_2^k+(Wu^{k+1}-d_2^{k+1}). \end{cases} \tag{3.9}$$

(1) Removing random-valued impulse noise from images;

(2) Image deblurring in the presence of random-valued impulse noise;

(3) Blind inpainting for images damaged by multiple factors including scratches, impulse noise and etc.

It is noted that through our experiments, besides the above-mentioned degradations, all images are also degraded by additional additive Gaussian amplifier, since amplifier noise is commonly seen in most image noises. In the applications (1) and (3), $H$ is set as identical matrix for the models (2.2) and (2.5), i.e. images are not blurred. In the application (2), $H$ is set to be the out-of-focus blurring kernel with radius 6 pixels. The results from our approaches are compared against that from the existing approaches (2.4) and (2.3). Also, the comparison between the proposed two models (2.2) and (2.5) are demonstrated to illustrate their performances for images with/without textures.

In our experiments, all the degradations by impulse noise and Gaussian noise are synthesized as follows.

$$f=N_p(Hu+\epsilon), \tag{4.1}$$

where $u$ is the original image before corruption and $f$ represents the corrupted image, $H$ is either identical operator or convolution operator, $\epsilon$ stands for i.i.d. Gaussian white noise with zero mean, and $N_p(x)$ is the operator that adds impulse noise to $x$. The operator $N_p$ is defined as follows:

*Random-valued impulsive noise:* a certain proportion of pixels (chosen randomly) are altered to be an uniformly random number in $[d_{min},\ d_{max}]$

$$N_p(x_{ij})=\begin{cases} d_{ij}, & \text{with probability } r,\\ x_{ij}, & \text{with probability } (1-r), \end{cases} \tag{4.2}$$

where $d_{ij}$ is a uniformly distribution random number in $[d_{min},\ d_{max}]$ and $r$ is the level of random valued noise.

The dynamic range of $f$ which is $[d_{min},\ d_{max}]$, is taken to be $[0,255]$. Besides the visual comparison of the results, the PSNR measurement is used to quantitatively evaluate the quality of the restoration results. Recall that given a signal $x$, the peak signal to noise ratio (PSNR) of its estimate is defined as

$$\mathrm{PSNR}(\widehat{x},x) = 10\log_{10}\frac{255^2}{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(\widehat{x}_{ij}-x_{ij})^2}.$$

where $m$ and $n$ are the dimensions of the image, $x_{ij}$ is the intensity value at the pixel location $(i,j)$, and $\widehat{x}_{ij}$ corresponds to the intensity value of the restored image at location $(i,j)$.

Through the numerical experiments, 100 iterations are executed in Algorithm 1, Algorithm 2 and Algorithm 4 when solving (2.2), (2.5) and (2.4), it takes approximately 60 seconds when running the matlab implementations of these two algorithms on a PC with 2Ghz Intel Core 2 CPU. When solving (2.3), 50 iterations of Algorithm 3 are executed which takes about 50 seconds in the same hardware setting.

### *4.1. Removing Random-valued Impulse Noise from Images*

Besides the most commonly seen Gaussian white noise that degrades images, impulse noise is also often seen in corrupted images due to transmission errors, faulty sensors and etc. There are mainly two types of impulse noise, one is salt-and-pepper noise and the other is random-valued impulse noise. Removing impulse noise from images is different from removing Gaussian noise as the values of damaged pixels contains no information of the truth at all. Thus, removing impulse noise is essentially an image inpainting problem. The pixels damaged by salt-and-pepper noise are much easier to find since their brightness values are either 0 or 255. The adaptive median filter has been widely used to accurately identify most pixels damaged by salt-and-pepper noise (See *e.g.* [48, 49, 29]). On the contrary, the detection of pixels damaged by random-valued impulse noise is much harder as the brightness value of damaged pixels can be arbitrary. The adaptive center-weighted median filter (ACWMF) was first proposed in [48] to detect pixels damaged by random-valued impulse noise. More recently, The ROLD detection method has been proposed by [50] with better accuracy on detecting such impulse noise. In our experiments, both detection techniques are used to provide the input needed for the two-stage method (2.3). It is noted that our proposed models do not require the input of inpainting regions from such a detection pre-process.

The parameters of the proposed denoising algorithms used in the experiments are set as follows. For the single system model (2.2), the value of the parameter $\lambda_1$ is dependent on the level of Gaussian white noise. The higher the Gaussian noise level, the larger the value of $\lambda_1$ should be. Through all experiments, the values of $\lambda_1$ is chosen from the set of $\{1.8,2,2.25,3\}$. The value of $\lambda_2$ is dependent on the impulse noise level and is chosen from $\{5,6\}$. The values of two parameters $\lambda_1,\lambda_2$ in (2.5) is set the same as that in (2.2). The value of the

parameter $\lambda_3$ in (2.5) is dependent on the percentage of textures in the given image. We set it to 1 for images with less textures and to 5 otherwise.

The PSNR values of the results from all six methods are summarized in Table 1 and 2. The visual comparison of some results are shown in Figure 1 and Figure 2. From the results, the ROLD detector clearly outperformed the ACWMF detector in terms of the accuracy of detecting pixels damaged by impulse noise. However, when impulse noise is mixed with Gaussian white noise, the detection reliability of ROLD detector noticeably decreases as the effect caused by Gaussian white noise is not considered in the design of the ROLD detector. It is seen that the results from our models (2.2) and (2.5) are noticeably better than (2.4) and (2.3) in terms of PSNR values, especially in the case of modest noise level. The visual inspection on Figure 1 and Figure 2 also leads to the same conclusion. When the impulse noise level is higher, the sparsity assumption of $v$ is less valid. As a result, the performance of our models (2.2) and (2.5) will decrease. However, they still manage to achieve modest gains in image quality, compared to the two-stage method. The results in this experiments clearly show the advantages of the models (2.2) and (2.5) by simultaneously identifying outlier and recovering damaged pixels.

It is noted that the proposed two models are not suitable for recovering image damaged by very high level of impulse noise, e.g, $r = 0.6$. The main reason is that the sparsity assumption on the impulse noise does not hold true anymore in such cases. To recover image with very high impulse noise level, the two-stage method with impulse noise detector such as ROLD may be a better choice. One possible solution is to first detect impulse noise in the image using ROLD in a conservative manner, then use the proposed methods to remove Gaussian noise and remained un-detected impulse noise from the image..

./

Table 1: PSNR value (dB) of the denoising results for cameraman image from all the three models from (2.3), (2.4) and (2.2) (our model 1), in the presence of random-valued impulse noise with ratio $r$ and Gaussian noise with std $\sigma$.

| Ratio $r$ and | r = 10% | | r = 20% | | r = 40% | |
|---|---|---|---|---|---|---|
| standard deviation | $\sigma$=0 | $\sigma$=10 | $\sigma$=0 | $\sigma$=10 | $\sigma$=0 | $\sigma$=10 |
| ROLD-ERR Model in [50] | 27.4 | 24.6 | 25.4 | 23.6 | 23.6 | 22.3 |
| Model (2.3) + ACWMF | 28.5 | 26.0 | 26.3 | 24.9 | 23.1 | 22.5 |
| Model (2.3) + ROLD | 28.4 | 27.5 | 26.3 | 25.8 | 23.7 | 23.3 |
| Model (2.4) | 29.9 | 27.5 | 27.1 | 26.0 | 23.1 | 22.9 |
| Model (2.2) | 30.3 | 28.4 | 27.4 | 26.6 | 23.6 | 23.3 |
| Model (2.5) | 30.3 | 28.4 | 27.4 | 26.6 | 23.6 | 23.3 |

### *4.2. Image Deblurring in the presence of Random-Valued Impulse Noise*

In this application, we applied our blind inpainting algorithm to images that not only contain random-valued impulse noise but are also blurry. Same as Section 4.1, we also assume the existence of Gaussian white noise in images. For image deblurring, $H$ in all models are now some convolution matrix and we

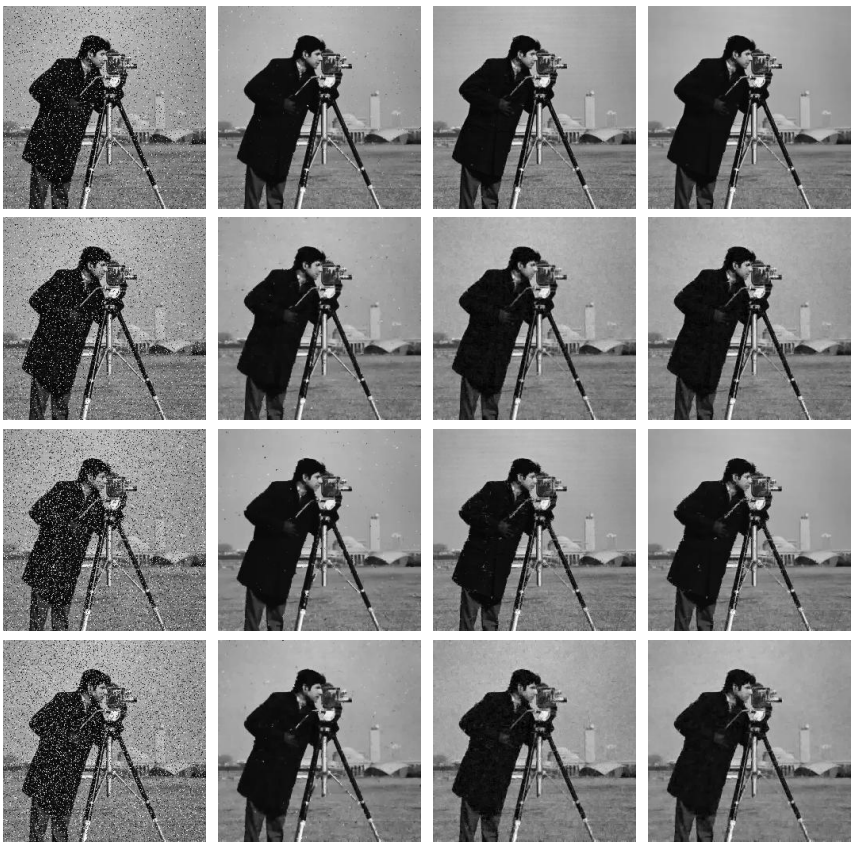


Figure 1: Denoising results of cameraman image contaminated by both random-valued impulse noise and Gaussian noise. Images in each column represent (from left to right) corrupted images, results from (2.3) combined with ROLD pre-detection, results from (2.4) and results from (2.2) respectively. The noise levels of corrupted images (from top to bottom) are as follows. (1) 10% random-valued impulse noise without Gaussian noise; (2) 10% random-valued impulse noise with Gaussian noise of $\sigma$=10; (3) 20% random-valued impulse noise without Gaussian noise (4) 20% random-valued impulse noise with Gaussian noise of $\sigma = 10$. The PSNR values of the results are given in Table 1.

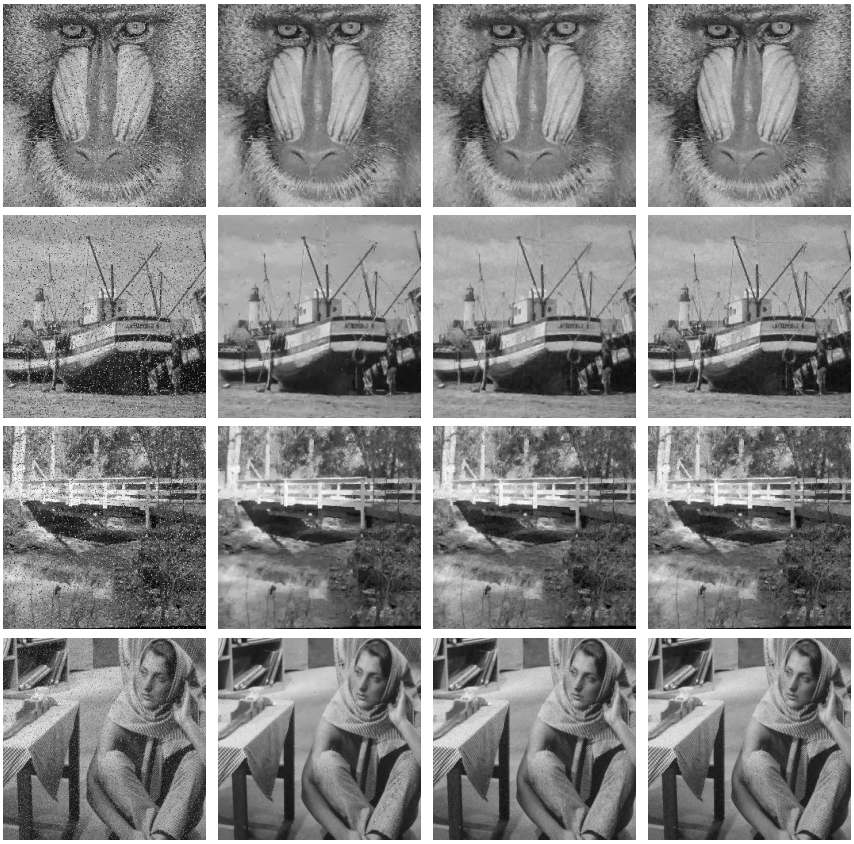


Figure 2: Denoising result of several images contaminated by random-valued impulse noise of rate 10% and Gaussian noise of $\sigma$=10. Images in each column represent (from left to right) corrupted images, results from (2.3) combined with ROLD pre-detection, results from (2.4) and results from (2.2) respectively. The PSNR values of the results are given in Table 2.

Table 2: PSNR value (dB) of the denoising results for other images from all the three models from (2.3), (2.4), (2.2) and (2.5), in the presence of random-valued impulse noise with ratio $r$ and Gaussian noise with std=10.

| Image and $r$ and ratio | Baboon | | Boat | | Bridge | | Barbara512 | |
|---|---|---|---|---|---|---|---|---|
| | 10% | 20% | 10% | 20% | 10% | 20% | 10% | 20% |
| ROLD-ERR Model in [50] | 23.0 | 21.6 | 24.7 | 23.8 | 23.3 | 22.1 | 25.3 | 23.9 |
| Model (2.3) + ACWMF | 23.3 | 22.2 | 26.6 | 25.1 | 24.2 | 22.9 | 26.0 | 24.6 |
| Model (2.3) + ROLD | 24.8 | 22.9 | 28.2 | 26.4 | 25.3 | 23.7 | 27.8 | 25.8 |
| Model from (2.4) | 24.5 | 23.2 | 27.6 | 26.1 | 25.0 | 23.4 | 27.0 | 25.5 |
| Model from (2.2) | 25.1 | 23.5 | 28.3 | 26.4 | 25.4 | 23.7 | 27.9 | 26.0 |
| Model from (2.5) | 25.2 | 23.5 | 28.2 | 26.4 | 25.4 | 23.7 | 27.9 | 26.0 |

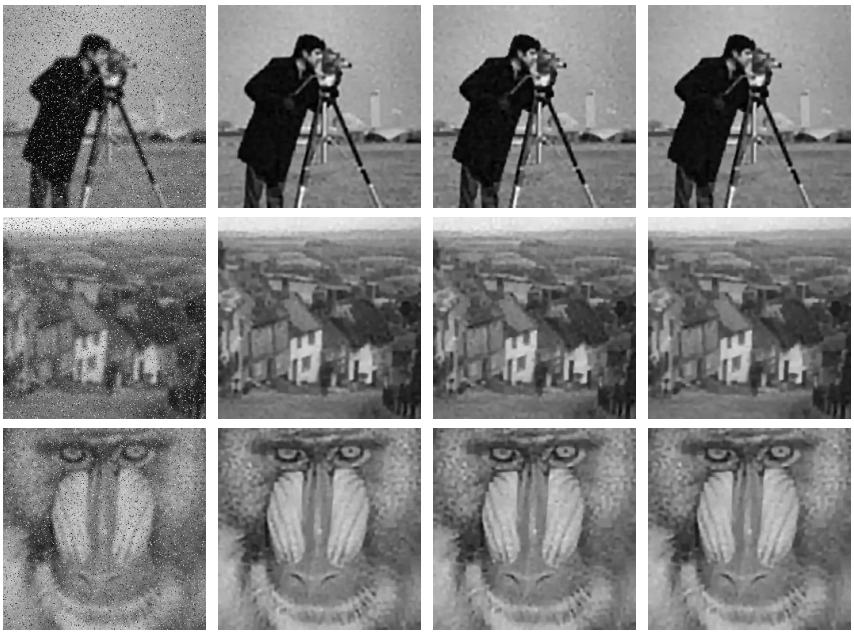


Figure 3: Deblurring result of several images in the presence of random-valued impulse noise of rate 10% and Gaussian noise of $\sigma$=10. Images in each column represent (from left to right) corrupted images, results from (2.3) combined with ROLD pre-detection, results from (2.4) and results from (2.2). The PSNR values of the results are given in Table 3.

use the out-of-focus kernel of radius 6 pixels. The values of parameters $\lambda_1, \lambda_2$ used in the models are dependent on the noise level and their values are chosen from the set $\{1, 10, 12\}$. The PSNR values of the results are summarized in Table 3, and the visual comparison of some results are shown in Figure 3. It is seen that the results from our models (2.2) and (2.5) are comparable to that from the other two models. Overall, there are not much differences among all five methods for image deblurring.

Table 3: PSNR value (dB) of the results from (2.3), (2.4), (2.2) and (2.5), for image deblurring in the presence of random-valued impulse noise and Gaussian noise.

| Image and $r$ and ratio | Cameraman | | Goldhill | | Baboon | |
|---|---|---|---|---|---|---|
| | 10% | 20% | 10% | 20% | 10% | 20% |
| Model (2.3) + ACWMF | 24.3 | 24.0 | 25.7 | 21.5 | 21.2 | 21.2 |
| Model (2.3) + ROLD | 24.3 | 24.1 | 25.8 | 21.6 | 21.3 | 21.2 |
| Model (2.4) | 24.1 | 23.9 | 25.5 | 21.2 | 21.2 | 21.1 |
| Model (2.2) | 24.2 | 24.0 | 25.7 | 21.4 | 21.2 | 21.1 |
| Model (2.5) | 24.2 | 24.0 | 25.7 | 21,4 | 21.3 | 21.2 |


Figure 4: The blind inpainting results for images damaged by both impulse noise, scratch and Gaussian noise with std=10. Three sample images are showned (from top to bottom): "Barbara", "goldhill" and "cameraman". Images in each column represent (from left to right) corrupted image, restored image by (2.3) with ROLD pre-detector, restored image by (2.2) and restored image (2.5). The PSNR values of the results are given in Table 4.

### *4.3. Blind Inpainting for images damaged by multiple factors*

In this application, images are corrupted by both random-valued impulse noise and scratches without any prior knowledge on their brightness values. The values of parameters $\lambda_1, \lambda_2$ in both (2.2) and (2.5) are set as $\lambda_1 = 3.5, \lambda_2 = 5$ for all images. The value of $\lambda_3$ in (2.5) is either $1/2$ or 1, dependent on the percentage of textures in the given image.

We compared the results from our proposed models (2.2), (2.5) against that from the two-stage method (2.3) with ROLD pre-detector for detecting both random-valued impulse noise and scratches. The PSNR values of results are summarized in Table 4 and the results of some sample images are shown in Figure 4. It is seen that the model (2.5) is the best performer among all three methods, in particular, for "Barbara512" and "Goldhill" with rich textures.

The model (2.2) outperformed the model (2.3) with ROLD pre-detector on images "Goldhill" and "Cameraman" which have fewer textures; the model (2.3) with ROLD pre-detector did better on the image "Barbara512" which has rich textures. The performance of (2.3) is highly dependent on the reliability of the ROLD detector on detecting impulse noise. The ROLD detector can not detect thick scratches and also it cannot reliably detect random-valued impulse

noise mixed with Gaussian white noise. As a result, the model (2.3) did not perform well on two images with few textures. The model (2.2) did not do well on the image with rich textures is because many textures in "Barbara512" are mis-marked as scratches and been included in $v$. Thus, the texture region could be over-smoothed in (2.2). Such a weakness of (2.2) is addressed in the model (2.5) by including an explicit variable for representing textures. Thus, it is not surprising to see that the model (2.5) achieved the best performance among all three methods.

Table 4: PSNR value (dB) of the results for inpainting experiments on images degraded by mixed factors, the rate of random-valued impulse noise is set as 10%.

| Image | Standard deviation $\sigma$ of Gaussian noise | (2.3) with ROLD | (2.2) | (2.5) |
|---|---|---|---|---|
| Barbara512 | 0 | 25.2 | 24.6 | 25.2 |
| | 10 | 24.7 | 24.3 | 24.7 |
| Goldhill | 0 | 25.6 | 26.5 | 27.2 |
| | 10 | 24.4 | 26.0 | 26.4 |
| Cameraman | 0 | 23.4 | 24.8 | 24.9 |
| | 10 | 23.3 | 24.5 | 24.6 |

### *4.4. Conclusions*

This paper presents two regularization approaches for blind image inpainting problems that are capable of simultaneously identifying corrupted regions and restoring the corrupted pixels. The basic idea is to utilize the sparsity prior of images in wavelet tight frame domain (or/and in discrete local cosine transform domain) and the sparsity prior of corrupted pixels in the image domain. It is shown in the experiments that the proposed approaches did equally well as or better than the existing approaches on some image restoration problems, such as removing random-valued impulse noise from images or image deblurring in the presence of random-valued impulse noise. Moreover, the proposed approaches are the first available methods that can automatically recover images corrupted by multiple factors without requiring any user interaction. In future, we will investigate the possible applications of the proposed models to other image restoration tasks such as blind deconvolution.

## References


[1] T. Goldstein, S. Osher, The split Bregman algorithm for L1 regularized problems, SIAM Journal on Imaging Sciences 2 (2) (2009) 323–343.

[2] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, in: SIGGRAPH, 2000, pp. 417–424.

[3] M. Bertalmio, L. Vese, G. Sapiro, S. Osher, Simultaneous structure and texture image inpainting, IEEE Transactions on Image Processing 12 (8) (2003) 882–889.

[4] R. Chan, L. Shen, Z. Shen, A framelet-based approach for image inpainting, Research Report 4 (2005) 325.

[5] T. Chan, J. Shen, Variational image inpainting, Commun. Pure Appl. Math 58 (2005) 579–619.

[6] T. Chan, J. Shen, H. Zhou, Total variation wavelet inpainting, Journal of Mathematical Imaging and Vision 25 (1) (2006) 107–125.

[7] T. Chan, S. Kang, J. Shen, Euler's elastica and curvature-based inpainting, SIAM Journal on Applied Mathematics (2002) 564–592.

[8] J. Cai, R. Chan, L. Shen, Z. Shen, Convergence analysis of tight framelet approach for missing data recovery, Advances in Computational Mathematics (2008) 1–27.

[9] J. Cai, R. Chan, Z. Shen, A framelet-based image inpainting algorithm, Applied and Computational Harmonic Analysis 24 (2) (2008) 131–149.

[10] J. Cai, R. Chan, Z. Shen, Simultaneous cartoon and texture inpainting, Inverse Probl. Imaging (to appear).

[11] J. Cai, R. Chan, Z. Shen, A framelet-based image inpainting algorithm, Applied and Computational Harmonic Analysis 24 (2) (2008) 131–149.

[12] Z. Shen, Wavelet frames and image restorations, Proceedings of the International Congress of Mathematicians, Hyderabad, India.

[13] B. Dong, Z. Shen, MRA based wavelet frames and applications, IAS Lecture Notes Series, Summer Program on "The Mathematics of Image Processing", Park City Mathematics Institute.

[14] T. F. Chan, J. Shen, Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods, Society for Industrial Mathematics, 2005.

[15] R. C. Gonzalez, R. E. Woods, Digital Image Processing, 2nd Edition, Prentice Hall, 2002.

[16] H. Ji, Z. Shen, Y.-H. Xu, Robust video restoration by simultaneous sparse and low-rank matrices decomposition, Tech. rep., NUS (2010).

[17] G. Gilboa, S. Osher, Nonlocal operators with applications to image processing, Multiscale Model Sim 7 (3) (2008) 1005–1028.

[18] H. Ji, Z. Shen, Y.-H. Xu, Wavelet frame based image restoration with missing/damaged pixels, East Asia Journal on Applied Mathematics (2011) To appear.

[19] A. Ayvaci, M. Raptis, S. Soatto, Optical flow and occlusion detection with convex optimization, in: Proc. of Neuro Information Processing Systems (NIPS), 2010.

[20] R. Coifman, D. Donoho, Translation-invariant de-noising, Lecture Notes in Statistics-New York-Springer Verlag (1995) 125–125.

[21] E. Candes, D. Donoho, New tight frames of curvelets and optimal representations of objects with C2 singularities, Comm. Pure Appl. Math 56 (2004) 219–266.

[22] J. Cai, S. Osher, Z. Shen, Split Bregman methods and frame based image restoration, Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal 8 (2) (2009) 337–369.

[23] C. Chaux, P. Combettes, J. Pesquet, V. Wajs, A variational formulation for frame-based inverse problems, Inverse Problems 23 (2007) 1495–1518.

[24] I. Daubechies, G. Teschke, L. Vese, Iteratively solving linear inverse problems under general convex constraints, Inverse Problems and Imaging 1 (1) (2007) 29.

[25] M. Elad, J. Starck, P. Querre, D. Donoho, Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA), Applied and Computational Harmonic Analysis 19 (3) (2005) 340–358.

[26] M. Fadili, J. Starck, F. Murtagh, Inpainting and zooming using sparse representations, The Computer Journal 52 (1) (2009) 64.

[27] A. Chai, Z. Shen, Deconvolution: A wavelet frame approach, Numerische Mathematik 106 (4) (2007) 529–587.

[28] T. Goldstein, X. Bresson, S. Osher, Geometric applications of the split bregman method: Segmentation and surface reconstruction, Journal of Scientific Computing 45 (1) (2009) 272–293.

[29] J. Cai, S. Osher, Z. Shen, Linearized Bregman iterations for compressed sensing, Math. Comp 78 (2009) 1515–1536.

[30] I. Daubechies, Ten lectures on wavelets, Society for Industrial Mathematics, 1992.

[31] A. Ron, Z. Shen, Affine Systems in $L_2(\mathbb{R}^d)$: The Analysis of the Analysis Operator, Journal of Functional Analysis 148 (2) (1997) 408–447.

[32] I. Daubechies, B. Han, A. Ron, Z. Shen, Framelets: MRA-based constructions of wavelet frames, Applied and Computational Harmonic Analysis 14 (1) (2003) 1–46.

[33] J. Cai, M. Nikolova, R. Chan, Fast two-phase image deblurring under impulse noise, J. Math. Imaging Vis. 36 (2010) 46–53.

[34] L. Bar, N. Kiryati, N. Sochen, Image deblurring in the presence of impulse noise, International Journal of Computer Vision 70 (3) (2006) 279–298.

[35] L. Bar, N. Sochen, N. Kiryati, Image deblurring in the presence of salt-and-pepper noise, in: Lecture Notes in Computer Science, Scale Space and PDE Methods in Computer Vision, LNCS), Vol. 3459, 2005, pp. 107–118.

[36] G. Healey, R. Kondepudy, Radiometric ccd camera calibration and noise estimation, IEEE Trans. PAMI 16 (3) (1994) 267–276.

[37] S. Alliney, Digital filters as absolute norm regularizers, IEEE Transactions on Signal Processing 40 (6) (1992) 1548–1562.

[38] T. Chan, S. Esedoglu, Aspects of Total Variation Regularized $L_1$ Function Approximation, SIAM Journal on Applied Mathematics 65 (5) (2005) 1817.

[39] M. Nikolova, A variational approach to remove outliers and impulse noise, Journal of Mathematical Imaging and Vision 20 (1) (2004) 99–120.

[40] X. Zhang, M. Burger, X. Bresson, S. Osher, Bregmanized nonlocal regularization for deconvolution and sparse reconstruction, SIAM Journal on Imaging Sciences 3 (1) (2010) 253–276.

[41] E. Esser, Applications of Lagrangian-based alternating direction methods and connections to split Bregman, Tech. Rep. TR 09-31, UCLA (March 2009).

[42] X. Tai, C. Wu, Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model, Scale Space and Variational Methods in Computer Vision (2009) 502–513.

[43] R. Glowinski, P. Le Tallec, Augmented Lagrangian and operator-splitting methods in nonlinear mechanics, Society for Industrial Mathematics, 1989.

[44] S. Setzer, Split Bregman algorithm, Douglas-Rachford splitting and frame shrinkage, Scale Space and Variational Methods in Computer Vision (2009) 464–476.

[45] D. Donoho, De-noising by soft-thresholding, IEEE transactions on information theory 41 (3) (1995) 613–627.

[46] P. Combettes, V. Wajs, Signal recovery by proximal forward-backward splitting, Multiscale Modeling and Simulation 4 (4) (2006) 1168–1200.

[47] B. Dong, E. Savitsky, S. Osher, A Novel Method for Enhanced Needle Localization Using Ultrasound-Guidance, Advances in Visual Computing (2009) 914–923.

[48] T. Chen, H. R. Wu, Adaptive impulse dectection using center-weighted median filters, IEEE Signal Processing Letters 8 (1) (2001) 1–3.

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

[50] Y. Dong, R. F. Chan, S. Xu, A detection statistic for random-valued impulse noise, IEEE Transactions on Image Processing 16 (4) (2007) 1112–1120.