Image Restoration with Mixed or Unknown Noises

Zheng Gong[*], Zuowei Shen[†], and Kim-Chuan Toh[‡]

Abstract

This paper proposes a simple model for image restoration with mixed or unknown noises. It can handle image restoration without assuming any prior knowledge of the noise distribution. It is particularly useful for solving real life image restoration problems, since under various constraints, images are always degraded with mixed noise and it is impossible to determine what type of noise is involved. The proposed model can remove mixed type of noises as well as unknown type of noises, and at the same time also works comparably well against the model whose data fitting term is designed for a specific given noise type. While most of the existing methods for image restorations are designed specifically for a given type of noise, our model appears to be the first universal one for handling image restoration with various mixed noises and unknown type of noises. Extensive simulations on synthetic data show that our method is effective and robust in restoring images contaminated by additive Gaussian noise, Poisson noise, random-valued impulse noise, multiplicative Gamma noise and mixtures of these noises. Numerical results on real data show that it can remove noises without any prior knowledge of the noise distribution.

1 Introduction

Image restoration is often formulated as an inverse problem, which amounts to recovering an unknown true image u from a degraded image (or a measurement) b given by

$$b \approx Hu, \tag{1.1}$$

where H is a linear operator denoting the degradation operations and Hu is also possibly contaminated by random noises. Such noises can be the additive noise (e.g. Gaussian noise, impulse noise, in which case $b = Hu + \varepsilon$), or Poisson noise and other multiplicative noise. A typical H can be the identity operator, a convolution or a

[*]Department of Mathematics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 117576. Email: gongzheng@nus.edu.sg.

[†]Department of Mathematics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076. Email: matzuows@nus.edu.sg.

[‡]Department of Mathematics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076; and Singapore-MIT Alliance, 4 Engineering Drive 3, Singapore 117576. Email: mattohkc@nus.edu.sg.

projection, and the corresponding inverse problems are known as denoising, deblurring and inpainting.

To restore u, one of the most popular approach is to solve a minimization problem of the following form

$$\min_u R_1(u) + R_2(u) \tag{1.2}$$

where $R_1(u)$ denotes certain data fitting term derived according to the assumed noise type and $R_2(u)$ is a regularization term that imposes the prior on u. The latter is necessary due to the ill-posedness of the inverse problem.

Typically, the regularization term $R_2(u)$ is determined by the prior assumptions imposed on the underlying solutions. One of the assumptions commonly used is the sparsity of the underlying solutions in some transformed domain. Such transforms can be gradient (e.g. the total variation (TV) model), wavelet tight frames, local cosine transforms etc. Since the TV model is closely related to the wavelet tight frame model (see [6]) and since wavelet tight frame model has proven to be efficient in image restorations (see e.g. [32, 13]), in this paper, we use the sparsity in the wavelet tight frame transform domain as the prior assumption on the underlying solutions. For this, we will take $R_2(u) = \|Wu\|_1$, where W is the wavelet tight frame transform.

Generally, the choice of the data fitting term $R_1(u)$ depends on the specified noise distribution. For example, the ℓ_2-norm fitting function

$$R_1(u) = \|Hu - b\|_2^2 \tag{1.3}$$

is used for additive Gaussian noise, which averages out the noise. When the image is corrupted by impulse noise, a typical choice of $R_1(u)$ is the ℓ_1-norm fitting function

$$R_1(u) = \|Hu - b\|_1, \tag{1.4}$$

which leads to the median of the data. Note that impulse noise can be considered as outliers, and a median approximation is a good choice since it is robust to outliers. For Poisson noise and multiplicative Gamma noise, the Bayesian maximum a posterior (MAP) likelihood estimation approach gives rise to the Csiszár's I-divergence of Hu from b [20, 31, 35] as the data fitting term, i.e.,

$$R_1(u) = \sum_i^n \left((Hu)_i - b_i \log(Hu)_i\right). \tag{1.5}$$

However, since the noise in an image seldom appears from a single distribution, nor could one determine the noise type in reality, a specific data fitting term for a given noise may not work for mixed or unknown noises. This paper aims to find a simple model which can remove mixed type of noises as well as unknown type of noises, while at the same time also works comparably well against the model whose data fitting term is designed for a specific given noise type.

The data fitting term $R_1(u)$ suggested here is surprisingly simple. It is the weighted sum of ℓ_1-norm and ℓ_2-norm fitting functions, which leads to the following model:

$$\min_{u\in\mathbb{R}^n} \lambda_1\|Hu - b\|_1 + \frac{\lambda_2}{2}\|Hu - b\|_2^2 + \rho\|Wu\|_1, \tag{1.6}$$

where λ_1, λ_2 and ρ are nonnegative parameters. Note that the parameter λ_1 in (1.6) is always fixed as 1 in our numerical simulations, but we keep it in (1.6) for ease of discussion when comparing to other models. While the model (1.6) looks too simple, our numerical simulation results show that this model, together with the proposed numerical algorithm, can efficiently remove various mixed noises and unknown noises. When it is applied to remove a given type of noise, such as additive Gaussian noise, impulse noise, Poisson noise and multiplicative Gamma noise, it performs as well as those models whose data fitting terms are designed according to the statistical distribution of the noise. We should mention that our proposal to use the data fitting term in (1.6) is primarily based on empirical experiments although it appears to be a natural choice given that numerous papers have considered ℓ_1, ℓ_2, and I-divergence fitting terms separately for various image processing tasks. In fact, we have extensively experimented with the weighted sum of the three fitting terms just mentioned, but find that a suitable weighted sum of ℓ_1 and ℓ_2 fitting terms are almost always as good as the weighted sum of three terms.

As mentioned, our primary goal is to find a simple model that is applicable for various type of noises removal in image restorations, and consequently can be adapted to different applications. Most existing noise removal models have the data fitting terms designed according to the probability distributions of the noises. For example, the ℓ_2-norm fitting term is used in removing Gaussian noise, and the ℓ_1-norm fitting term has been used for impulse noise removal [4, 3], as well as mixed Gaussian and impulse noise removal [19]. Although the usefulness of ℓ_1 fidelity has been demonstrated for specific noise removal tasks in the literature, we believe our paper is the first to demonstrate its suitability for universal noise removal when combined with an ℓ_2 fidelity. Our first main contribution in this paper is in discovering that when a weighted sum of ℓ_1 and ℓ_2 fidelities is used as the data fitting term, the model works effectively and robustly for the removal of mixed noise, or almost any type of unknown noise in general. This means that the model can be used without prior knowledge of the noise. Thus, it can be applied to restore images from real life problems, where the noise distribution is typically mixed or unknown. Indeed, we have demonstrated the versatility of our proposed model in denosing real life color images in the numerical experiments. Our second contribution is in designing an efficient augmented Lagrangian method (ALM) for solving a constrained reformulation of the proposed model and its main feature is that at each iteration, we are able to construct a smooth objective function for the subproblem by substituting out the non-smooth ℓ_1 terms. This allows us to use a fast gradient method, known as an accelerate proximal gradient (APG) method [5], to solve a smooth unconstrained inner subproblem at each ALM iteration. The resulting ALM-APG algorithm is demonstrated to be superior in solving the proposed model when compared to the commonly employed alternating direction method of multipliers [16, 15] or split Bregman method [17, 8] for solving image restoration problems with ℓ_1 regularization terms.

The rest of the paper is organized as follows. In Section 2, we provide the numerical algorithm for the model and its convergence analysis. The numerical results for various image restorations with mixed noise and examples on noise removal of real digital colour images are presented in Section 3. While there is no result for image restorations with

such a wide range of mixed noise available in the literature as far as we are aware of, comparisons with some of the available models for removing noises such as single type of noise, mixed Poisson-Gaussian noise, and impulse noise mixed with Gaussian noise are given in Section 4. Some additional remarks on our proposed model and numerical algorithm will be addressed in Section 5.

2 The ALM-APG Algorithm

In this section, we introduce an augmented Lagrangian method (ALM) with the inner subproblem being solved by an accelerated proximal gradient (APG) algorithm to solve (1.6). The convergence of the ALM for general convex programming was established in [28], based on which, we provide the convergence analysis for our algorithm.

For this, we shall consider the following more general problem:

$$\min_{x\in\mathbb{R}^n} \; F(x) + \beta^T|Ax - c| \tag{2.1}$$

where $F(x) : \mathbb{R}^n \to \mathbb{R}$ is assumed to be continuously differentiable with Lipschitz continuous gradient and convex, but not necessarily strictly convex, $A \in \mathbb{R}^{m\times n}$ is a given matrix, and $c \in \mathbb{R}^m$, $\beta \in \mathbb{R}^m_{++}$ are given vectors. Here, $|x|$ denotes the vector obtained from x by taking absolute values of its components. We assume that (2.1) has a finite infimum.

To link (2.1) to the model (1.6), we define

$$A = \begin{pmatrix} H \\ W \end{pmatrix}, \quad c = \begin{pmatrix} b \\ 0 \end{pmatrix}, \quad \text{and} \quad \beta = \begin{pmatrix} \lambda_1 e \\ \rho\hat{e} \end{pmatrix},$$

where e and $\hat{e}$ are vectors of ones. Then, (2.1) corresponds to (1.6) when we set $F(u) = \frac{\lambda_2}{2}\|Hu - b\|^2$.

To solve (2.1), we reformulate it to the following equivalent problem by introducing a new variable z:

$$\min_{x\in\mathbb{R}^n, z\in\mathbb{R}^m} \Big\{ f(x,z) := F(x) + \beta^T|z| \;\; | \;\; Ax + z = c \Big\}. \tag{P}$$

Now we are ready to apply the augmented Lagrangian method (ALM) to solve (P).

2.1 The Augmented Lagrangian Method

The augmented Lagrangian method (ALM), also known as the method of multipliers is widely used for solving convex programming problems. Before stating the ALM, let us define the augmented Lagrangian function of the primal problem (P) associated with a given parameter $\sigma > 0$:

$$\mathcal{L}_\sigma(x, z; y) \;\; = \;\; f(x,z) + \langle y, c - Ax - z\rangle + \frac{\sigma}{2}\|c - Ax - z\|^2. \tag{2.2}$$

Then the ALM for solving (P) is summarized as follows:

Augmented Lagrangian Method
Given a tolerance $\varepsilon > 0$. Input y^0 and $\sigma_0 > 0$. For $k = 0, 1, 2, ...$, iterate:

Step 1. Find an approximate minimizer

$$w^{k+1} \approx \arg\min_w \{\psi_k(w) := \mathcal{L}_{\sigma_k}(w; y^k)\}. \tag{2.3}$$

Step 2. Compute $y^{k+1} = y^k + \sigma_k(c - \tilde{A}w^{k+1})$.

Step 3. If $\|(y^k - y^{k+1})/\sigma_k\| \leq \varepsilon$, stop; else update σ_k such that

$$0 < \sigma_k \uparrow \sigma_\infty \leq +\infty. \tag{2.4}$$

Here we define $w = (x^T,\, z^T)^T$, $\tilde{A} = (A,\, I_m)$, and $w^{k+1} \approx \arg\min_w \psi_k(w)$ means that $\psi_k(w^{k+1}) \approx \min_w \psi_k(w)$.

2.1.1 Convergence

In the following, we provide the convergence result of the ALM applied to (P) by adapting Theorem 4 in [28], which was proved via the well-developed theory on proximal point algorithm for maximum monotone operators [29]. Essentially, applying the ALM to solve (P) is equivalent to applying the proximal point algorithm to solve the corresponding dual problem:

$$\max_{y \in \mathbb{R}^m} \quad g(y) := \inf_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} \ell(x, z; y), \tag{2.5}$$

where ℓ is the ordinary Lagrangian function of (P), i.e.

$$\ell(x, z; y) = f(x, z) + \langle y, c - Ax - z \rangle. \tag{2.6}$$

More explicitly, the dual problem (2.5) can be written as:

$$\max_{y \in \mathbb{R}^m} \Big\{ \langle c, y \rangle - F^*(A^T y) \;\; | \;\; |y_i| \leq \beta_i,\, i = 1, 2, ..., m \Big\} \tag{D}$$

where $F^*(\cdot)$ denotes the convex conjugate function of F. Observe that the feasible set of (D) is bounded.

Now we show that the ALM applied to (P) is the proximal point algorithm applied to the dual problem (D). Let G_σ be the Moreau-Yosida regularization of g associated with σ, i.e.

$$\begin{aligned}
G_\sigma(y) &= \max_{s \in \mathbb{R}^m} \{g(s) - \frac{1}{2\sigma}\|s - y\|^2\} \\
&= \max_{s \in \mathbb{R}^m} \inf_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} \{f(x, z) + \langle s, c - Ax - z \rangle - \frac{1}{2\sigma}\|s - y\|^2\} \\
&= \inf_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} \Big\{ f(x, z) + \max_{s \in \mathbb{R}^m} \{\langle s, c - Ax - z \rangle - \frac{1}{2\sigma}\|s - y\|^2\} \Big\} \\
&= \inf_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} \{f(x, z) + \langle y, c - Ax - z \rangle + \frac{\sigma}{2}\|c - Ax - z\|^2\} \\
&= \inf_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} \mathcal{L}_\sigma(x, z; y),
\end{aligned}$$

where the interchange of $\max_s$ and $\inf_{x,z}$ in the second equality follows from Theorem 37.3 in [27], since there is no direction of recession for the objective function of the minimax problem in the variable s. It is easy to show that the optimal solution to the resulted inner maximization problem is $s^* = y + \sigma(c - Ax - z)$, and by substituting in s^*, we thus obtain the augmented Lagrangian function.

To establish the convergence result of our algorithm via a direct application of Theorem 4 in [28], one requires $\sup(\mathrm{D}) > -\infty$, which guarantees that the operator associated with the dual objective function g defined by

$$\mathcal{T}_g(y) = \{u \in \mathbb{R}^m : u \in -\partial g(y)\},\ y \in \mathbb{R}^m, \tag{2.7}$$

is maximum monotone, and hence the solutions to the inclusion problem $0 \in \mathcal{T}_g(y)$ are the optimal solutions to (D).

Theorem 2.1. *Let $\{\epsilon_k\}$ be a given summable sequence of positive numbers. Suppose the augmented Lagrangian method is executed with the following stopping criterion*

$$\psi_k(w^{k+1}) - \inf \psi_k \leq \frac{\epsilon_k^2}{2\sigma_k} \tag{2.8}$$

in (2.3). Then the generated sequence $\{y^k\} \subset \mathbb{R}^m$ is bounded and $y^k \to y^*$, where y^* is some optimal solution to (D). Moreover, the sequence $\{w^k\} = \{(x^k; z^k)\}$ is asymptotically minimizing for (P) with

$$\|c - Ax^{k+1} - z^{k+1}\| = \sigma_k^{-1}\|y^{k+1} - y^k\| \to 0 \tag{2.9}$$

$$f(x^{k+1}, z^{k+1}) - \inf(P) \leq (\epsilon_k^2 + \|y^k\|^2 - \|y^{k+1}\|^2)/(2\sigma_k). \tag{2.10}$$

If there exists an α such that the set of all feasible (x, z) in (P) satisfying $f(x, z) \leq \alpha$ is nonempty and bounded, then $\{(x^k, z^k)\}$ is also bounded, and any of its cluster points is an optimal solution to (P).

Proof. Since the primal problem (P) has only equality constrains, which guarantees the Slater's condition, there exists an optimal solution to (D) and $\max(D) = \inf(P) = \operatorname{asym}\inf(P)$. By Theorem 4 in [28], $\{y^k\}$ is bounded and converges to y^*, which is some optimal solution to (D); and $\{(x^k, z^k)\}$ is asymptotically minimizing for (P), as given in (2.9) and (2.10).

Moreover, since $\{y^k\}$ is bounded, the right-hand-side of (2.10) is bounded, and therefore the final statement follows from Corollary 8.7.1 in [27] that for a closed proper convex function, if there exist a nonempty and bounded level set, then all of its level sets are bounded. □

Remark 2.2. For image denoising problems, the corresponding objective function of (2.1) is coercive. Thus, the solution set of (2.1) is nonempty and bounded and so is the solution set of (P). In that case, $\{(x^k; z^k)\}$ is bounded, and any of its cluster points is an optimal solution to (P).

2.2 An APG Algorithm for solving the inner subproblem

Note that the main task in each of the ALM iteration is to solve the minimization problem in Step 1. Here we propose an accelerated proximal gradient (APG) algorithm to solve this inner subproblem.

To introduce the APG algorithm, we consider the following unconstrained convex minimization problem

$$\min_{x\in\mathbb{R}^n} h(x), \tag{2.11}$$

where $h(x) \in C^{1,1}(\mathbb{R}^n)$, i.e. there exists a constant $L > 0$ such that $\forall\ x, \bar{x} \in \mathbb{R}^n$

$$\|\nabla h(x) - \nabla h(\bar{x})\| \le L\|x - \bar{x}\|. \tag{2.12}$$

Fix any $\bar{x} \in \mathbb{R}^n$, consider the linear approximation of $h(x)$ at $\bar{x}$:

$$\ell_h(x;\bar{x}) := h(\bar{x}) + \langle \nabla h(\bar{x}), x - \bar{x}\rangle.$$

It follows from (2.12) and the convexity of h that

$$h(x) - \frac{L}{2}\|x - \bar{x}\|^2 \le \ell_h(x;\bar{x}) \le h(x).$$

The main step of the APG algorithm for solving (2.11) is via the following subproblem

$$\min_x\ \ell_h(x;\bar{x}) + \frac{L}{2}\|x - \bar{x}\|^2,$$

which is a strictly convex problem, and whose unique solution is given by

$$x = \bar{x} - L^{-1}\nabla h(\bar{x}). \tag{2.13}$$

The APG algorithm for solving (2.11) can be summarized as follows:

APG Algorithm
Choose $x_0 = x_1 \in \mathbb{R}^n$ and $t_0 = t_1 = 1$. For $i = 1, 2, ...$, iterate:

Step 1. Set $\bar{x}_i = x_i + \frac{t_{i-1}-1}{t_i}(x_i - x_{i-1})$;

Step 2. Compute $x_{i+1} = \bar{x}_i - L^{-1}\nabla h(\bar{x}_i)$;

Step 3. Set $t_{i+1} = \frac{1+\sqrt{1+4(t_i)^2}}{2}$.

Remark 2.3. When $t_i = 1$ for all i, the above APG algorithm is reduced to the standard gradient descent method with step length $1/L$. However, the APG algorithm described here can achieve ε-optimality (in terms of function value) in $O(\sqrt{L/\varepsilon})$ iterations [5] instead of the complexity of $O(L/\varepsilon)$ iterations for the standard gradient descent method. Note that there are many other variants of $O(\sqrt{L/\varepsilon})$ methods for nonsmooth convex optimization, such as those studied by Nesterov, Nemirovski [24, 23, 25]. Some of those may also be applicable for the situation here, but it is not our main focus to compare various first-order methods for solving (2.11).

The application of the APG algorithm to solve the inner subproblem, i.e. for a given $y^k \in \mathbb{R}^m$ and $\sigma_k > 0$, one seeks to find

$$(x^{k+1}, z^{k+1}) \approx \arg\min_{x,z} \mathcal{L}_{\sigma_k}(x, z; y^k), \tag{2.14}$$

is based on the observation that the variable z can be solved explicitly in terms of x. By reformulating the augmented Lagrangian function $\mathcal{L}_{\sigma_k}$, we have

$$\begin{aligned} \mathcal{L}_{\sigma_k}(x, z; y^k) &= f(x, z) + \frac{\sigma_k}{2}\|c - Ax - z + \frac{1}{\sigma_k}y^k\|^2 - \frac{1}{2\sigma_k}\|y^k\|^2 \\ &= F(x) + \beta^T|z| + \frac{\sigma_k}{2}\|c - Ax - z + \frac{1}{\sigma_k}y^k\|^2 - \frac{1}{2\sigma_k}\|y^k\|^2. \end{aligned}$$

First we consider the minimization in items of z:

$$\min_{z\in\mathbb{R}^m} \quad \beta^T|z| + \frac{\sigma_k}{2}\|c - Ax - z + \frac{1}{\sigma_k}y^k\|^2 = \frac{1}{\sigma_k}\sum_{i=1}^{m}\phi_{\beta_i}(\eta_i), \tag{2.15}$$

where

$$\eta = \sigma_k(c - Ax) + y^k,$$

and $\phi_\varepsilon(t)$ is the Huber function defined by

$$\phi_\varepsilon(t) = \begin{cases} \frac{1}{2}t^2 & |t| \le \varepsilon; \\ \varepsilon|t| - \frac{1}{2}\varepsilon^2 & |t| > \varepsilon. \end{cases} \tag{2.16}$$

The optimal z in (2.15) in terms of x is given by:

$$z = \frac{1}{\sigma_k}S_\beta(\eta). \tag{2.17}$$

Here, for a given nonnegative vector $\nu \in \mathbb{R}^n$, $S_\nu : \mathbb{R}^n \to \mathbb{R}^n$ is the soft-threshholding map defined by

$$S_\nu(x) = \mathrm{sgn}(x) \circ \max\{|x| - \nu, 0\}, \tag{2.18}$$

where "$\circ$" denotes the component-wise product of two vectors, i.e. $(x \circ y)_i = x_i y_i$, and $\mathrm{sgn}(\cdot)$ is the sign function such that $\mathrm{sgn}(t)$ is the sign of t if $t \neq 0$, and $\mathrm{sgn}(0) = 0$.

Now we are ready to apply the APG algorithm to compute the optimal x from the following problem:

$$\min_{x\in\mathbb{R}^n} \; h(x) := \frac{1}{\sigma_k}\sum_{i=1}^{m}\phi_{\beta_i}(\eta_i) + F(x), \tag{2.19}$$

where the gradient of h is given by

$$\nabla h(x) = -A^T\left(\mathrm{sgn}(\eta) \circ \min\{|\eta|, \beta\}\right) + \nabla F(x). \tag{2.20}$$

Note that when $F(x)$ is strictly convex or A has full column rank, $h(x)$ is a strictly convex function and hence, the above minimization has a unique solution.

Remark 2.4. The reason that $x^* := \arg\min_x h(x)$ and $z^* := S_\beta(\eta^*)/\sigma_k$, where $\eta^* = \sigma_k(c - Ax^*) + y^k$, are indeed the optimal solution to $\min_{x,z} \mathcal{L}_{\sigma_k}(x, z; y^k)$ is the following:

$$\begin{aligned}&\mathcal{L}_{\sigma_k}(x^*, z^*; y^k) = h(x^*) - \frac{1}{2\sigma_k}\|y^k\|^2 \le h(x) - \frac{1}{2\sigma_k}\|y^k\|^2 \\ &= F(x) + \min_z\{\beta^T|z| + \frac{\sigma_k}{2}\|c - Ax - z + \frac{1}{\sigma_k}y^k\|^2\} - \frac{1}{2\sigma_k}\|y^k\|^2 \le \mathcal{L}_{\sigma_k}(x, z; y^k)\,\forall\, x, z.\end{aligned}$$

Remark 2.5. Since

$$\partial\nabla h(x) = \left\{ \sigma_k A^T \mathrm{diag}(\alpha) A + \nabla^2 F(x) \mid \alpha_i \in \left\{ \begin{array}{ll} \{0\} & \text{if } |\eta_i| > \beta_i \\ \{1\} & \text{if } |\eta_i| < \beta_i \\ [0,1] & \text{if } |\eta_i| = \beta_i \end{array} \right\}, \right.$$

we have the following estimation of the Lipschitz constant L of $\nabla h(\cdot)$:

$$L \le \sigma_k \lambda_{\max}(A^T A) + \lambda_{\max}(\nabla^2 F(x)),$$

where $\lambda_{\max}(A^T A)$ denotes the largest eigenvalue of $A^T A$.

As a summary, the ALM-APG algorithm we use to solve (P) is as follows:

ALM-APG Algorithm
Given a tolerance $\varepsilon > 0$. Input x^0, y^0 and $\sigma_0 > 0$. For $k = 0, 1, 2, ...$, iterate:

Step 1. Let $x_0 = x_1 = x^k$, and $t_0 = t_1 = 1$.

For $i = 1 : p$, iterate the following steps:

$$\left\{ \begin{array}{l} \bar{x}_i = x_i + \dfrac{t_{i-1} - 1}{t_i}(x_i - x_{i-1}) \\ \\ x_{i+1} = \bar{x}_i - L^{-1}\nabla h(\bar{x}_i), \quad t_{i+1} = \dfrac{1 + \sqrt{1 + 4(t_i)^2}}{2}. \end{array} \right.$$

Set $x^{k+1} = x_{p+1}$, $\quad z^{k+1} = \frac{1}{\sigma_k} S_\beta\big(\sigma_k(c - Ax^{k+1}) + y^k\big)$.

Step 2. Compute $y^{k+1} = y^k + \sigma_k(c - Ax^{k+1} - z^{k+1})$.

Step 3. If $\|(y^k - y^{k+1})/\sigma_k\| \le \varepsilon$, stop; else update σ_k such that $0 < \sigma_k \uparrow \sigma_\infty \le +\infty$.

3 Numerical Simulation

This section is mainly devoted to numerical simulation of image restorations in the presence of mixed noise. We show the efficiency, effectiveness and robustness of the model (1.6) by applying the ALM-APG algorithm to restore images with a wide range of mixed noise. Specifically, we consider image restorations with the mixture of Gaussian, Poisson, and impulse noises. To the best of our knowledge, there is no discussion of image restorations with such a wide range of mixed noises in the current literature. It should be noted that video denoising for the mixture of these three noises has been

considered in [18]. However, the method given there is not applicable for the image case, since it relies heavily on the temporal direction information.

In addition, some denoising examples of real life digital color images are also presented to show that the proposed method is capable of removing unknown noises.

The performance on restorations of synthetic data is measured by the PSNR value defined as

$$\text{PSNR} = 10 \log_{10} \left(\frac{I_{\max}^2}{\text{MSE}} \right), \tag{3.1}$$

where $I_{\max}$ is the maximum intensity of the original image. In the presence of Poisson noise, the maximum intensity of the original noise free image is varied in order to create images with different levels of Poisson noise.

We omit giving any details on wavelet tight frames and how they can be used for image restorations, since there is a rich literature on this topic. The details can be found in the survey paper [32] and the long note [13] and the references therein. The wavelet tight frame transform W used in this paper is generated from piecewise linear B-spline constructed via the unitary extension principle [30]. The wavelet transform W and its inverse related to the wavelet frame system via the wavelet tight frame decomposition and reconstruction operators can be found in [11].

3.1 Image Denoising

We start with the simpler case of the mixed noise removal. The more difficult case of deblurring in the presence of mixed noise will be given in the next subsection. The key point for the noise removal here is that we do not need to know a priori what kind of noises contaminate the image. The aim is to remove noises in real images, and examples will be given later.

3.1.1 Synthetic Images

The mixed noise considered in this section includes additive Gaussian noise, Poisson noise and impulse noise. The Poisson noise in this experiment was generated using the "`poissrnd`" function in MATLAB with the input image scaled to the maximum intensity ($\text{I}_{\max}$) as specified in each experiment. For the impulse noise, we only consider the random-valued impulse noise, because a pixel contaminated by such an impulse noise is not as distinctively an outlier as that contaminated by the salt-and-pepper noise, and consequently is more difficult to detect. The random-valued impulse noise is defined as follows: with probability r, the pixel value u_{ij} is altered to be a uniform random number in the interval between the minimum and maximum intensity of the image. For all cases, impulse noise was the last to be added.

First, we consider the case when images' maximum intensity is not rescaled. The results in terms of PSNR using model (1.6) are summarized in Table 1 and Table 2, and some of the restored images are presented in Figure 1 and Figure 2. The results show that the model together with our proposed ALM-APG algorithm are effective in removing random-valued impulse noise mixed with Poisson noise and Gaussian noise. In the case when all three types of noises are involved, we consider both cases of generating Poisson noise before and after adding Gaussian noise. It turns out that

Random-valued impulse noise (r)	10%		20%		40%	
Gaussian noise ($\hat{\sigma}$)	0	10	0	10	0	10
Poisson+Gaussian	29.07	28.31	27.10	26.60	23.46	23.68
Gaussian+Poisson		28.38		26.57		23.35

Table 1: Denoising results (PSNR) for the image "Cameraman", in the presence of random-valued impulse noise, Gaussian noise and Poisson noise at image peak intensity of 255.

Image	Baboon		Boat		Bridge		Barbara512	
Random-valued impulse noise (r)	10%	20%	10%	20%	10%	20%	10%	20%
Poisson+Gaussian	25.26	24.08	27.60	26.17	26.38	25.23	27.20	25.58
Gaussian+Poisson	25.23	24.13	27.57	26.19	26.26	25.06	27.24	25.54

Table 2: Denoising results (PSNR) for various testing images, in the presence of random-valued impulse noise, Gaussian noise with standard deviation $\hat{\sigma} = 10$ and Poisson noise at image peak intensity of 255.

the performance of our method is robust regardless of the intrinsic distribution of the image noise.

Second, we conduct a more extensive test of our method in removing mixed noise of the three types as previously discussed, where the Poisson noise is generated from noise-free images rescaled to the maximum intensity ranging from 120 to 1. In the cases when both Gaussian noise and Poisson noise are involved, Gaussian noise is added after Poisson noise with standard deviation $\hat{\sigma} = \mathrm{I}_{\max}/10$. The results in terms of PSNR are summarized in Table 3. The results show that the simple model (1.6) together with the proposed ALM-APG algorithm is effective in removing mixed impulse noise, Poisson noise and Gaussian noise at different levels.

The choice of the parameters were $\lambda_1 = 1$, $\lambda_2 = 0.01 \sim 0.1$ and $\rho = 1 \sim 2$. The value of ρ depends on the general noisiness of the image: the noisier the image is, the larger ρ is chosen; however, the value of λ_2 mainly depends on the impulse noise level, where smaller value is preferred when the impulse noise level gets higher. Such a choice of λ_2 is reasonable since a distinctive outlier will make a significant contribution to the ℓ_2 fitting term in (1.6), and hence a smaller λ_2 should be chosen.

3.1.2 Real Images

In digital color photos, there are two most significant sources of noise: the photon shot noise due to the random striking of the photons on the image sensor, and the leakage current due to the additional electrical signal generated by the semiconductor when converting energy from photons to electrical energy. Besides, interpolation of the captured partial color data to complete the RGB channels, quantization and artifacts caused by JPEG format and the build-in sharpening, denoising functions in cameras etc, make it difficult or rather impossible to model the noise. Therefore, real image denoising problems are much more challenging than those of synthetic data.

The main difficulty behind the noise removal for real images is that there is no

(a) $r = 20\%$, Poisson PSNR=14.40 (b) $r = 20\%$, $\hat{\sigma} = 10$ $P + G$, PSNR=13.86 (c) $r = 20\%$, $\hat{\sigma} = 10$ $G + P$, PSNR=14.14

(d) denoised: PSNR=27.10 (e) denoised: PSNR=26.60 (f) denoised: PSNR=26.57

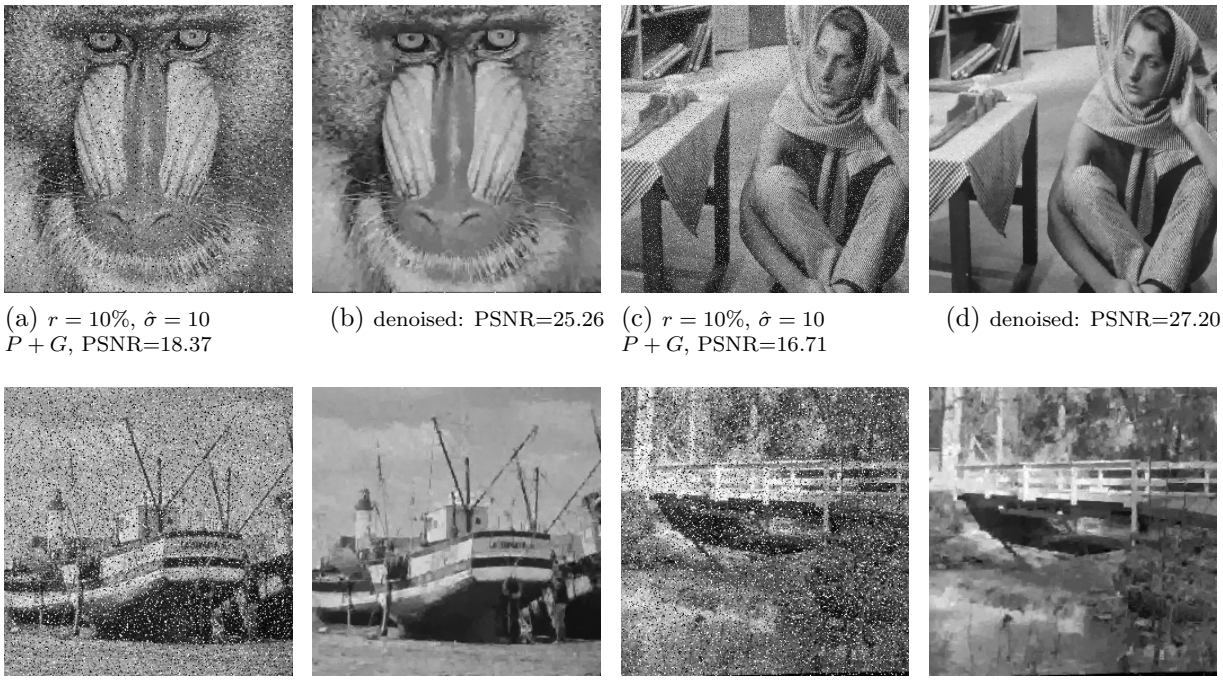
Figure 1: Denoising results for the image "Cameraman", in the presence of random-valued impulse noise, Poisson noise and Gaussian noise.

prior knowledge of the noise and its statistical distribution, which itself is the result of a mixture of different noises. Hence, models based on a specific type of noise distribution is hard to be effective. Since our model does not assume any prior statistical distribution of the noise, it has the potential to perform well in real image denoising. Here, we show the promise of the method via a few examples.

Most digital color images are in the RGB color space. It is known that due to the uneven distribution of the noise in each channel, by denoising each channel separately, one tends to excessively denoise the blue channel, which can lead to undesirable color artifacts. A standard practice is to transform the RGB color space to YCrCb color space (linear transformation) or LAB color space (nonlinear transformation), both of which separate the luminance and chrominance. However, the luminance resulted from both transformations is still contaminated by the noise from the blue channel, and if a substantial denoising process is performed on the luminance channel, the quality of the denoised image can be adversely affected. Chan *et al.* proposed a modified YCrCb (m-YCrCb) color space [9], which is more effective since the luminance channel does not contain any information from the blue channel. More precisely, the m-YCrCb color space is obtained from the RGB color space via following linear transformation:

$$\mathrm{Y} = 0.666G + 0.334R,\ \mathrm{Cr} = \frac{0.666}{1.6}(R - G),\ \mathrm{Cb} = \frac{1}{2}(B - 0.666G - 0.334R). \quad (3.2)$$

In [9], the authors use the multiscale total variational (MTV) method for denoising.



(a) $r = 10\%$, $\hat{\sigma} = 10$ $P + G$, PSNR=18.37 (b) denoised: PSNR=25.26 (c) $r = 10\%$, $\hat{\sigma} = 10$ $P + G$, PSNR=16.71 (d) denoised: PSNR=27.20

(e) $r = 20\%$, $\hat{\sigma} = 10$ $P + G$, PSNR=14.87 (f) denoised: PSNR=26.17 (g) $r = 20\%$, $\hat{\sigma} = 10$ $P + G$, PSNR=13.97 (h) denoised: PSNR=25.23

Figure 2: Denoising results for various testing images, in the presence of random-valued impulse noise, Poisson noise and Gaussian noise.

Here, we adopted their m-YCrCb transformation (3.2) and apply our model (1.6) to each of the transformed channel, as numerical simulation showed that its performance is superior to a direct application of our method to each of the RGB channel. We present both of the denoised result in [9] and ours in Figure 3.

One can observe that while there is still noise visible in the denoised image using MTV (3b), our model provides a clearer result (3c) with equally sharpe edges and details. The improvement in blue channel is not that significant, but one can still notice the difference in the shadow part.

Besides the example used in [9], we also present another example in Figure 4. The image "toys" was taken from a room without artificial lighting. The result shows that our model is effective in removing unknown type of noise in real color images.

3.2 Image Deblurring

This subsection is devoted to the harder problem of image deblurring in the presence of mixed noise, where H in (1.1) is a convolution operator.

In this experiment, the image's maximum intensity is not rescaled and the blurring kernel is the "disk, 3" kernel generated by the MATLAB command "`fspecial`".

Table 4 summarizes the PSNR results of the deblurring problems in the presence of additive Gaussian noise, Poisson noise, and random-valued impulse noise. Some of the

Image		Cameraman		Baboon		Boat		Bridge		Barbara512	
Gaussian noise ($\hat{\sigma}$)		0	12	0	12	0	12	0	12	0	12
$I_{max} = 120$	r=10%	28.13	26.08	24.87	23.18	27.34	25.14	26.08	24.02	26.75	24.59
	r=20%	26.31	24.96	23.82	22.66	25.92	24.44	24.93	23.41	25.20	23.87
	r=40%	23.38	22.87	22.29	21.71	23.39	22.72	22.64	21.81	23.36	22.66
Gaussian noise ($\hat{\sigma}$)		0	6	0	6	0	6	0	6	0	6
$I_{max} = 60$	r=10%	27.32	25.78	23.86	22.90	26.16	24.82	24.93	23.63	25.48	24.14
	r=20%	25.97	24.63	23.20	22.39	25.10	24.13	24.23	23.11	24.48	23.58
	r=40%	23.06	22.48	21.97	21.49	22.87	22.49	22.15	21.52	22.90	22.42
Gaussian noise ($\hat{\sigma}$)		0	3	0	3	0	3	0	3	0	3
$I_{max} = 30$	r=10%	25.83	24.97	22.95	22.39	24.83	24.19	23.89	23.16	24.21	23.60
	r=20%	24.97	24.15	22.50	22.02	24.09	23.54	23.20	22.55	23.69	23.20
	r=40%	22.38	22.15	21.50	21.13	22.35	21.89	21.29	21.04	22.17	22.10
Gaussian noise ($\hat{\sigma}$)		0	1	0	1	0	1	0	1	0	1
$I_{max} = 10$	r=10%	23.69	23.43	21.73	21.46	22.92	22.59	22.19	21.98	22.84	22.64
	r=20%	23.09	22.77	21.44	21.27	22.48	22.23	21.79	21.46	22.51	22.34
Gaussian noise ($\hat{\sigma}$)		0	0.5	0	0.5	0	0.5	0	0.5	0	0.5
$I_{max} = 5$	r=10%	22.39	22.09	21.12	20.92	21.83	21.60	21.08	21.01	22.15	21.92
	r=20%	21.90	21.56	20.81	20.58	21.32	21.26	20.76	20.66	21.78	21.68
Gaussian noise ($\hat{\sigma}$)		0	0.1	0	0.1	0	0.1	0	0.1	0	0.1
$I_{max} = 1$	r=10%	19.55	19.51	19.61	19.60	19.22	19.44	18.63	18.76	19.99	19.95
	r=20%	19.03	18.82	19.53	19.51	19.00	19.05	18.45	18.37	19.69	19.67

Table 3: Denoising results (PSNR) for various testing images, in the presence of random-valued impulse noise, Gaussian noise and mixed Poisson-Gaussian noise.

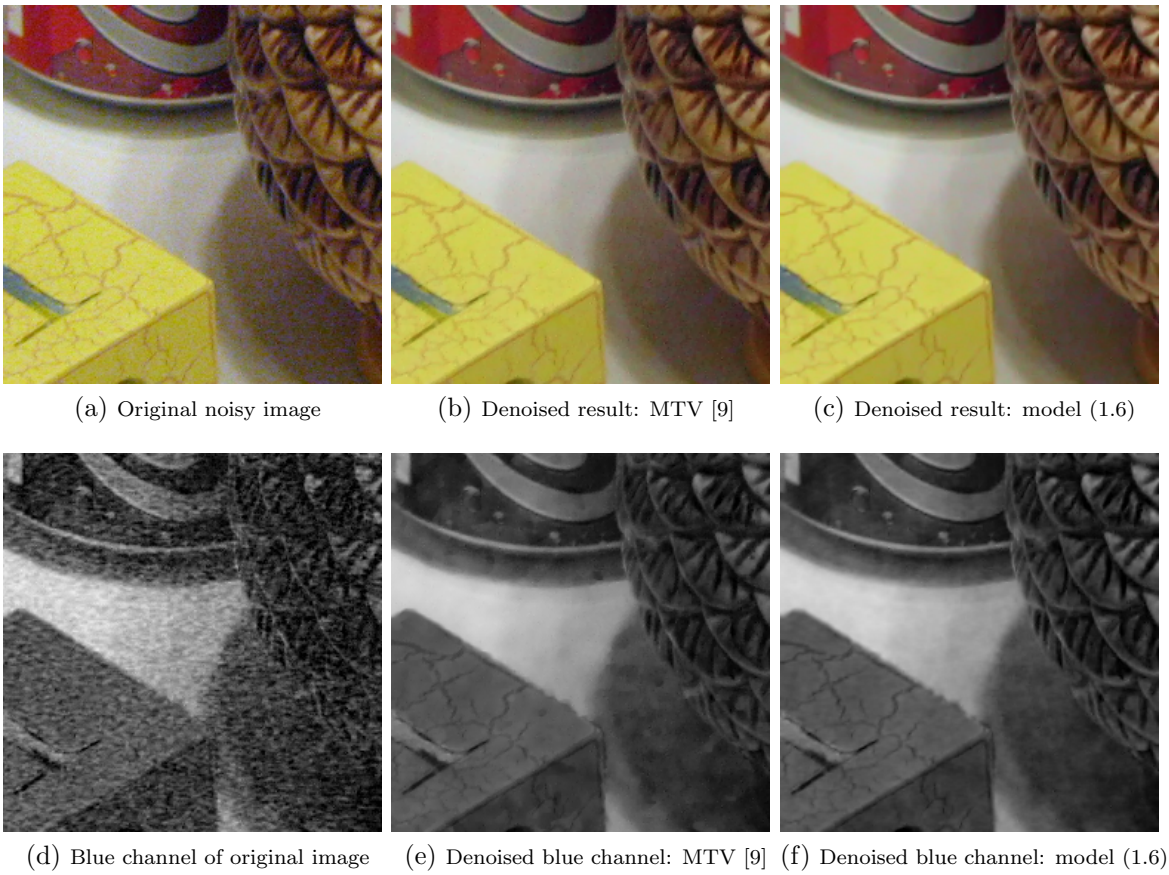
restored images are presented in Figure 5. The results again confirm the effectiveness and robustness of our method in dealing with mixed noises.

Image	Baboon		Goldhill		Cameraman	
Random-valued impulse noise (r)	10%	20%	10%	20%	10%	20%
Poisson+Gaussian	22.34	22.22	25.30	25.00	24.10	23.82
Gaussian+Poisson	22.33	22.22	25.24	24.95	24.12	23.85

Table 4: Deblurring results (PSNR) for various testing images with blurring kernel "disk, 3", in the presence of random-valued impulse noise, Gaussian noise with standard deviation $\hat{\sigma} = 10$ and Poisson noise at image peak intensity of 255. For all cases, the parameters were set to $\lambda_1 = 1$, $\lambda_2 = 0.1$, $\rho = 0.2$.

3.3 Stopping Criteria

In our numerical experiments, the APG algorithm for solving the inner subproblem is stopped when either the relative norm of the gradient or the relative difference of u is smaller than 10^{-5} and 10^{-6} respectively, and for efficiency purpose, the number of APG iterations is capped at 50 for solving each subproblem. We should mention that based on the theoretical stopping criterion (2.8) of the ALM, the inner subproblem must be solved more accurately when the augmented Lagrangian parameter σ_k is larger. In



(a) Original noisy image (b) Denoised result: MTV [9] (c) Denoised result: model (1.6)

(d) Blue channel of original image (e) Denoised blue channel: MTV [9] (f) Denoised blue channel: model (1.6)

Figure 3: Real image denoising (image size: 256×256). The parameters used were $\lambda_1 = 1$, $\lambda_2 = 0.1$ and $\rho = 0.08, 0.1, 1$ for the three channels.

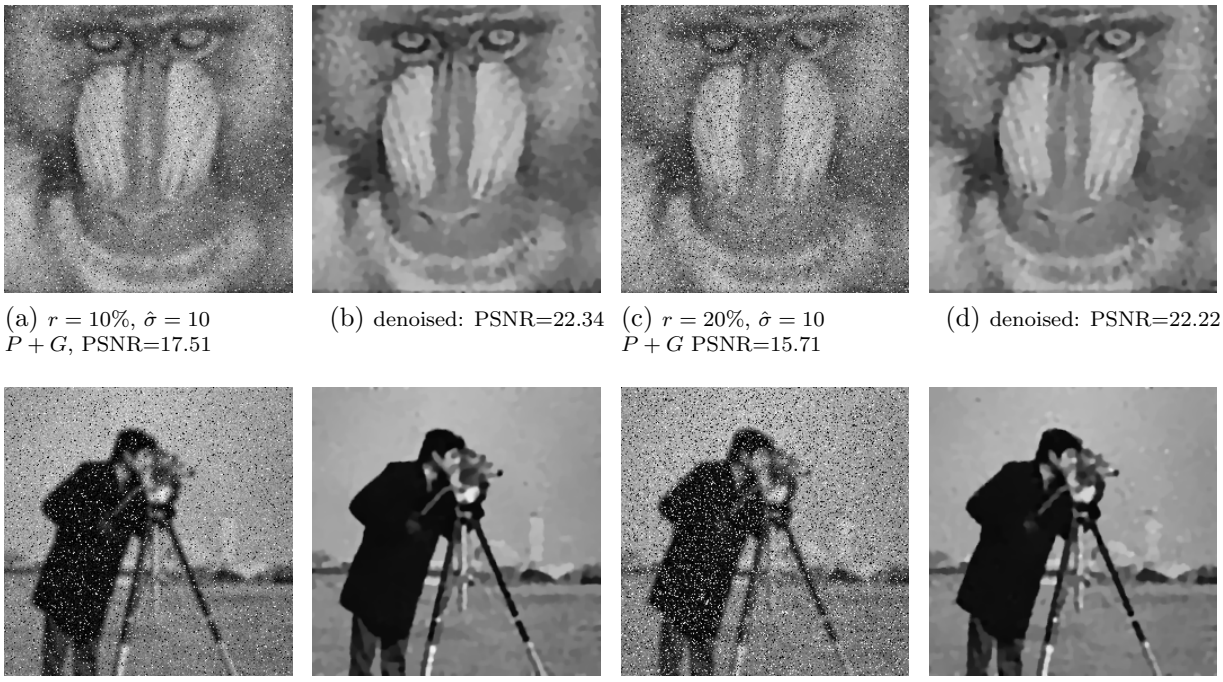
our numerical experiments, the parameter σ_k is typically chosen to be in the range of 10–50, and hence a constant tolerance for solving the inner subproblem is acceptable when k is small, say less than 20.

For the outer ALM iterations, because of multiple degradation factors, there is no generic stopping criterion based on image residue (such as terminating the algorithm when the residue is about the noise level for the pure Gaussian noise case) which can be adopted here. On the other hand, it is far too expensive to terminate the ALM based on the convergence criterion of the dual variable in Step 3 of the ALM, which is in fact not necessary either, since for image restoration, one seeks a moderately accurate sparse solution to (1.6) rather than an optimal solution. Therefore, we simply pre-set the number of outer ALM iterations based on the degradation level of the observed image b. In the cases when the image is very badly damaged by multiple factors, e.g. Poisson noise at low image peak intensity plus blurry effect etc., we terminate the algorithm in 2 to 3 outer iterations; whereas in the cases when only a small percentage of the image pixels are damaged, e.g. with only low level of random-valued impulse noise etc., more iterations help to regain the missing information and we set it to 20

(a) Original noisy "toys"

(b) Denoised "toys" using the proposed model

Figure 4: Real image denoising (image size: 4272×2848). The parameters used were $\lambda_1 = 1$, $\lambda_2 = 0.1$ and $\rho = 0.08, 0.1, 3$ for the three channels respectively. We set $\rho = 3$ for the luminance channel since it is much noisier than the example in Figure 3.



(a) $r = 10\%$, $\hat{\sigma} = 10$ $P + G$, PSNR=17.51 (b) denoised: PSNR=22.34 (c) $r = 20\%$, $\hat{\sigma} = 10$ $P + G$ PSNR=15.71 (d) denoised: PSNR=22.22

(e) $r = 10\%$, $\hat{\sigma} = 10$ $P + G$, PSNR=16.17 (f) denoised: PSNR=24.10 (g) $r = 20\%$, $\hat{\sigma} = 10$ $P + G$ PSNR=13.80 (h) denoised: PSNR=23.82

Figure 5: Deblurring results for various testing images with blurring kernel "disk, 3", in the presence of random-valued impulse noise, Poisson noise and Gaussian noise.

or more. Aside from the two extremes, most of the examples given in the paper are the results obtained after 5 to 7 outer iterations.

In both this section and the next, extensive numerical experiments have been carried out to show the effectiveness and robustness of our model. This also include the robustness of the parameters, in the sense that they are not too sensitive to different images, different noise types, and different noise levels. This means that the parameters are not optimally tuned case by case, and hence, the PSNR may not be the best achievable for every single image.

By pursuing sparse approximation of the underlying solution in the frame domain, some artifacts may inevitably occur. As known and also suggested in [7], a post-processing procedure by passing through a bilateral filter can remove artifacts effectively. To further improve the visual quality of the restored images produced by the algorithm, we build in this post-processing procedure to reduce the possible artifacts. This improves the visual effect of images. However, the improvement in terms of PSNR due to the bilateral filter is actually negligible, as shown in Table 5, for instance.

While the non-smoothness of the model usually makes it difficult to design an efficient solver, we mention that on average, it takes about 200 APG steps in total for our proposed algorithm to obtain a fairly good restored image, which is about 20 to 30 seconds (including the bilateral filtering) for an image of size 256×256 on a MacBook Pro with Intel Core2 Duo 2.26GHz CPU and 4GB RAM. In our numerical simulations,

Image		Baboon		Boat		Bridge		Barbara512	
Random-valued impulse noise (r)		10%	20%	10%	20%	10%	20%	10%	20%
Poisson	w filter	26.31	24.69	28.67	26.83	27.36	25.83	28.44	26.35
	w/o filter	26.31	24.71	28.59	26.81	27.35	25.84	28.40	26.33
Poisson+Gaussian	w filter	25.26	24.08	27.60	26.17	26.38	25.23	27.20	25.58
	w/o filter	25.26	24.09	27.48	26.11	26.38	25.25	27.15	25.56
Gaussian+Poisson	w filter	25.23	24.13	27.57	26.19	26.26	25.06	27.24	25.54
	w/o filter	25.23	24.14	27.46	26.13	26.24	25.07	27.19	25.52

Table 5: Comparison of denoising results (PSNR) for various testing images with/without the bilateral filter, in the presence of random-valued impulse noise, Gaussian noise with standard deviation $\hat{\sigma} = 10$ and Poisson noise at image peak intensity of 255.

all algorithms are implemented in MATLAB (version 7.14).

4 Comparison with Others on Specified Noises

As it has been shown in last section, our proposed model is effective and robust in image restoration with a wide range of mixed noise and unknown noise. We may not expect an image restoration model/algorithm to perform comparably well with those models/algorithms which are designed specifically for a particular noise type. However, as we shall show in this section, for many specified noises, such as additive Gaussian noise, Poisson noise, mixed Poisson-Gaussian noise, multiplicative Gamma noise etc, the results produced by our method are actually comparable to, and sometimes even better than those specialized algorithms. Nevertheless, the purpose of this section is to show that the model (1.6) together with the algorithm given here is versatile in its ability to handle various types of single noise and mixed noise. This robustness is particularly important because in the situations where the noise type is not known a priori, our method has a good chance to produce a reasonable result, while those methods tailored to a specific noise may no longer be reliable.

Note that we do not make comparisons with some of the well known noise removal cases, e.g. Gaussian noise removal, because they are well studied cases and the standard quality of the restored results in terms of PSNR is commonly known. In the case when we compare with other methods, we state the simulation results from the original papers, and the results from our algorithm are generated from images with the same noise level.

4.1 Denoising

4.1.1 Gaussian noise and mixed Gaussian noises

First, we consider the additive mixed Gaussian noises with standard deviation $\hat{\sigma}_m = [2, 5, 10]$, $[4, 10, 20]$, $[6, 15, 30]$ and $[8, 20, 40]$. For comparison purpose, we also give denoising results for single Gaussian noise with standard deviations $\hat{\sigma} = 10$, 20, 30 and 40. The results in terms of PSNR values for the images "Lena" and "Cameraman" are listed in Table 6. For mixed Gaussian noises, although there are no available results for

Gaussian noise ($\hat{\sigma}$)	10	20	30	40
Lena	33.70	30.01	27.85	26.76
Cameraman	33.72	29.91	27.64	26.29
Mixed Gaussian noise ($\hat{\sigma}_m$)	$[2, 5, 10]$	$[4, 10, 20]$	$[6, 15, 30]$	$[8, 20, 40]$
Lena	32.91	29.23	27.37	26.20
Cameraman	32.76	29.29	27.21	25.91

Table 6: Denoising results (PSNR) for the images "Lena" and "Cameraman", in the presence of single Gaussian noise and a mixture of Gaussian noises with different standard deviations. The parameters used were $\lambda_1 = 1$, $\lambda_2 = 0.1 \sim 0.5$ and $\rho = 0.4 \sim 0.7$, where a smaller λ_2 and a larger ρ were chosen as the noise level increased.

comparison, one can observe that the PSNR values are only slightly lower than that of the single Gaussian noise cases with $\hat{\sigma} = \max\{\hat{\sigma}_m\}$. Note that the denoising results for single Gaussian noise cases are obtained by using the same set of parameters for the mixed cases, and are comparable to the results obtained from existing regularization models.

4.1.2 Poisson noise and mixed Poisson-Gaussian noise

In the second experiment, we consider Poisson noise at the image peak intensity ($\mathrm{I}_{\max}$) ranging from 120 to 1, as well as their mixture with a Gaussian noise with standard deviation $\hat{\sigma} = \mathrm{I}_{\max}/10$. The results are presented in Figure 6 and 7, and a comparison of PSNR values with other methods is summarized in Table 7, where we compared our results with two different approaches: the first is the Anscombe variance-stabilizing transform (VST) [1] followed by a white Gaussian noise denoiser [26], and the second is the PURE-LET model proposed in [21], whose fidelity term and minimization algorithm were specially designed for denoising mixed Poisson-Gaussian noise based on the statistical analysis of the Poisson and Gaussian random variables. All PSNR results are directly extracted from [21]. The VST used in our experiment is the generalized Anscombe transform [22] defined as follows

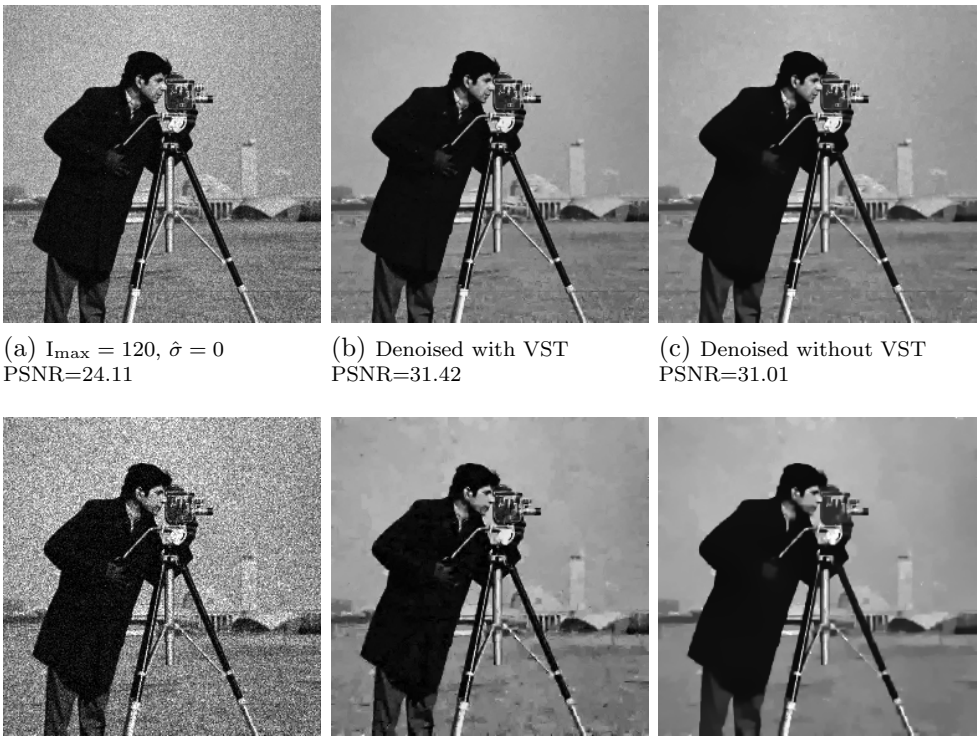
$$u_{VST} = 2\sqrt{u + 3/8 + \hat{\sigma}^2}. \tag{4.1}$$

Note that we provide both results for our model with and without the preprocessing step using VST only for comparison purpose.

The results listed in Table 7 show that when Poisson noise is at moderate levels, i.e. $\mathrm{I}_{\max} = 120, 60, 30$, even without a VST preprocessing step, our method outperforms the VST plus a state-of-the-art multiresolusion-based Gaussian noise reduction algorithm, which consists of a multivariate estimator resulting from a Bayesian least-squares (BLS) optimization, assuming Gaussian scale mixtures (GSM) as a prior for neighborhoods of coefficients at adjacent positions and scales. For the PURE-LET approach, it optimizes a linear expansion of threshold (LET) by relying on a purely data-adaptive unbiased estimate of the mean-squared error, derived from a Poisson-Gaussian unbiased risk estimate (PURE) [21]. We listed two cases here. For the first case, the LET spans on the transformed domain with Haar undecimated wavelet transform (UWT) only,

and for the second case, it spans on both UWT and block discrete cosine transform (BDCT). In the case where the image is corrupted by Poisson noise only, UWT PURE-LET produced slightly better results than our method without VST, while with the VST, our results are even comparable to those of UWT/BDCT PURE-LET. We note that our method is based on piecewise linear B-spline wavelet tight frame and it can easily be extended to a model with two systems including a local DCT, which generally can produce better results, especially for images with rich textures, as demonstrated in [33, 12]. In the case where both Poisson and Gaussian noises are present, the VST preprocessing step becomes redundant for our method, and our results are slightly better than those of UWT/BDCT PURE-LET. This shows that our model is especially effective in removing mixed noises.

In the cases when Poisson noise is at extremely high levels, the variance reduction procedure (4.1) becomes ineffective, and in fact, could be inappropriate. This is because, as studied in [22], the variance of the stabilized Poisson data is approximately equal to 1 irrespective of the mean value of the original data, and for Poisson parameter values under 10, the VST looses control over the bias. Therefore, it tends to underestimate the pixel values when I_{max} is less than 10. However, as shown in Table 7, our method without using VST are still able to produce reasonable results, but about $0.2 \sim 0.5$dB lower than those obtained by the specialized method UWT PURE-LET, which is tailored to the Poisson noise.



(a) $I_{max} = 120$, $\hat{\sigma} = 0$ PSNR=24.11 (b) Denoised with VST PSNR=31.42 (c) Denoised without VST PSNR=31.01

(d) $I_{max} = 30$, $\hat{\sigma} = 0$ PSNR=18.05 (e) Denoised with VST PSNR=27.56 (f) Denoised without VST PSNR=27.44

Figure 6: Denoising results for the image "Cameraman", in the presence of Poisson only.

(a) $I_{max} = 120$, $\hat{\sigma} = 12$ PSNR=18.56 (b) Denoised with VST PSNR=27.99 (c) Denoised without VST PSNR=28.06

(d) $I_{max} = 30$, $\hat{\sigma} = 3$ PSNR=15.92 (e) Denoised with VST PSNR=26.58 (f) Denoised without VST PSNR=26.58

Figure 7: Denoising results for the image "Cameraman", in the presence of both Poisson and Gaussian noises.

4.1.3 Random-valued impulse noise and Gaussian noise

The third example we show is to remove outliers mixed with Gaussian noise. This problem has been considered in [12], where the authors proposed a model that separates the outliers while estimating u:

$$\min_{u,v} \frac{1}{2}\|Hu + v - b\|_2^2 + \lambda_1\|Wu\|_1 + \lambda_2\|v\|_1, \tag{4.2}$$

where H is the identity operator for the denoising case here. In this model, a new variable v is introduced to explicitly represent the outliers in b and the ℓ_1-regularization on v is based on the assumption that the outliers are sparse. The experiments in [12] show that the model (4.2) outperforms the available two-phase approaches (including two-phase methods with pre-detection using the adaptive center-weighted median filter (ACWMF) [10] or ROLD detection [14]), as well as the model with only the ℓ_1 fitting term (1.4) as the data fitting term (introduced in [4, 3] for image deblurring with impulse noise) and solved using a split Bregman method [19]. Since it is shown in [12] that (4.2) compares favorably against all the above mentioned models, here we only take the results from (4.2) for comparison, and all comparing PSNR values are directly extracted from [12].

Peak intensity (I_{max})		120		60		30	
Gaussian noise ($\hat{\sigma}$)		0	12	0	6	0	3
VST+BLS-GSM		30.85	27.56	29.13	27.02	27.54	26.19
UWT PURE-LET		31.03	27.68	29.29	27.14	27.67	26.32
UWT/BDCT PURE-LET		31.35	27.92	29.58	27.37	27.93	26.53
Model (1.6)	without VST	31.01	28.06	29.23	27.61	27.44	26.58
	with VST	31.42	27.99	29.72	27.54	27.56	26.58
Peak intensity (I_{max})		10		5		1	
Gaussian noise ($\hat{\sigma}$)		0	1	0	0.5	0	0.1
VST+BLS-GSM		24.63	24.43	22.50	22.58	14.44	14.63
UWT PURE-LET		25.10	24.56	23.50	23.22	20.44	20.42
UWT/BDCT PURE-LET		25.29	24.74	23.65	23.36	20.48	20.44
Model (1.6)	without VST	25.02	24.60	23.44	23.11	20.09	19.98
	with VST	23.53	24.01	21.64	21.30	8.71	12.07

Table 7: Denoising results (PSNR) for the image "Cameraman", in the presence of single Poisson noise and mixed Poisson-Gaussian noise. The parameters used in the experiments were $\lambda_1 = 1$, $\lambda_2 = 0.1$ and $\rho = 0.2 \sim 1.2$, with ρ increased as I_{max} decreased.

Table 8 and Table 9 summarize the results in terms of PSNR values, from which we can see that our method outperforms the outliers model (4.2), e.g. a gain of 1–2dB in the PSNR values for the images "Baboon" and "Bridge". Some of the restored images are presented in Figure 8. All parameters used in this experiment were the same as in Section 3.1.1, based on the level of random-valued impulse noise r.

Random-valued impulse noise (r)	10%		20%		40%	
Gaussian noise ($\hat{\sigma}$)	0	10	0	10	0	10
Model outlier [12]	30.3	28.4	27.4	26.6	23.6	23.3
Model (1.6)	30.83	28.85	27.57	27.21	24.10	24.17

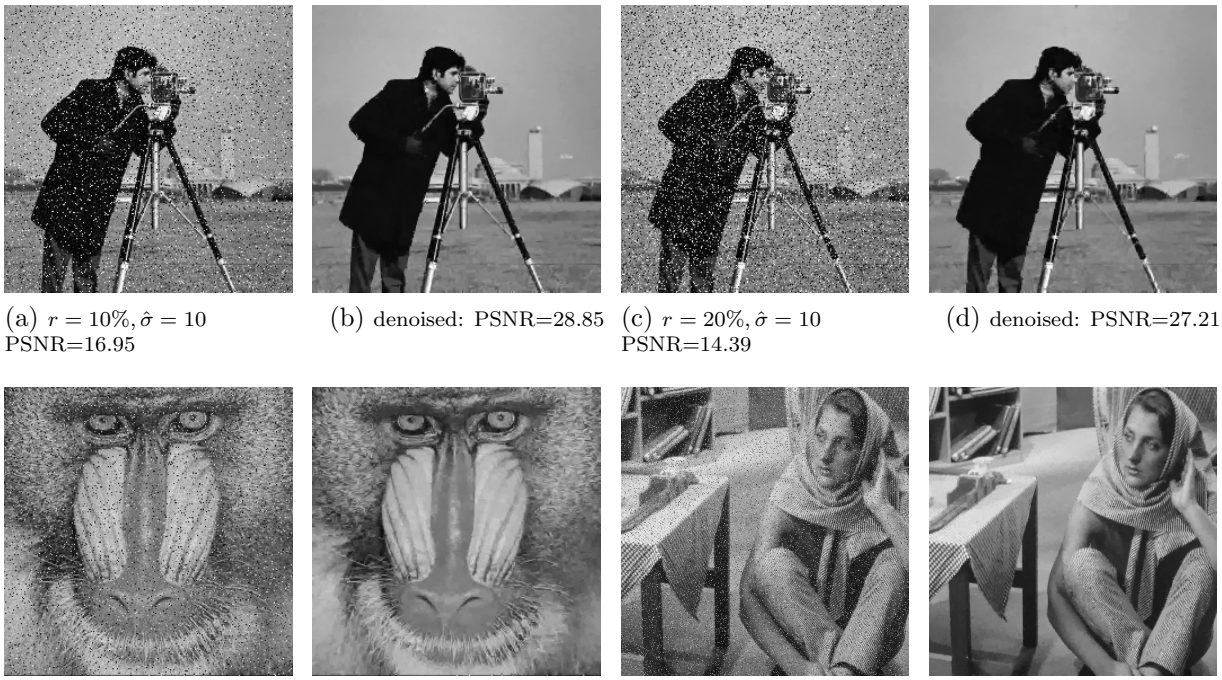
Table 8: Denoising results (PSNR) for the image "Cameraman", in the presence of random-valued impulse noise and Gaussian noise.

Image	Baboon		Boat		Bridge		Barbara512	
Random-valued impulse noise (r) Gaussian noise ($\hat{\sigma} = 10$)	10%	20%	10%	20%	10%	20%	10%	20%
Model outlier [12]	25.1	23.5	28.3	26.4	25.4	23.7	27.9	26.0
Model (1.6)	26.31	24.69	28.67	26.83	27.36	25.83	28.44	26.35

Table 9: Denoising results (PSNR) for various images, in the presence of random-valued impulse noise and Gaussian noise with standard deviation $\hat{\sigma} = 10$.

4.1.4 Multiplicative Gamma noise and mixed Gamma-Gaussian noise

Finally, we test our model on the image "House" with multiplicative Gamma noise and its mixture with a Gaussian noise. For the multiplicative Gamma noise, Aubert



(a) $r = 10\%, \hat{\sigma} = 10$ PSNR=16.95 (b) denoised: PSNR=28.85 (c) $r = 20\%, \hat{\sigma} = 10$ PSNR=14.39 (d) denoised: PSNR=27.21

(e) $r = 10\%, \hat{\sigma} = 10$ PSNR=19.31 (f) denoised: PSNR=26.31 (g) $r = 10\%, \hat{\sigma} = 10$ PSNR=17.80 (h) denoised: PSNR=28.44

Figure 8: Denoising results for various testing images, in the presence of random-valued impulse noise and Gaussian noise.

and Aujol [2] introduced a non-convex data fitting term based on the MAP likelihood estimation approach, i.e.

$$R_1(u) = \sum_i^n \left(\log u_i + \frac{b_i}{u_i} \right). \tag{4.3}$$

Subsequently, several models with an equivalent data fitting term are proposed in order to overcome the numerical difficulties arising from the non-convexity of (4.3), such as the exponential model [34], the I-divegence model [35] and the m-V model [36].

We listed the results in Table 10, where the comparing models are essentially all based on the MAP model (4.3) with a TV regularization: the exponential model [34] uses a logarithmic transformation to overcome the non-convexity of (4.3); the m-V model [36] uses an m-th root transformation to achieve the same purpose; and the I-divergence model [35], whose solution is shown to be theoretically equivalent to that of the exponential model. All PSNR values are extracted from [36].

In this experiment, the noisy images were generated by multiplying a Gamma random variable with shape parameter L and scale parameter $1/L$ to the original noise free image and then, adding a Gaussian noise with standard deviation $\hat{\sigma}$.

Table 10 shows that the results from our model are noticeably better than the others in terms of PSNR and the performance of our model is stable even with an additive Gaussian noise.

Speckle	$L = 1$			$L = 3$		
Gaussian noise ($\hat{\sigma}$)	0	10	20	0	10	20
I-div model	22.5			24.7		
exp model	22.4			24.6		
m-V model	22.6			25.1		
Model (1.6)	22.99	22.88	22.95	25.77	25.69	25.55

Table 10: Denoising results (PSNR) for the image "House", in the presence of a mixture of speckle and Gaussian noise. The parameters used were $\lambda_1 = 1, \lambda_2 = 0.01$, and $\rho = 0.7, 0.5$ for $L = 1, 3$ respectively.

Peak intensity (I_{max})		120		60		30	
Gaussian noise ($\hat{\sigma}$)		0	12	0	6	0	3
Cameraman	disk, 3	24.29	23.36	23.91	23.24	23.43	22.84
Cameraman	motion, 15, 30	22.89	21.74	22.28	21.56	21.69	21.27
Cameraman	Gaussian, 15, 2	23.77	22.87	23.30	22.79	22.86	22.51
Cameraman	average, 9	22.76	21.89	22.30	21.95	21.89	21.62
Lena	disk, 3	26.31	24.89	25.66	24.81	24.93	24.31
Lena	motion, 15, 30	24.24	22.99	23.57	22.72	22.85	22.39
Lena	Gaussian, 15, 2	25.74	24.51	25.24	24.24	24.47	23.94
Lena	average, 9	24.65	23.44	23.97	23.27	23.55	23.10

Table 11: Deblurring results (PSNR) for the images "Cameraman" and "Lena", in the presence of Poisson noise and mixed Poisson-Gaussian noise. The parameters used were $\lambda_1 = 1, \lambda_2 = 0.01, \rho = 0.15$, except for the case $I_{max} = 120, \hat{\sigma} = 0$ where $\rho = 0.1$.

4.2 Deblurring

4.2.1 Poisson noise and mixed Poisson-Gaussian noise

First, we consider the deblurring problems in the presence of Poisson noise and its mixture with Gaussian noise. We use the following 4 blurring kernels generated by the MATLAB command "`fspecial`": "disk, 3" kernel; "15×30 motion" kernel; "15×15 Gaussian" kernel with standard deviation 2; and "9×9 average" kernel. We vary the image's maximum intensity from 120 to 30 and the Gaussian noise level correspondingly as in Section 4.1.2. Note that we did not consider the cases for I_{max} below 30 because blurred images are less likely to be extremely noisy.

Table 11 summarizes the PSNR values we obtained for the deblurring problems with Poisson and Gaussian noises. As far as we are aware of, no such thorough testing have been done in the literature. Note that deblurring in the presence of Poisson noise has been considered in [31]; however, the maximum intensity are scaled up to thousands, thus, the noisiness is much weaker than the cases given here.

4.2.2 Random-valued impulse noise and Gaussian noise

In the second experiment, we consider the problem of deblurring the images contaminated by both impulse noise and Gaussian noise. We again compare our results with that of the outlier model (4.2) proposed in [12].

Image	Baboon		Goldhill		Cameraman	
Random-valued impulse noise (r)	10%	20%	10%	20%	10%	20%
Model outlier [12]	21.2	21.1	25.7	21.4	24.2	24.0
Model (1.6)	22.62	22.49	25.65	25.48	24.48	24.24

Table 12: Deblurring results (PSNR) for various images with blurring kernel "disk, 3", in the presence of random-valued impulse noise, Gaussian noise with standard deviation $\hat{\sigma} = 10$.

The blurring kernel used in this experiment is the "disk, 3" kernel, and the image's maximum intensity is not rescaled.

The results in terms of PSNR are summarized in Table 12, and some of the restored images are presented in Figure 9. We can see that our method generally produces better results, especially for the image "Baboon" for which there is a gain of more than 1dB in the PSNR value. The setting of parameters was the same as in Section 3.2.

(a) $r = 10\%$, $\hat{\sigma} = 10$ PSNR=18.28 (b) $r = 10\%$, $\hat{\sigma} = 10$ PSNR=16.66 (c) $r = 10\%$, $\hat{\sigma} = 10$ PSNR=17.67

(d) restored: PSNR=22.62 (e) restored: PSNR=24.58 (f) restored: PSNR=25.65

Figure 9: Deblurring results for various images with blurring kernel "disk, 3", in the presence of random-valued impulse noise and Gaussian noise.

4.3 Recovery from Images with Randomly Missing Pixels

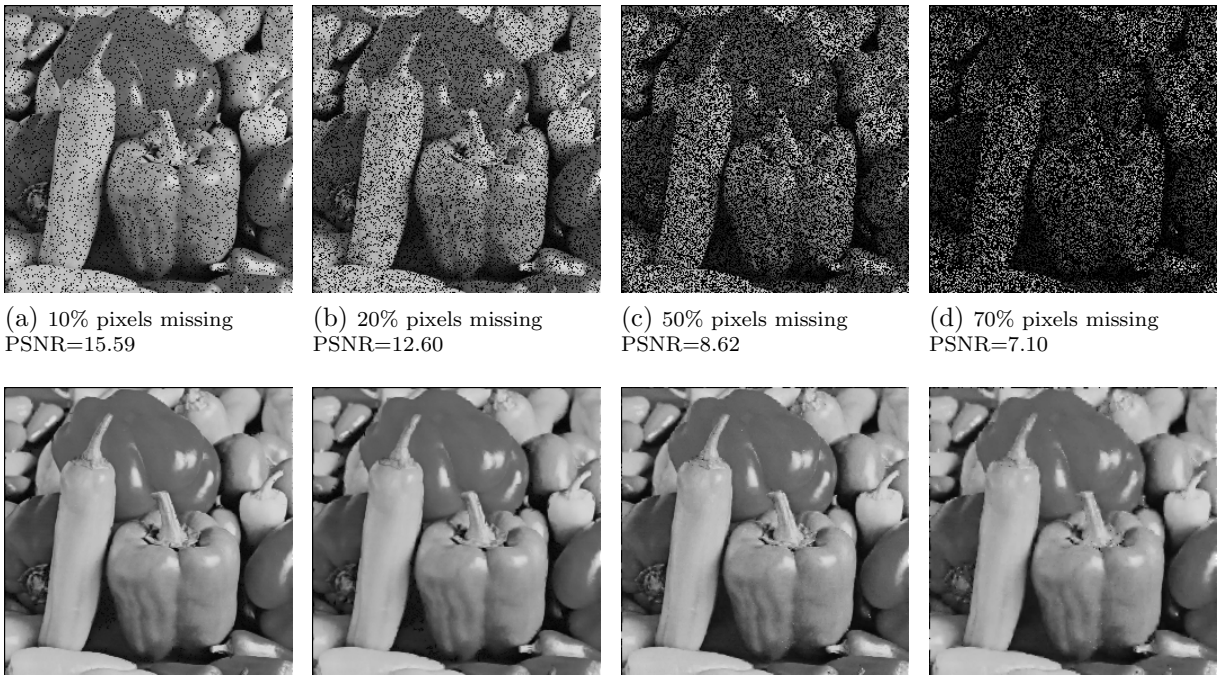
From the results presented in previous sections, we have already seen that our method is robust to outliers. To further demonstrate this, we present in this section the recovery results from images with randomly missing pixels. We consider both cases when the

	Without mask			With mask		
Percentage of pixels missing	10%	20%	30%	50%	70%	90%
Lena	31.81	27.87	24.59	31.33	28.04	23.46
Peppers	33.77	30.04	25.85	30.80	26.20	22.43
Cameraman	29.34	25.56	22.11	28.60	25.34	21.55

Table 13: Recovery results (PSNR) for several images with different percentage of missing pixels. The parameters used were $\lambda_1 = 1, \lambda_2 = 0.01, \rho = 1.2, 1.5, 1.8$ for the recovery of missing $10\%, 20\%, 30\%$ pixels without mask, and $\lambda_1 = 1, \lambda_2 = 0.1, \rho = 0.05, 0.1, 0.15$ for the recovery of missing $50\%, 70\%, 90\%$ pixels with mask.

positions of the missing pixels are known and unknown; in other words, one can refer them as the inpainting and blind inpainting problems respectively.

Table 13 shows that without knowing which pixels are missing, our method is able to recover the original image to a moderate extent with at most 30% missing pixels. However, knowing their exact positions allows us to increase this number to 90%. The visual results for the image "Peppers" are presented in Figure 10.



(a) 10% pixels missing PSNR=15.59
(b) 20% pixels missing PSNR=12.60
(c) 50% pixels missing PSNR=8.62
(d) 70% pixels missing PSNR=7.10
(e) Recovered without mask PSNR=33.77
(f) Recovered without mask PSNR=30.04
(g) Recovered with mask PSNR=30.80
(h) Recovered with mask PSNR=26.20

Figure 10: Recovery results for the image "Peppers" with different percentage of missing pixels.

Gaussian noise ($\hat{\sigma}$)	10	20	30	40
Lena	33.67	29.96	27.81	26.75
Cameraman	33.62	29.84	27.61	26.24
Mixed Gaussian noise ($\hat{\sigma}_m$)	$[2, 5, 10]$	$[4, 10, 20]$	$[6, 15, 30]$	$[8, 20, 40]$
Lena	32.88	29.24	27.35	26.21
Cameraman	32.67	29.28	27.20	25.91

Table 14: Same as Table 6, except that the results are obtained based on the reduced model (5.1) instead on (1.6).

5 Further Remarks

In this section, we propose an even simpler model with only the ℓ_1 fitting function as the data fitting term. In addition, we present comparisons of our proposed ALM-APG algorithm with the widely used alternating direction method of multipliers (ADMM) and split Bregman algorithm.

As we shall observe from the numerical results to be presented later, the reduced model works almost as well as the model (1.6). We believe the reason why the ℓ_2 fitting term is not critically needed in the model (1.6) is because (a) the ALM we use already have the ℓ_2 fitting term implicitly built into the reduced model, and (b) we only seek a good sparse approximate solution (in the frame domain) to the minimization problem (which is just a regularization model for ill-posed image restoration problems) for which a minimizer may not necessarily provide a superior recovered solution for the underlying image restoration problem.

5.1 Reduced Model

For the numerical experiments presented the previous two sections, the parameter λ_2 is usually much smaller than $\lambda_1 = 1$, especially when impulse noise is involved. In fact, for those cases involving only Gaussian noise, where the ℓ_2 fitting function (1.3) is the best choice based on statistical analysis among all possible data fitting terms, reducing λ_2 to 0 in the experiments in subsection 4.1.1 does not change the results much. Table 14 summarizes the PSNR results for the same denoising problems as presented in Table 6. Note that except setting $\lambda_2 = 0$, all the other parameters used were the same for the two experiments.

Moreover, for the denoising of mixed Poisson-Gaussian noise in subsection 4.1.2, where λ_2 is chosen to be relatively larger than in those cases with impulse noise, the results with $\lambda_2 = 0$, as listed in Table 15, are still very close to those presented in Table 7.

The above observations suggest that we can reduce the model (1.6) to the following even simpler version:

$$\min_{u \in \mathbb{R}^n} \ \lambda_1 \|Hu - b\|_1 + \rho \|Wu\|_1. \tag{5.1}$$

This simplification will not reduce the effectiveness of the model (1.6) because the ℓ_2 fitting term is implicitly built in by the proposed ALM. This is readily shown via the augmented Lagrangian functions of the corresponding two models. If we write $y =$

Peak intensity (I_{max})		120		60		30	
Gaussian noise ($\hat{\sigma}$)		0	12	0	6	0	3
Model (5.1)	without VST	30.99	28.10	29.19	27.60	27.44	26.53
	with VST	31.40	27.97	29.69	27.52	27.77	26.49
Peak intensity (I_{max})		10		5		1	
Gaussian noise ($\hat{\sigma}$)		0	1	0	0.5	0	0.1
Model (5.1)	without VST	25.02	24.56	23.35	22.99	19.95	19.94
	with VST	23.58	24.12	22.13	21.57	9.34	13.79

Table 15: Same as Table 7, except that the results are obtained based on the reduced model (5.1) instead of (1.6).

$(y_1; y_2)$ and $z = (z_1; z_2)$, then the augmented Lagrangian function of (1.6) associated with $\sigma > 0$ can be written as

$$\begin{aligned}\mathcal{L}_{2,\sigma}(u, z_1, z_2; y_1, y_2) &= \frac{\lambda_2 + \sigma}{2}\|Hu - b\|^2 + \lambda_1\|z_1\|_1 + \rho\|z_2\|_1 + \langle y_1, b - Hu - z_1\rangle \\ &\quad + \frac{\sigma}{2}\|z_1\|^2 + \sigma\langle Hu - b, z_1\rangle + \langle y_2, -Wu - z_2\rangle + \frac{\sigma}{2}\|Wu + z_2\|^2.\end{aligned}$$

The augmented Lagrangian function of (5.1) can be obtained simply by setting $\lambda_2 = 0$ in the above expression:

$$\begin{aligned}\mathcal{L}_{1,\sigma}(u, z_1, z_2; y_1, y_2) &= \frac{\sigma}{2}\|Hu - b\|^2 + \lambda_1\|z_1\|_1 + \rho\|z_2\|_1 + \langle y_1, b - Hu - z_1\rangle \\ &\quad + \frac{\sigma}{2}\|z_1\|^2 + \sigma\langle Hu - b, z_1\rangle + \langle y_2, -Wu - z_2\rangle + \frac{\sigma}{2}\|Wu + z_2\|^2.\end{aligned}$$

Thus we see that $\mathcal{L}_{1,\sigma}$ also contains the ℓ_2 fitting term $\|Hu - b\|^2$. Although there is a difference of $\lambda_2/2$ in the weight from $\mathcal{L}_{2,\sigma}$, we note that the relative difference is insignificant in our numerical experiments as σ is usually $10 \sim 50$ times larger than λ_2. As a result, the necessity of having the ℓ_2 fitting term in the model (1.6) is often obscured by the augmented Lagrangian based method we use to solve the reduced model (5.1). (We give a more detailed explanation in the next remark.)

If instead of using an ALM, one uses a subgradient method to solve (5.1), then the importance of the ℓ_2 fitting term would become clearer since the subgradient method does not involve an implicit ℓ_2 fitting term. To verify this claim, we used a subgradient method to denoised the 256×256 "Cameraman" image contaminated by a mixed Poisson noise (with $I_{\max} = 255$) and Gaussian noise (with $\hat{\sigma} = 10$) based on the reduced model (5.1) with $\lambda_1 = 1, \rho = 1$. The best PSNR value obtained by the subgradient method (running for 200 iterations) is 28.80, whereas the corresponding PSNR value obtained by ALM-APG when solving (5.1) is 30.46.

We should mention that by running more subgradient (or ALM-APG) iterations can be counter-productive in terms of reducing the PSNR value. Though by performing more iterations, we get a better approximate minimizer (for having a lower objective value) for the model (5.1), but it does not necessarily give a better recovered solution for the original ill-posed image restoration problem. One needs to remember that the model (5.1) is only a regularization model for an ill-posed problem, which is only meant to be used as a guide to find a good recovered solution for the ill-posed problem, and

there is no guarantee that a minimizer of the model would provide a superior recovered solution. Similar remarks also apply to the model (1.6).

Remark 5.1. Observe that the ALM automatically builds in an ℓ_2 data fitting term in each iteration, thus the solution path generated by the proposed ALM when solving the reduced model (5.1) is, therefore, similar to that generated by the ALM for solving the original model (1.6) when the parameter λ_2 is much smaller than σ. More specifically, because the approximate solutions $(u^{k+1}, z_1^{k+1}, z_2^{k+1})$ and $(\bar{u}^{k+1}, \bar{z}_1^{k+1}, \bar{z}_2^{k+1})$ for the models (1.6) and (5.1) are generated from minimizing $\mathcal{L}_{2,\sigma}(u, z_1, z_2; y_1^k, y_2^k)$ and $\mathcal{L}_{1,\sigma}(u, z_1, z_2; \bar{y}_1^k, \bar{y}_2^k)$ respectively, the small relative difference between $\mathcal{L}_{2,\sigma}$ and $\mathcal{L}_{1,\sigma}$ would imply that both models tend to produce similar results when $(y_1^k, y_2^k) \approx (\bar{y}_1^k, \bar{y}_2^k)$. Thus when the ALMs applied to (1.6) and (5.1) both use the same starting iterate $(y_1^0, y_2^0) = (\bar{y}_1^0, \bar{y}_2^0)$ and the same sequence of parameters $\{\sigma_k \gg \lambda_2\}$, the iterates for the first few outer iterations of the ALMs would have the property that $(u^{k+1}, z_1^{k+1}, z_2^{k+1}) \approx (\bar{u}^{k+1}, \bar{z}_1^{k+1}, \bar{z}_2^{k+1})$, and $(y_1^{k+1}, y_2^{k+1}) \approx (\bar{y}_1^{k+1}, \bar{y}_2^{k+1})$. This does not contradict the fact that different optimal solutions for the two models are expected at convergence since eventually (y_1^k, y_2^k) and $(\bar{y}_1^k, \bar{y}_2^k)$ would become different enough for the ALMs to converge to different optimal solutions. Nor does it contradict the less prominent results of (5.1) presented in [13], because different numerical algorithms generate different solution paths and subsequently, lead to different approximate solutions.

Recall that all minimization problems with ℓ_1-regularization terms in image restorations are regularization models for ill-posed problems. The optimization models are used to serve as a guide to obtain a good sparse solution in a certain transformed domain, and the solver is usually terminated when a good sparse approximate solution is obtained instead of running to full convergence to an optimal solution. For our ALMs applied to (1.6) and (5.1), because we only need less than two dozens outer iterations to get good feasible solutions, the solutions obtained for both models would be quite similar based on the explanation given in the previous paragraph. Thus it is not surprising that the results obtained in Tables 14 and 15 are quite close to those in Tables 6 and 7, respectively.

5.2 ALM-APG Versus ADMM

To tackle the non-separable ℓ_1-term in (P), most existing efficient algorithms are variants of the augmented Lagrangian method (ALM) with the inner subproblem (2.14) being solved in an alternating manner, i.e., update x, z alternatingly. In the extreme case where only one alternating iteration is applied to the inner subproblem, the algorithm becomes the well known alternating direction method of multipliers (ADMM) [16, 15], for which the split Bregman algorithm [17, 8] is equivalent to. The detail of the ADMM is as follows: given y^0, z^0 and $\sigma > 0$, iterate the following steps:

$$\begin{aligned}
x^{k+1} &= \arg\min_u \Big\{ F(x) + \frac{\sigma}{2}\|c - Ax - z^k + \sigma^{-1}y^k\|^2 \Big\} \\
z^{k+1} &= \arg\min_z \Big\{ \beta^T|z| + \frac{\sigma}{2}\|c - Ax^{k+1} - z - \sigma^{-1}y^k\|^2 \Big\} \\
y^{k+1} &= y^k + \sigma(c - Ax^{k+1} - z^{k+1}).
\end{aligned}$$

The difference between our algorithm and the ADMM (or the split Bregman algorithm) just mentioned is that by expressing z in terms of x as in (2.17) and substituting it into the objective function, we essentially solve an inner subproblem with only one variable x, which therefore can be solved to a moderate accuracy before updating the Lagrangian dual variable y.

Here we also provide some numerical results from the model solved by ADMM for comparison. The parameter σ used in ADMM algorithm have been tuned and the total number of ADMM iterations is capped at 200, because additional iterations usually do not produce significant improvement. The denoising results in terms of PSNR are summarized in Table 16 and Table 17, and in Table 16, we also listed the CPU runtime. The time recorded is in seconds and includes the bilateral filtering, which takes around 5s. For the cases with Poisson noise, images' maximum intensity were not rescaled and Gaussian noise was added after Poisson noise. The results show that ALM-APG generally performs better than ADMM. Since in [12], the authors also have considered the reduced model (5.1) which is solved by a split Bregman algorithm for denoising of mixed Gaussian noise and random-valued impulse noise, we include their results in the table for reference purpose.

Both ADMM and the ALM-APG algorithm proposed in this paper are first-order methods. However, it is well-known that for many problems, the ADMM may converge to a low accuracy solution very fast but slows down drastically in subsequent iterations; in contrast, the proposed ALM-APG algorithm does not suffer from stagnation in all the numerical experiments conducted in this paper.

Random-valued impulse noise		$r = 10\%$		$r = 20\%$		$r = 40\%$	
Gaussian noise ($\tilde{\sigma}$)		0	10	0	10	0	10
without Poisson noise		PSNR (Time)					
Model (1.6)	ADMM	30.30 (32.8)	28.20 (30.6)	27.36 (32.5)	26.77 (30.0)	23.95 (32.2)	23.82 (30.0)
	ALM-APG	30.83 (33.0)	28.85 (20.5)	27.57 (33.4)	27.21 (21.5)	24.10 (16.2)	24.17 (21.3)
Model (5.1)	ADMM	30.30 (32.3)	28.20 (30.2)	27.36 (32.9)	26.77 (30.0)	23.95 (32.0)	23.83 (30.5)
	ALM-APG	30.82 (32.7)	28.89 (20.6)	27.58 (33.8)	27.21 (20.9)	24.10 (16.4)	24.17 (21.2)
	Split Bregman [12]	29.9	27.5	27.1	26.0	23.1	22.9
with Poisson noise		PSNR (Time)					
Model (1.6)	ADMM	28.26 (30.0)	27.08 (29.8)	26.72 (30.0)	25.99 (30.0)	23.27 (33.7)	23.40 (30.5)
	ALM-APG	29.07 (20.1)	28.31 (17.6)	27.10 (20.9)	26.60 (18.8)	23.46 (19.2)	23.68 (18.7)
Model (5.1)	ADMM	28.25 (30.2)	27.09 (30.4)	26.72 (30.1)	26.00 (29.9)	23.28 (29.8)	23.41 (29.8)
	ALM-APG	29.11 (20.4)	28.36 (17.8)	27.10 (20.9)	26.60 (18.2)	23.48 (19.6)	23.69 (18.8)

Table 16: Denoising results (PSNR) for the image "Cameraman", in the presence of Gaussian noise, Poisson noise and random-valued impulse noise using different models and algorithms.

6 Conclusion

In this paper, we proposed a simple model to solve various image restoration problems in the presence of mixed or unknown noises. The proposed model essentially takes the weighted sum of ℓ_1 and ℓ_2 fitting functions as the data fitting term and utilizes the sparsity prior of images in wavelet tight frame domain. We also designed the ALM-APG algorithm to solve the proposed model numerically, and convergence analysis

Image		Baboon		Boat		Bridge		Barbara512	
Random-valued impulse noise (r)		10%	20%	10%	20%	10%	20%	10%	20%
without Poisson noise									
Model (1.6)	ADMM	25.97	24.68	27.99	26.25	27.11	25.41	27.76	25.93
	ALM-APG	26.31	24.69	28.67	26.83	27.36	25.83	28.44	26.35
Model (5.1)	ADMM	25.97	24.69	27.99	26.26	27.11	25.42	27.76	25.94
	ALM-APG	26.32	24.69	28.68	26.83	27.37	25.83	28.45	26.34
	Split Bregman [12]	24.5	23.2	27.6	26.1	25.0	23.4	27.0	25.5
with Poisson noise									
Model (1.6)	ADMM	24.92	23.75	26.38	24.88	25.85	24.37	26.25	24.78
	ALM-APG	25.26	24.08	27.60	26.17	26.38	25.23	27.20	25.58
Model (5.1)	ADMM	24.92	23.76	26.38	24.90	25.85	24.38	26.26	24.80
	ALM-APG	25.26	24.08	27.60	26.17	26.38	25.24	27.21	25.57

Table 17: Denoising results (PSNR) for various images, in the presence of Poisson noise, Gaussian noise with standard deviation $\hat{\sigma} = 10$ and random-valued impulse noise using different models and numerical algorithms.

of the algorithm is also given. The numerical simulation results show that the performance of the proposed method is surprisingly robust for various image restoration tasks, including denoising, deblurring and inpainting, in the presence of both additive and non-additive noises and their mixtures. This single one-for-all fitting model does not depend on any prior knowledge of the noise. Thus it has the potential of performing effectively in real color image denoising problems, where the noise type is difficult to model. The numerical experiments presented in this paper for real color image denoising have demonstrated the effectiveness and robustness of our model and numerical algorithm.

References

[1] F. J. Anscombe, *The transformation of Poisson, binomial and negative-binomial data*, Biometrika **35** (1948), no. 3/4, 246–254.

[2] G. Aubert and J.-F. Aujol, *A variational approach to remove multiplicative noise*, SIAM Journal on Applied Mathematics **68** (2008), no. 4, 925–946.

[3] L. Bar, N. Kiryati, and N. Sochen, *Image deblurring in the presence of impulse noise*, International Journal of Computer Vision **70** (2006), no. 3, 279–298.

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

[5] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM Journal on Imaging Sciences **2** (2009), no. 1, 183–202.

[6] J.-F. Cai, B. Dong, S. Osher, and Z. Shen, *Image restoration: total variation; wavelet frames; and beyond*, Journal of the American Mathematical Society, Accepted.

[7] J.-F. Cai, S. Osher, and Z. Shen, *Linearized Bregman iteration for frame based image deblurring*, SIAM Journal on Imaging Sciences **2** (2009), no. 1, 226–252.

[8] ———, *Split Bregman methods and frame based image restoration*, Multiscale Modeling and Simulations: A SIAM Interdisciplinary Journal **8** (2009), no. 2, 337–369.

[9] T. Chan, Y. Wang, and H. Zhou, *Denoising natural color photos in digital photography*, Preprint, 2007.

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

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

[12] B. Dong, H. Ji, J. Li, Z. Shen, and Y. Xu, *Wavelet frame based blind image inpainting*, Applied and Computational Harmonic Analysis **32** (2012), no. 2, 268–279.

[13] B. Dong and Z. Shen, *MRA-based wavelet frames and applications*, IAS/Park City Mathematics Series **19** (2010), Summer Program on "The Mathematics of Image Processing".

[14] Y. Dong, R. F. Chan, and S. Xu, *A detection statistic for random-valued impulse noise*, IEEE Transactions on Image Processing **16** (2007), no. 4, 1112–1120.

[15] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*, Computers & Mathematics with Applications **2** (1976), 17–40.

[16] R. Glowinski and A. Marrocco, *Sur l'approximation par éléments finis d'ordre un, et la resolution par pénalisation-dualité d'une classe de problèmes de Dirichlet nonlinéaires*, Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique **9** (1975), no. 2, 41–76.

[17] T. Goldstein and S. Osher, *The split Bregman algorithm for L1 regularized problems*, SIAM Journal on Imaging Sciences **2** (2009), no. 2, 323–343.

[18] H. Ji, C. Liu, Z. Shen, and Y. Xu, *Robust video denoising using low rank matrix completion*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (San Francisco), 2010.

[19] H. Ji, Z. Shen, and Y. Xu, *Wavelet frame based image restoration with missing/damaged pixels*, East Asia Journal on Applied Mathematics **1** (2011), no. 2, 108–131.

[20] Triet Le, Rick Chartrand, and Thomas J. Asaki, *A variational approach to reconstructing images corrupted by poisson noise*, Journal of Mathematical Imaging and Vision **27** (2007), no. 3, 257–263.

[21] Florian Luisier, *Image denoising in mixed Poisson-Gaussian noise*, IEEE Transactions on Image Processing **20** (2011), no. 3, 696–707.

[22] F. Murtagh, J.-L. Starck, and A. Bijaoui, *Image restoration with noise suppression using a multiresolution support*, Astron. Astrophys. Supplement Series **112** (1995), 179–189.

[23] A. Nemirovski and D. Yudin, *Problem Complexity and Method Efficiency in Optimization*, Wiley, New York, 1983.

[24] Y. Nesterov, *A method of solving a convex programming problem with convergence rate* $O(1/k^2)$, Soviet Mathematics Doklady **27** (1983), 372–376.

[25] Yu. Nesterov, *Smooth minimization of non-smooth functions*, Math. Program. **103** (2005), no. 1, Ser. A, 127–152.

[26] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, *Image denoising using scale mixtures of Gaussians in the wavelet domain*, IEEE Transactions on Image Processing **12** (2003), no. 11, 1338–1351.

[27] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, 1970.

[28] ———, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Mathematics of Operations Research **1** (1976), no. 2, 97–116.

[29] ———, *Monotone operators and the proximal point algorithm*, SIAM Journal on Control and Optimization **14** (1976), no. 5, 877–898.

[30] A. Ron and Z. Shen, *Affine systems in* $l_2(\mathbb{R}^d)$*: the analysis of the analysis operator*, Journal of Functional Analysis **148** (1997), 408–447.

[31] S. Setzer, G. Steidl, and T. Teuber, *Deblurring poissonian images by split bregman techniques*, Journal of Visual Communication and Image Representation **21** (2010), 193–199.

[32] Z. Shen, *Wavelet frames and image restorations*, Proceedings of the International Congress of Mathematicians (Hyderabad, India) (Rajendra Bhatia, ed.), vol. IV, Hindustan Book Agency, 2010, pp. 2834–2863.

[33] Z. Shen, K.-C. Toh, and S. Yun, *An accelerated proximal gradient algorithm for frame based image restoration via the balanced approach*, SIAM Journal on Imaging Sciences **4** (2011), 573–596.

[34] J. Shi and S. Osher, *A nonlinear inverse scale space method for a convex multiplicative noise model*, SIAM Journal on Imaging Sciences **1** (2008), 294–321.

[35] G. Steidl and T. Teuber, *Removing multiplicative noise by Douglas-Rachford splitting*, Journal of Mathematical Imaging and Vision **36** (2010), 168–184.

[36] S. Yun and H. Woo, *A new multiplicative denoising variational model based on m-th root transformation*, accepted, 2011.