

Primal-Dual Algorithms for Total Variation Based Image Restoration under Poisson Noise

Dedicated to Professor Lin Qun on the Occasion of his 80th Birthday

You-Wei Wen^{1,*}, Raymond Chan^{2,†} & Tieyong Zeng^{3,‡}

¹*Faculty of Science, Kunming University of Science and Technology, Yunnan, China;*

²*Department of Mathematics, The Chinese University of Hong Kong, Shatin, Hong Kong, China;*

³*Department of Mathematics, Hong Kong Baptist University, Kowloon, Hong Kong, China*

Email: wenyouwei@gmail.com, rchan@math.cuhk.edu.hk, zeng@hkbu.edu.hk

Received November 17, 2014; accepted January 22, 2015

Abstract We consider the problem of restoring images corrupted by Poisson noise. Under the framework of maximum a posteriori estimator, the problem can be converted into a minimization problem where the objective function is composed of a Kullback-Leibler (KL)-divergence term for the Poisson noise and a total variation (TV) regularization term. Due to the logarithm function in the KL-divergence term, the non-differentiability of TV term and the positivity constraint on the images, it is not easy to design stable and efficiency algorithm for the problem. Recently, many researchers proposed to solve the problem by Alternating Direction Method of Multipliers (ADMM). Since the approach introduces some auxiliary variables and requires the solution of some linear systems, the iterative procedure can be complicated. Here we formulate the problem as two new constrained minimax problems and solved them by Chambolle-Pock's first order primal-dual approach. The convergence of our approach is guaranteed by their theory. Comparing with ADMM approaches, our approach requires about half of the auxiliary variables and is matrix-inversion free. Numerical results show that our proposed algorithms are efficient and outperform the ADMM approach.

Keywords Image restoration, Poisson noise, total variation (TV), alternating direction method of multipliers (ADMM), primal-dual, minimax problem

MSC(2010) 65K10, 68U10

Citation: Y. Wen, R. Chan, T. Zeng. Primal-Dual Algorithms for Total Variation Based Image Restoration under Poisson Noise. *Sci China Math*, 2013, 56, doi: 10.1007/s11425-000-0000-0

1 Introduction

Photon noise is a dominant source of image noise in photon-counting devices such as computed tomography, magnetic resonance imaging, electronic microscopy, and astronomical imaging [28, 30, 39, 42]. The noise is directly dependent on the number of photons recorded in the observed image. Since the number of photons depends on the scene brightness and is independent of the photon detections, the magnitude of the photon noise is signal-dependent. The total number of photons collected by image sensors follows a Poisson distribution, and therefore photon noise is also called Poisson noise [41]. In general, the acquired digital images are also degraded due to the relative motion between the sensor and the original scene,

*Research supported in part by NSFC Grant 11361030 and a SRF for ROCS of SEM

†Corresponding author. Research supported by HKRGC Grant CUHK400412, HKRGC AoE Grant AoE/M-05/12, HKRGC Grant HKBU502814.

‡Research supported by HKRGC Grant 211911, 12302714, NSFC 11271049 and FRGs of Hong Kong Baptist University.

defocusing of the sensor systems, and the nature of the sensors. The degradation process of a static scene can be modeled with a spatially linear shift-invariant system, where the original image is convolved with a spatially-invariant point spread function and corrupted by Poisson noise, see [1, 4, 30].

Mathematically, the discrete imaging model of the degradation process can be expressed as follows: an ideal image $\mathbf{f} \in \mathbb{R}^{mn}$ with size $n \times m$ is observed in the presence of a signal-dependent Poisson noise and the observed image $\mathbf{g} \in \mathbb{R}^{mn}$ is given by

$$\mathbf{g} = \text{Poisson}(H\mathbf{f} + b\mathbf{1}), \quad (1.1)$$

where b is a fixed background which can take the value 0, $\mathbf{1}$ is a vector of all ones, and the matrix $H \in \mathbb{R}^{mn \times nm}$ may model a convolution or some other linear observation mechanism, such as emission tomography, see [3, 37]. We denote the $(i + (j - 1)n)$ -th component of the vector \mathbf{f} as $\mathbf{f}_{i,j}$. The value $\mathbf{g}_{i,j}$ of the $(i + (j - 1)n)$ -th entry of the image \mathbf{g} is the realization of a Poisson random variable with expected value $(H\mathbf{f})_{i,j} + b$. In general, the detected value $\mathbf{g}_{i,j}$ is an integer number. When the values $(H\mathbf{f})_{i,j}$ are not sufficiently large, some of the detected values can be zero [4, 44].

We assume that the noise in different pixels is statistically independent, and the joint probability of all the pixels is the product of the probabilities of the individual pixels, i.e.,

$$\Pr(\mathbf{g}|\mathbf{f}) = \prod_{i,j} \frac{[(H\mathbf{f} + b)_{i,j}]^{\mathbf{g}_{i,j}} \exp(-(H\mathbf{f} + b)_{i,j})}{(\mathbf{g}_{i,j})!}.$$

The aim of Poisson noise removal is to recover the original image \mathbf{f} from the degraded image \mathbf{g} . The popular Richardson-Lucy (RL) algorithm [27, 31] was derived from the maximum likelihood approach, where the original image \mathbf{f} is recovered from the degraded image \mathbf{g} by maximizing the likelihood distribution with respect to \mathbf{f} . Maximizing the likelihood distribution is equivalent to minimizing the Kullback-Leibler (KL) divergence $D_{KL}(H\mathbf{f} + b\mathbf{1}, \mathbf{g})$, where the KL-divergence $D_{KL}(\mathbf{z}, \mathbf{g})$ is defined by

$$D_{KL}(\mathbf{z}, \mathbf{g}) = \left\langle \mathbf{g}, \ln \frac{\mathbf{g}}{\mathbf{z}} \right\rangle + \langle \mathbf{1}, \mathbf{z} - \mathbf{g} \rangle, \quad (1.2)$$

with $0 \ln 0 = 0$ and $\ln 0 = -\infty$. Here and for the remainder of the paper, the vector division notation is to be interpreted as component-wise division. Since RL algorithm uses an iterative multiplicative form, it ensures non-negativity if the initial value is nonnegative. However, it is known that the convergence of the RL algorithm is slow. Also, the RL algorithm will amplify noise after a few iterations because restoring the image \mathbf{f} is a very ill-conditioned problem where small perturbations in the observed image \mathbf{g} can produce large perturbations in the restored image \mathbf{f} [24].

To deal with the ill-conditioning problem, a regularization term should be added to control noise and artifacts in the recovered images. The regularization term is designed to represent the prior of the original image. From a statistical point of view, this approach can be derived by maximum a posterior estimation. Instead of maximizing only the likelihood, a prior of the original image is considered and we maximize the a posterior probability $\Pr(\mathbf{f}|\mathbf{g})$. The total variation (TV) norm, proposed by Rudin, Osher and Fatemi [34], has become very popular to represent the prior of images due to its ability to preserve edges in images. The discrete TV of \mathbf{f} is defined as

$$\text{TV}(\mathbf{f}) = \sum_{i,j} |(\nabla \mathbf{f})_{i,j}|, \quad (1.3)$$

where the discrete gradient operator ∇ is defined by $(\nabla \mathbf{f})_{i,j} = ((\nabla_x \mathbf{f})_{i,j}, \nabla_y \mathbf{f})_{i,j}$ with

$$(\nabla_x \mathbf{f})_{i,j} = \mathbf{f}_{i+1,j} - \mathbf{f}_{i,j}, \quad (\nabla_y \mathbf{f})_{i,j} = \mathbf{f}_{i,j+1} - \mathbf{f}_{i,j}, \quad i = 1, \dots, n, \quad j = 1, \dots, m.$$

We can apply the circulant boundary condition or the reflective boundary condition to extend the values of \mathbf{f} . For the 2-tuple $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2) \in \mathbb{R}^2$, we define $|\mathbf{a}| = \sqrt{\mathbf{a}_1^2 + \mathbf{a}_2^2}$.

Using the Bayesian rule, the Poisson image restoration problem can be represented as a minimization problem

$$\min_{\mathbf{f} \in \mathcal{S}} \Psi(\mathbf{f}) \equiv D_{KL}(H\mathbf{f} + b\mathbf{1}, \mathbf{g}) + \lambda \text{TV}(\mathbf{f}), \quad (1.4)$$

where $\mathbf{b} = b\mathbf{1}$, λ is a regularization parameter which measures the trade off between a good fit and a regularized solution, and the set S is defined by

$$S = \{\mathbf{f} : 0 \leq \mathbf{f} \leq M\mathbf{1}\} \quad (1.5)$$

for some positive M . Since \mathbf{f} represents the intensity of the unknown image, it is commonly assumed that $0 \leq \mathbf{f} \leq M\mathbf{1}$, i.e., $\mathbf{f} \in S$.

Difficulties on computational stability and efficiency for solving the minimization problem (1.4) arise due to the following reasons. Firstly, the KL-divergence functional is non-quadratic and contains a logarithm function (see (1.2)). Image restoration algorithms based on quadratic data fitting term can not be readily extended to solve the problem. Secondly, the TV regularization term is non-differentiable and it is difficult to solve the Euler-Lagrange equation associated with the minimization problem (1.4). Thirdly, in order to obtain a physically-meaningful solution, a box-constraint (see (1.5)) has to be imposed in the recovered image \mathbf{f} . Recently several efficient methods were proposed to solve the constrained TV- D_{KL} problem (1.4). These methods include the gradient decent method [14, 26], the Poisson Bregman-EM-TV approaches [9], alternating direction method of multipliers (ADMM) [21, 36], primal-dual approaches [10, 13] etc. Bardsley and Golde [2] proposed to approximate the KL divergence using a quadratic Taylor series approximation.

It is well known that the convergence rate of the gradient descent method is very slow. The variable splitting methods in [9, 21, 36] need to introduce some auxiliary variables where each variable is of the same size as the given image. Some of these methods require the inversion of matrices involving H and ∇ . These make the algorithms less efficient. We remark that H is easily invertible only when it is a spatially-invariant blur under circulant or reflective boundary condition assumption [29]. For some applications such as magnetic resonance imaging, the matrices may be extremely difficult to invert, see e.g. [19, 20].

In this paper, we focus on developing convergent numerical schemes to solve the constrained minimization problem (1.4) that require less storage and no matrix inversion. Our idea is to reformulate (1.4) as new constrained minimax problems and then apply the Chambolle-Pock first order primal-dual approach [12] to compute the saddle points of the resulting problems. Comparing to existing ADMMs, our approach is simpler since there is no matrix inversions and there are fewer auxiliary variables in our algorithms. For example, the storage required for auxiliary variables for the ADMM in [35] and [21] are $O(8mn)$ and $O(6mn)$ respectively. For the two algorithms we proposed in this paper, the storage for auxiliary variables are $O(4nm)$ and $O(3mn)$ respectively.

The outline of this paper is as follows. In Section 2, we review ADMM on TV-based Poisson image restoration problem (1.4). In Section 3, we transform (1.4) into two different equivalent minimax problems and then apply Chambolle-Pock's primal-dual scheme to solve them. In Section 4, numerical results are given to show the efficiency of our proposed algorithms. Finally, some concluding remarks are given in Section 5.

2 Alternating Direction Method for Multipliers

In this section, we review the alternating direction method for multiplier (ADMM) on TV-based Poisson noise removal.

2.1 ADMM

ADMM can be viewed as an application of the Douglas-Rachford splitting algorithm [18]. We consider the convex problem of the form

$$\min_{\mathbf{x}} \Phi_1(\mathbf{x}) + \Phi_2(G\mathbf{x}), \quad (2.1)$$

here $\Phi_i, i = 1, 2$ are given closed proper convex functions and G is a given linear transform. The ADMM introduces an auxiliary variable $\mathbf{z} = G\mathbf{x}$, and consider solving the augmented Lagrangian \mathcal{L}_1

$$\mathcal{L}_1(\mathbf{x}, \mathbf{z}, \boldsymbol{\beta}) = \Phi_1(\mathbf{x}) + \Phi_2(\mathbf{z}) - \langle \boldsymbol{\beta}, G\mathbf{x} - \mathbf{z} \rangle + \frac{1}{2\alpha} \|G\mathbf{x} - \mathbf{z}\|_2^2$$

by alternating direction method. Here $\boldsymbol{\beta}$ is the Lagrangian multiplier associated with the equality constraint $\mathbf{z} = G\mathbf{x}$, and α is the penalty parameter for the violation of the constraint. Let $\mathbf{d} = \alpha\boldsymbol{\beta}$, the corresponding ADMM is then defined as

$$\begin{cases} \mathbf{x}^{(k+1)} &= \operatorname{argmin}_{\mathbf{x}} \Phi_1(\mathbf{x}) + \frac{1}{2\alpha} \|G\mathbf{x} - \mathbf{z}^{(k)} - \mathbf{d}^{(k)}\|_2^2, \\ \mathbf{z}^{(k+1)} &= \operatorname{argmin}_{\mathbf{z}} \Phi_2(\mathbf{z}) + \frac{1}{2\alpha} \|G\mathbf{x}^{(k+1)} - \mathbf{z} - \mathbf{d}^{(k)}\|_2^2, \\ \mathbf{d}^{(k+1)} &= \mathbf{d}^{(k)} - (G\mathbf{x}^{(k+1)} - \mathbf{z}^{(k+1)}). \end{cases} \quad (2.2)$$

Algorithm (2.2) consists in finding the saddle point of $\mathcal{L}_1(\mathbf{x}, \mathbf{z}, \mathbf{d})$ by minimizing it in a consecutive order with respect to the variables \mathbf{x} and \mathbf{z} , and then updating the multipliers \mathbf{d} by a proximal point approach [32, 33]. It is discovered in [38, 43] that Algorithm (2.2) coincides with the split-Bregman algorithm [23] with the only difference that the iterates $\mathbf{d}^{(k+1)}$ are scaled by α .

Notice that the variable \mathbf{x} in (2.2) requires only $(\mathbf{z}^{(k)}, \mathbf{d}^{(k)})$ to generate the iteration $\mathbf{x}^{(k+1)}$, it can therefore be regarded as an intermediate variable in algorithm (2.2), see [8]. Denote $\mathbf{v}^{(k)} = (\mathbf{z}^{(k)}, \mathbf{d}^{(k)})$, He and Yuan [25] showed that the sequence $\{\|\mathbf{v}^{(k+1)} - \mathbf{v}^{(k)}\|_2^2\}$ is monotonically non-increasing and therefore it can be used as the stopping criterion.

2.2 Poisson Image Restoration by Augmented Lagrangian

Notice that the objective function in the standard ADMM is the sum of two functions, see (2.1). However, our problem (1.4) involves a box-constraint, a KL-divergence term and a TV regularization term. We therefore need to find a way of mapping (1.4) into the standard form (2.1).

By introducing an indicator function, (1.4) can be reformulated as

$$\Psi_1(\mathbf{f}) = D_{KL}(H\mathbf{f} + \mathbf{b}, \mathbf{g}) + \lambda\text{TV}(\mathbf{f}) + \delta_S(\mathbf{f}),$$

where $\delta_S(\mathbf{x})$ is the indicator function of S given in (1.5):

$$\delta_S(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in S, \\ \infty, & \text{otherwise.} \end{cases} \quad (2.3)$$

In [21], the authors introduced the variables $\mathbf{u}_1 = H\mathbf{f}, \mathbf{u}_2 = \mathbf{f}, \mathbf{u}_3 = \mathbf{f}$ and grouped the variables into two blocks: $\mathbf{f} \in \mathbb{R}^{mn}$ and $\mathbf{u} = (\mathbf{u}_1^T, \mathbf{u}_2^T, \mathbf{u}_3^T)^T \in \mathbb{R}^{3mn}$. Setting $G = \begin{pmatrix} H^T & I & I \end{pmatrix}^T$,

$$\Phi_1(\mathbf{f}) = 0 \quad \text{and} \quad \Phi_2(\mathbf{u}) = D_{KL}(\mathbf{u}_1 + \mathbf{b}, \mathbf{g}) + \lambda\text{TV}(\mathbf{u}_2) + \delta(\mathbf{u}_3),$$

the multi-block problem is converted into a two-block problem

$$\min \Phi_1(\mathbf{f}) + \Phi_2(\mathbf{u}), \quad \text{subject to } \mathbf{u} = G\mathbf{f}.$$

Hence (1.4) completely fits into the framework of (2.1). The iterative scheme can be written as

$$\begin{cases} \mathbf{f}^{(k+1)} = \operatorname{argmin}_{\mathbf{f}} \left\| H\mathbf{f} - \mathbf{u}_1^{(k)} - \mathbf{d}_1^{(k)} \right\|_2^2 + \left\| \mathbf{f} - \mathbf{u}_2^{(k)} - \mathbf{d}_2^{(k)} \right\|_2^2 + \left\| \mathbf{f} - \mathbf{u}_3^{(k)} - \mathbf{d}_3^{(k)} \right\|_2^2, \\ \mathbf{u}_1^{(k+1)} = \operatorname{argmin}_{\mathbf{u}_1} D_{KL}(\mathbf{u}_1 + \mathbf{b}, \mathbf{g}) + \frac{1}{2\alpha} \left\| H\mathbf{f}^{(k+1)} - \mathbf{u}_1 - \mathbf{d}_1^{(k)} \right\|_2^2, \\ \mathbf{u}_2^{(k+1)} = \operatorname{argmin}_{\mathbf{u}_2} \lambda\text{TV}(\mathbf{u}_2) + \frac{1}{2\alpha} \left\| \mathbf{f}^{(k+1)} - \mathbf{u}_2 - \mathbf{d}_2^{(k)} \right\|_2^2, \\ \mathbf{u}_3^{(k+1)} = \operatorname{argmin}_{\mathbf{u}_3} \delta(\mathbf{u}_3) + \frac{1}{2\alpha} \left\| \mathbf{f}^{(k+1)} - \mathbf{u}_3 - \mathbf{d}_3^{(k)} \right\|_2^2, \\ \mathbf{d}_1^{(k+1)} = \mathbf{d}_1^{(k)} - (H\mathbf{f}^{(k+1)} - \mathbf{u}_1^{(k+1)}), \\ \mathbf{d}_2^{(k+1)} = \mathbf{d}_2^{(k)} - (\mathbf{f}^{(k+1)} - \mathbf{u}_2^{(k+1)}), \\ \mathbf{d}_3^{(k+1)} = \mathbf{d}_3^{(k)} - (\mathbf{f}^{(k+1)} - \mathbf{u}_3^{(k+1)}). \end{cases} \quad (2.4)$$

We note that for the subproblem of the variable \mathbf{u} , the variables $\{\mathbf{u}_i\}_{i=1}^3$ are separable and can be solved separately as in (2.4). The subproblem for the variable \mathbf{f} is equivalent to a least squares problem. The minimization for \mathbf{u}_1 and \mathbf{u}_3 are the Moreau proximity operator for the KL-divergence function and the indicator function respectively. The minimization for the variable \mathbf{u}_2 can be expressed as a TV-denoising problem, where the Chambolle projection method [11] can be applied to obtain an approximate solution. Besides the storage required for the matrix H , the observed image \mathbf{g} and the restored image \mathbf{f} , the algorithm requires additionally $O(6mn)$ storage for the auxiliary variables $\{\mathbf{u}_i\}_{i=1}^3$ and $\{\mathbf{d}_i\}_{i=1}^3$.

The authors in [35] applied another way to map (1.4) into (2.1). They introduced the variables: $\mathbf{u}_1 = H\mathbf{f}$, $\mathbf{u}_2 = \nabla\mathbf{f}$, $\mathbf{u}_3 = \mathbf{f}$, and defined the linear operator G as $G = \begin{pmatrix} H^T & \nabla^T & I \end{pmatrix}^T$. Correspondingly, the function $\Phi_2(\mathbf{u})$ was defined as

$$\Phi_2(\mathbf{u}) = D_{KL}(\mathbf{u}_1 + \mathbf{b}, \mathbf{g}) + \lambda \|\mathbf{u}_2\|_1 + \delta(\mathbf{u}_3).$$

The corresponding iterative scheme can be written as

$$\begin{cases} \mathbf{f}^{(k+1)} &= \operatorname{argmin}_{\mathbf{f}} \left\| H\mathbf{f} - \mathbf{u}_1^{(k)} - \mathbf{d}_1^{(k)} \right\|_2^2 + \left\| \nabla\mathbf{f} - \mathbf{u}_2^{(k)} - \mathbf{d}_2^{(k)} \right\|_2^2 + \left\| \mathbf{f} - \mathbf{u}_3^{(k)} - \mathbf{d}_3^{(k)} \right\|_2^2 \\ \mathbf{u}_1^{(k+1)} &= \operatorname{argmin}_{\mathbf{u}_1} D_{KL}(\mathbf{u}_1 + \mathbf{b}, \mathbf{g}) + \frac{1}{2\alpha} \left\| H\mathbf{f}^{(k+1)} - \mathbf{u}_1 - \mathbf{d}_1^{(k)} \right\|_2^2, \\ \mathbf{u}_2^{(k+1)} &= \operatorname{argmin}_{\mathbf{u}_2} \lambda \|\mathbf{u}_2\|_1 + \frac{1}{2\alpha} \left\| \nabla\mathbf{f}^{(k+1)} - \mathbf{u}_2 - \mathbf{d}_2^{(k)} \right\|_2^2, \\ \mathbf{u}_3^{(k+1)} &= \operatorname{argmin}_{\mathbf{u}_3} \delta(\mathbf{u}_3) + \frac{1}{2\alpha} \left\| \mathbf{f}^{(k+1)} - \mathbf{u}_3 - \mathbf{d}_3^{(k)} \right\|_2^2, \\ \mathbf{d}_1^{(k+1)} &= \mathbf{d}_1^{(k)} - (H\mathbf{f}^{(k+1)} - \mathbf{u}_1^{(k+1)}), \\ \mathbf{d}_2^{(k+1)} &= \mathbf{d}_2^{(k)} - (\nabla\mathbf{f}^{(k+1)} - \mathbf{u}_2^{(k+1)}), \\ \mathbf{d}_3^{(k+1)} &= \mathbf{d}_3^{(k)} - (\mathbf{f}^{(k+1)} - \mathbf{u}_3^{(k+1)}). \end{cases} \quad (2.5)$$

Note that $\mathbf{u}_2 \in \mathbb{R}^{2mn}$ and $\mathbf{d}_2 \in \mathbb{R}^{2mn}$. Hence the storage required for auxiliary variables is $O(8mn)$. However, each variable in (2.5) has a closed-form solution.

3 Primal-Dual Approach

In this section, we find the minimizer of the Poisson image restoration problem (1.4) by transforming it into two different minimax problems and then solve them by a primal-dual method. Primal-dual methods [12, 15–18, 32, 33, 40, 46, 47], where the primal variable and the dual variable are solved alternatively, were widely applied to compute the saddle point of minimax problems. In [46, 47], a gradient descent method is employed to find the primal and the dual variables alternatively. In [15], a predictor-corrector scheme is used in the alternating direction iterations for finding the dual variable. The Chambolle-Pock first order primal-dual algorithm [12] resembles in some way the dual method in [22] which uses a predictor-corrector scheme [15] in the alternating direction iterations for the dual variable. We will apply the algorithm in [12] to seek the saddle point of our minimax problems. We therefore give a brief introduction of the method here.

3.1 Chambolle-Pock's first-order primal-dual algorithm

In [12], Chambolle and Pock considered solving the minimax problem:

$$\min_{\mathbf{x} \in X} \max_{\mathbf{z} \in Z} \phi(\mathbf{x}) + \langle K\mathbf{x}, \mathbf{z} \rangle - \psi(\mathbf{z}). \quad (3.1)$$

Here $X, Z \subseteq \mathbb{R}^{nm}$, ϕ, ψ are proper, convex and lower semi-continuous functions, and K is a linear operator with induced norm $\|K\|$. They proposed to solve the problem by a first-order primal-dual

algorithm as follows:

$$\begin{cases} \mathbf{x}^{(k+1)} &= \operatorname{argmin}_{\mathbf{x} \in X} \phi(\mathbf{x}) + \langle K\mathbf{x}, \mathbf{z} \rangle + \frac{1}{2s} \|\mathbf{x} - \mathbf{x}^{(k)}\|_2^2, \\ \widehat{\mathbf{x}}^{(k+1)} &= \mathbf{x}^{(k+1)} + \theta(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}), \\ \mathbf{z}^{(k+1)} &= \operatorname{argmax}_{\mathbf{z} \in Z} \langle K\widehat{\mathbf{x}}^{(k+1)}, \mathbf{z} \rangle - \psi(\mathbf{z}) - \frac{1}{2t} \|\mathbf{z} - \mathbf{z}^{(k)}\|_2^2. \end{cases} \quad (3.2)$$

The parameters $s, t > 0$ are step sizes of the primal and dual variables respectively, and θ is the combination parameter. In the iterative procedure, proximal-point iterations are applied to the sub-differentials of the \mathbf{x} and \mathbf{z} subproblems in (3.2) alternately with the primal variable and the dual variable fixed. The iterative scheme can also be equivalently interpreted as a first-order primal–dual relaxed Arrow-Hurwitz algorithm.

This primal-dual algorithm has several advantages. First of all, it enjoys convergence with rate $O(1/k)$ when $\theta = 1$ and the step sizes satisfy $st < 1/\|K^T K\|$. Secondly, the algorithm is matrix inversion-free if the functions $\phi(\mathbf{x})$ and $\psi(\mathbf{z})$ are separable for the variables \mathbf{x} and \mathbf{z} respectively. This is very useful when the matrix is huge and difficult to invert, e.g. in magnetic resonance imaging [19, 20].

We will transform the Poisson image restoration problem (1.4) into two different minimax problems and apply (3.2) to solve them.

3.2 First Minimax Problem

Let us describe the notations that we will be using in the followings. For $\mathbf{p} \in \mathbb{R}^{nm} \times \mathbb{R}^{nm}$, $\mathbf{p}_{i,j} = (\mathbf{p}_{i,j,1}, \mathbf{p}_{i,j,2}) \in \mathbb{R}^2$ denotes the $(i + (j - 1)n)$ -th component of \mathbf{p} . Define the inner product $\langle \mathbf{f}, \mathbf{g} \rangle = \sum_{i,j} \mathbf{f}_{i,j} \mathbf{g}_{i,j}$ for $\mathbf{f}, \mathbf{g} \in \mathbb{R}^{nm}$, $\langle \mathbf{p}, \mathbf{q} \rangle = \sum_{i,j} \mathbf{p}_{i,j} \mathbf{q}_{i,j}$ for $\mathbf{p}, \mathbf{q} \in \mathbb{R}^{nm} \times \mathbb{R}^{nm}$. Define $\|\mathbf{p}\|_2 = \max_{i,j} |\mathbf{p}_{i,j}|$ and $\operatorname{div} = -\nabla^T$ as the discrete version of the divergence operator, where ∇^T is the adjoint of ∇ . We represent the TV norm using the dual form, i.e.,

$$\operatorname{TV}(\mathbf{f}) = \max_{\mathbf{p} \in A} \langle \mathbf{f}, \operatorname{div} \mathbf{p} \rangle,$$

where the set A is given by

$$A \equiv \{\mathbf{p} \in \mathbb{R}^{nm} \times \mathbb{R}^{nm} : \|\mathbf{p}\|_\infty \leq 1\}. \quad (3.3)$$

Introducing the auxiliary variable $\mathbf{u} = H\mathbf{f} + \mathbf{b}$, problem (1.4) can be written as the following minimax problem

$$\min_{\mathbf{u}=H\mathbf{f}+\mathbf{b}} \max_{\mathbf{p} \in A} \gamma D_{KL}(\mathbf{u}, \mathbf{g}) + \langle \mathbf{f}, \operatorname{div} \mathbf{p} \rangle + \delta_A(\mathbf{f}), \quad (3.4)$$

where $\gamma = 1/\lambda$ and $\delta_A(\cdot)$ is defined in (2.3). The regularization parameter λ (or γ) is chosen to satisfy the discrepancy principle. According to (1.4), we have $\Psi(\mathbf{f}) = \lambda(\gamma D_{KL}(H\mathbf{f} + \mathbf{b}, \mathbf{g}) + \operatorname{TV}(\mathbf{f}))$. By the implicit function theorem, the minimizer \mathbf{f} is a continuous function with respect to γ (or λ). The discrepancy principle is a method to select the regularization parameter. The idea behind the discrepancy principle for Poisson noise removal is to choose a γ such that the minimizer \mathbf{f} satisfies $D_{KL}(H\mathbf{f}(\gamma), \mathbf{g}) = nm/2$, see [44]. In this sense, the parameter γ in (3.4) can be regarded as a Lagrange multiplier of the constraint $D_{KL}(H\mathbf{f}(\gamma), \mathbf{g}) = mn/2$.

We define the Lagrangian function,

$$\mathcal{L}(\mathbf{u}, \mathbf{f}, \mathbf{y}, \mathbf{p}) = \gamma D_{KL}(\mathbf{u}, \mathbf{g}) + \langle \mathbf{y}, \mathbf{u} - H\mathbf{f} - \mathbf{b} \rangle + \langle \mathbf{f}, \operatorname{div} \mathbf{p} \rangle + \delta(\mathbf{f}), \quad (3.5)$$

where \mathbf{y} is the Lagrangian multiplier associated with the equality constraint $\mathbf{u} = H\mathbf{f} + \mathbf{b}$. The next theorem states that the Poisson image restoration problem (1.4) can be written as

$$\min_{\mathbf{f}, \mathbf{u}} \max_{\mathbf{p} \in A, \mathbf{y}} \mathcal{L}(\mathbf{u}, \mathbf{f}, \mathbf{y}, \mathbf{p}) \quad (3.6)$$

Theorem 3.1. Define the function $q(\mathbf{u}, \mathbf{f}) : \mathbb{R}^{nm} \rightarrow (-\infty, \infty]$,

$$q(\mathbf{u}, \mathbf{f}) = \begin{cases} \max_{\mathbf{p} \in A, \mathbf{y}} \mathcal{L}(\mathbf{u}, \mathbf{f}, \mathbf{y}, \mathbf{p}), & \text{if } \mathbf{f} \in S, \\ \infty, & \text{otherwise.} \end{cases}$$

Then we have

$$\min_{\mathbf{f} \in S} \Psi(\mathbf{f}) = \min_{\mathbf{f} \in S, \mathbf{u}} \lambda q(\mathbf{u}, \mathbf{f}).$$

Proof. It is easy to check that if $\mathbf{u} = H\mathbf{f} + \mathbf{b}$ and $\mathbf{f} \in S$, we have $q(\mathbf{u}, \mathbf{f}) = \gamma D_{KL}(\mathbf{u}, \mathbf{g}) + \text{TV}(\mathbf{f})$, otherwise we have $q(\mathbf{u}, \mathbf{f}) = \infty$. Therefore,

$$\min_{\mathbf{f} \in S, \mathbf{u}} q(\mathbf{u}, \mathbf{f}) = \min_{\mathbf{u} = H\mathbf{f} + \mathbf{b}, \mathbf{f} \in S} (\gamma D_{KL}(\mathbf{u}, \mathbf{g}) + \text{TV}(\mathbf{f})) = \min_{\mathbf{f} \in S} \gamma \Psi(\mathbf{f}).$$

□

For the function \mathcal{L} , we will show that there exists a point $(\mathbf{u}^*, \mathbf{f}^*, \mathbf{y}^*, \mathbf{p}^*)$ that satisfies the following inequalities

$$\mathcal{L}(\mathbf{u}^*, \mathbf{f}^*, \mathbf{y}, \mathbf{p}) \leq \mathcal{L}(\mathbf{u}^*, \mathbf{f}^*, \mathbf{y}^*, \mathbf{p}^*) \leq \mathcal{L}(\mathbf{u}, \mathbf{f}, \mathbf{y}^*, \mathbf{p}^*), \quad \forall \mathbf{f}, \mathbf{u}, \mathbf{y} \text{ and } \mathbf{p} \in A,$$

i.e. $(\mathbf{u}^*, \mathbf{f}^*, \mathbf{y}^*, \mathbf{p}^*)$ is a saddle point of \mathcal{L} . We first need the following two lemmas.

Lemma 3.2. *Assume that the entries of the matrix H are nonnegative and $H\mathbf{1} = \mathbf{1}$. Then the function $\Psi(\mathbf{f})$ is proper.*

Proof. We recall that a function $\phi(\mathbf{f}) : F \mapsto [-\infty, \infty]$ is called proper if $\phi(\mathbf{f}) < \infty$ for at least one $\mathbf{f} \in F$ and $\phi(\mathbf{f}) > -\infty$ for all $\mathbf{f} \in F$. Since the entries of H are nonnegative, it is easy to see that for any given $\mathbf{f} > 0$, we have $D_{KL}(H\mathbf{f}, \mathbf{g}) < \infty$ and for any \mathbf{f} , $D_{KL}(H\mathbf{f}, \mathbf{g}) > -\infty$. Therefore the function $D_{KL}(H\mathbf{f}, \mathbf{g})$ is proper. Also, according to the definition of the TV norm, the function $\text{TV}(\mathbf{f})$ is also proper. Therefore the sum of $D_{KL}(H\mathbf{f}, \mathbf{g})$ and $\text{TV}(\mathbf{f})$ is proper. □

Because of the physics underlying the image formation process, the point spread function takes on nonnegative values only. Also since there is no energy absorbed and generated in the process, the sum of all entries of the point spread function is equal to 1, see [7]. Consequently, the entries of H are nonnegative and $H\mathbf{1} = \mathbf{1}$. Hence the assumption in Lemma 3.2 is reasonable.

Notice that the function $D_{KL}(\mathbf{f}, \mathbf{g})$ is coercive, i.e., $D_{KL}(\mathbf{f}, \mathbf{g}) \rightarrow \infty$ if $\|\mathbf{f}\|_2 \rightarrow \infty$; therefore the functions $\Psi(\mathbf{f})$ and $q(\mathbf{u}, \mathbf{f})$ are coercive. By the coercivity of $q(\mathbf{u}, \mathbf{f})$, it is obvious that its nonempty lower level sets are bounded. Hence the level sets $\{(\mathbf{f}; \mathbf{u}) \mid q(\mathbf{u}, \mathbf{f}) \leq \rho\}$ are compacts. Thus we have the following lemma.

Lemma 3.3. *The functions $\Psi(\mathbf{f})$ is coercive, i.e., $\Psi(\mathbf{f}) \rightarrow \infty$ if $\|\mathbf{f}\|_2 \rightarrow \infty$. Moreover, $q(\mathbf{u}, \mathbf{f})$ is also coercive and the level sets $\{(\mathbf{f}; \mathbf{u}) \mid q(\mathbf{u}, \mathbf{f}) \leq \rho\}$ are compacts.*

According to [5, Proposition 5.5.4], we know that the minimum and the maximum in (3.6) can be swapped and there exists a saddle point of \mathcal{L} . Thus we have the following theorem.

Theorem 3.4. *Assume that the entries of the matrix H are nonnegative and $H\mathbf{1} = \mathbf{1}$. We have*

$$\min_{\mathbf{f}, \mathbf{u}} \max_{\mathbf{p} \in A, \mathbf{y}} \mathcal{L}(\mathbf{u}, \mathbf{f}, \mathbf{y}, \mathbf{p}) = \max_{\mathbf{p} \in A, \mathbf{y}} \min_{\mathbf{f}, \mathbf{u}} \mathcal{L}(\mathbf{u}, \mathbf{f}, \mathbf{y}, \mathbf{p})$$

and the minimum in the left-hand side above is attained at a set of points that is nonempty and compact.

In order to apply Chambolle-Pock's algorithm to (3.6), we need to formulate (3.6) in the form of (3.1). Denote

$$\mathbf{x} = \begin{pmatrix} \mathbf{u} \\ \mathbf{f} \end{pmatrix}, \quad \mathbf{z} = \begin{pmatrix} \mathbf{y} \\ \mathbf{p} \end{pmatrix}, \quad K = \begin{pmatrix} I & -H \\ 0 & -\nabla \end{pmatrix}$$

$\phi(\mathbf{x}) = \gamma D_{KL}(\mathbf{u}, \mathbf{g})$, and $\psi(\mathbf{y}) = \langle \mathbf{y}, \mathbf{b} \rangle$. We see that (3.6) completely fits into the framework of (3.1). Hence (3.2) can readily be used to solve (3.6). We note that in (3.5), \mathbf{u} can be separable from \mathbf{f} and \mathbf{y} can be separable from \mathbf{p} . Therefore (3.2) can be rewritten as follows.

Algorithm 1. Starting from an initial guess $(\mathbf{u}^{(0)}, \mathbf{f}^{(0)}, \mathbf{y}^{(0)}, \mathbf{p}^{(0)})$, compute:

$$\mathbf{u}^{(k+1)} = \underset{\mathbf{u}}{\operatorname{argmin}} \mathcal{L}(\mathbf{u}, \mathbf{f}, \mathbf{y}^{(k)}, \mathbf{p}^{(k)}) + \frac{1}{2s} \left\| \mathbf{u} - \mathbf{u}^{(k)} \right\|_2^2, \quad (3.7)$$

$$\mathbf{f}^{(k+1)} = \underset{\mathbf{f}}{\operatorname{argmin}} \mathcal{L}(\mathbf{u}, \mathbf{f}, \mathbf{y}^{(k)}, \mathbf{p}^{(k)}) + \frac{1}{2s} \left\| \mathbf{f} - \mathbf{f}^{(k)} \right\|_2^2, \quad (3.8)$$

$$\widehat{\mathbf{u}}^{(k+1)} = \mathbf{u}^{(k+1)} + \theta(\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}), \quad (3.9)$$

$$\widehat{\mathbf{f}}^{(k+1)} = \mathbf{f}^{(k+1)} + \theta(\mathbf{f}^{(k+1)} - \mathbf{f}^{(k)}), \quad (3.10)$$

$$\mathbf{y}^{(k+1)} = \underset{\mathbf{y}}{\operatorname{argmax}} \mathcal{L}(\widehat{\mathbf{u}}^{(k+1)}, \widehat{\mathbf{f}}^{(k+1)}, \mathbf{y}, \mathbf{p}) - \frac{1}{2t} \left\| \mathbf{y} - \mathbf{y}^{(k)} \right\|_2^2, \quad (3.11)$$

$$\mathbf{p}^{(k+1)} = \underset{\mathbf{p} \in A}{\operatorname{argmax}} \mathcal{L}(\widehat{\mathbf{u}}^{(k+1)}, \widehat{\mathbf{f}}^{(k+1)}, \mathbf{y}, \mathbf{p}) - \frac{1}{2t} \left\| \mathbf{p} - \mathbf{p}^{(k)} \right\|_2^2. \quad (3.12)$$

Note that besides the storage for H , \mathbf{f} and \mathbf{g} , we need storage for the auxiliary variables $\mathbf{u} \in \mathbb{R}^{mn}$, $\mathbf{y} \in \mathbb{R}^{mn}$ and $\mathbf{p} \in \mathbb{R}^{2mn}$. We need not store $\widehat{\mathbf{u}}$ and $\widehat{\mathbf{f}}$, as we can use the right hand side of (3.9) and (3.10) to generate them every time they are required. Thus the storage for auxiliary variables is $O(4mn)$. We now discuss how each minimization problem in (3.7)–(3.12) is solved.

3.2.1 Subproblem for \mathbf{u}

The problem in (3.7) can be written as

$$\begin{aligned} \mathbf{u}^{(k+1)} &= \underset{\mathbf{u}}{\operatorname{argmin}} \gamma D_{KL}(\mathbf{u}, \mathbf{g}) + \langle \mathbf{y}^{(k)}, \mathbf{u} \rangle + \frac{1}{2s} \left\| \mathbf{u} - \mathbf{u}^{(k)} \right\|_2^2 \\ &= \underset{\mathbf{u}}{\operatorname{argmin}} \gamma (\langle \mathbf{1}, \mathbf{u} \rangle - \langle \mathbf{g}, \ln \mathbf{u} \rangle) + \frac{1}{2s} \left\| \mathbf{u} - \boldsymbol{\zeta}^{(k)} \right\|_2^2, \end{aligned}$$

where $\boldsymbol{\zeta}^{(k)} = \mathbf{u}^{(k)} - s\mathbf{y}^{(k)}$. Its optimality condition is

$$\gamma \left(1 - \frac{\mathbf{g}}{\mathbf{u}} \right) + \frac{1}{s} (\mathbf{u} - \boldsymbol{\zeta}^{(k)}) = 0.$$

It can be turned into

$$\mathbf{u}^2 - (\boldsymbol{\zeta}^{(k)} - s\gamma) \mathbf{u} - s\gamma \mathbf{g} = 0.$$

Notice that \mathbf{u} should be positive, thus we have

$$\mathbf{u}^{(k+1)} = \frac{1}{2} \left((\boldsymbol{\zeta}^{(k)} - s\gamma) + \sqrt{(\boldsymbol{\zeta}^{(k)} - s\gamma)^2 + 4s\gamma \mathbf{g}} \right). \quad (3.13)$$

3.2.2 Subproblem for \mathbf{f}

We see that the objective function in (3.8) for \mathbf{f} is quadratic and is restricted onto the constrained set S . Hence, $\mathbf{f}^{(k+1)}$ can be easily computed by

$$\mathbf{f}^{(k+1)} = \mathcal{P}_S \left(\mathbf{f}^{(k)} + s(H^T \mathbf{y}^{(k)} - \operatorname{div} \mathbf{p}^{(k)}) \right).$$

The projection of a vector \mathbf{x} onto S can be conveniently expressed as $\mathcal{P}_S(\mathbf{u}) = \operatorname{argmin}_{\mathbf{x} \in S} \|\mathbf{x} - \mathbf{u}\|_2^2$. From the definition of S in (1.5), it is easy to check that

$$(\mathcal{P}_S(\mathbf{u}))_{i,j} = \begin{cases} \mathbf{u}_{i,j}, & \text{if } 0 \leq \mathbf{u}_{i,j} \leq M, \\ 0, & \text{if } \mathbf{u}_{i,j} < 0, \\ M, & \text{if } \mathbf{u}_{i,j} > M. \end{cases}$$

3.2.3 Subproblem for \mathbf{y}

The maximization of (3.11) reduces to

$$\max_{\mathbf{y}} \left\langle \mathbf{y}, \widehat{\mathbf{u}}^{(k+1)} - H\widehat{\mathbf{f}}^{(k+1)} - \mathbf{b} \right\rangle - \frac{1}{2t} \left\| \mathbf{y} - \mathbf{y}^{(k)} \right\|_2^2.$$

Hence

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + t \left(\widehat{\mathbf{u}}^{(k+1)} - H\widehat{\mathbf{f}}^{(k+1)} - \mathbf{b} \right).$$

3.2.4 Subproblem for the dual variable \mathbf{p}

We change the maximization problem for \mathbf{p} in (3.12) to a minimization one and obtain:

$$\begin{aligned}\mathbf{p}^{(k+1)} &= \arg \min_{\mathbf{p} \in A} \left\{ \langle \mathbf{p}, \nabla \widehat{\mathbf{f}}^{(k+1)} \rangle_Y + \frac{1}{2t} \|\mathbf{p} - \mathbf{p}^{(k)}\|_2^2 \right\} \\ &= \arg \min_{\mathbf{p} \in A} \|\mathbf{p} - (\mathbf{p}^{(k)} - t \nabla \widehat{\mathbf{f}}^{(k+1)})\|_2^2.\end{aligned}$$

Thus $\mathbf{p}^{(k+1)} = \mathcal{P}_A(\mathbf{p}^{(k)} - t \nabla \widehat{\mathbf{f}}^{(k+1)})$, the projection of $(\mathbf{p}^{(k)} - t \nabla \widehat{\mathbf{f}}^{(k+1)})$ onto the set A . In the following, we derive a formula for the projection operator $\mathcal{P}_A(\mathbf{q}) \equiv \arg \min_{\mathbf{p} \in A} \|\mathbf{p} - \mathbf{q}\|_2^2$ for any vector \mathbf{q} . By the definition of A in (3.3), its Lagrangian function is

$$\|\mathbf{p} - \mathbf{q}\|_2^2 + \sum_{i,j} \beta_{i,j} (|p_{i,j}|^2 - 1),$$

where $\beta_{i,j} \geq 0$ are the Lagrangian multipliers associated with the constraints $|p_{i,j}|^2 \leq 1$. Its complementarity condition implies that for the optimal $\beta_{i,j}$, either $\beta_{i,j} = 0$ with $|p_{i,j}|, |q_{i,j}| < 1$, or $\beta_{i,j} > 0$ with $|p_{i,j}| = 1$ and $|q_{i,j}| \geq 1$. In the former case, we have $p_{i,j} = q_{i,j}$. In the latter case, the KKT condition yields

$$p_{i,j} - q_{i,j} + \beta_{i,j} p_{i,j} = 0, \quad \forall i, j.$$

Therefore, we have $\beta_{i,j} = |q_{i,j}| - 1$, and thus $p_{i,j} = q_{i,j}/|q_{i,j}|$. Hence, we obtain

$$(\mathcal{P}_A(\mathbf{q}))_{i,j} = \frac{q_{i,j}}{\max(1, |q_{i,j}|)}, \quad \forall i, j.$$

3.3 Second Minimax Problem

Here we convert the Poisson image restoration problem (1.4) into another minimax problem. Then we apply Chambolle-Pock's algorithm (3.2) to solve it. According to Theorem 3.4, the minimum and the maximum can be swapped in (3.6). If we define

$$\mathcal{L}_2(\mathbf{f}, \mathbf{y}, \mathbf{p}) = \min_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \mathbf{f}, \mathbf{y}, \mathbf{p}).$$

By $\frac{\partial \mathcal{L}}{\partial \mathbf{u}} = 0$, we obtain $\mathbf{u} = \frac{\mathbf{g}}{1+\mathbf{y}}$. We can easily check that

$$\mathcal{L}_2(\mathbf{f}, \mathbf{y}, \mathbf{p}) = \langle \mathbf{g}, \ln(1 + \mathbf{y}) \rangle - \langle \mathbf{y}, H\mathbf{f} + \mathbf{b} \rangle + \lambda \langle \mathbf{f}, \operatorname{div} \mathbf{p} \rangle + \delta(\mathbf{f}) + C.$$

Here C is a constant which is independent to the variables $\mathbf{f}, \mathbf{y}, \mathbf{p}$. We note that $\mathcal{L}_2(\mathbf{f}, \cdot, \cdot)$ and $\mathcal{L}_2(\cdot, \cdot, \mathbf{p})$ are linear functions with respect to the variable \mathbf{f} and \mathbf{p} respectively. Hence they are both convex and concave. Next, we show that the function $\mathcal{L}_2(\cdot, \mathbf{y}, \cdot)$ is concave with respect to \mathbf{y} . Note that $\mathcal{L}_2(\cdot, \mathbf{y}, \cdot)$ is separable, which implies that the Hessian matrix of $\mathcal{L}_2(\cdot, \mathbf{y}, \cdot)$ is diagonal. We have

$$\frac{\partial \mathcal{L}_2}{\partial \mathbf{y}_i} = \frac{\mathbf{g}_i}{1 + \mathbf{y}_i} - (H\mathbf{f} + \mathbf{b})_i$$

and

$$\frac{\partial^2 \mathcal{L}_2}{\partial \mathbf{y}_i^2} = -\frac{\mathbf{g}_i}{(1 + \mathbf{y}_i)^2}.$$

Since $\mathbf{g}_i > 0$, the Hessian matrix of $\mathcal{L}_2(\cdot, \mathbf{y}, \cdot)$ with respect to \mathbf{y} is negative definite. Hence $\mathcal{L}_2(\cdot, \mathbf{y}, \cdot)$ is concave with respect to \mathbf{y} . Hence (1.4) can be reformulated as the following maximin problem:

$$\max_{\mathbf{p} \in A, \mathbf{y}} \min_{\mathbf{f}} \mathcal{L}_2(\mathbf{f}, \mathbf{y}, \mathbf{p}). \quad (3.14)$$

We state this clearly in the next theorem.

Theorem 3.5. *Assume that the entries of the matrix H are nonnegative and $H\mathbf{1} = \mathbf{1}$. We have*

$$\min_{\mathbf{f}} \Psi(\mathbf{f}) = \min_{\mathbf{f}} \max_{\mathbf{p} \in A, \mathbf{y}} \mathcal{L}_2(\mathbf{f}, \mathbf{y}, \mathbf{p}) = \max_{\mathbf{p} \in A, \mathbf{y}} \min_{\mathbf{f}} \mathcal{L}_2(\mathbf{f}, \mathbf{y}, \mathbf{p}).$$

and the minimum in the left-hand side above is attained at a set of points that is nonempty and compact.

Proof. It is easy to check that

$$\min_{\mathbf{f}} \max_{\mathbf{p} \in A, \mathbf{y}} \mathcal{L}_2(\mathbf{f}, \mathbf{y}, \mathbf{p}) = \min_{\mathbf{f} \in S} \bar{\Psi}(\mathbf{f}).$$

Applying [5, Proposition 5.5.4], we know the theorem holds. \square

According to the above Theorem, we can apply Chambolle-Pock's first-order primal-dual method (3.2) to solve the minimax version of (3.14) and get the following algorithm.

Algorithm 2: Starting from an initial guess $(\mathbf{f}^{(0)}, \mathbf{y}^{(0)}, \mathbf{p}^{(0)})$, compute:

$$\mathbf{f}^{(k+1)} = \underset{\mathbf{f}}{\operatorname{argmin}} \mathcal{L}_2(\mathbf{f}, \mathbf{y}^{(k)}, \mathbf{p}^{(k)}) + \frac{1}{2s} \left\| \mathbf{f} - \mathbf{f}^{(k)} \right\|_2^2, \quad (3.15)$$

$$\widehat{\mathbf{f}}^{(k+1)} = \mathbf{f}^{(k+1)} + \theta(\mathbf{f}^{(k+1)} - \mathbf{f}^{(k)}), \quad (3.16)$$

$$\mathbf{y}^{(k+1)} = \underset{\mathbf{y}}{\operatorname{argmax}} \mathcal{L}_2(\widehat{\mathbf{f}}^{(k+1)}, \mathbf{y}, \mathbf{p}) - \frac{1}{2t} \left\| \mathbf{y} - \mathbf{y}^{(k)} \right\|_2^2, \quad (3.17)$$

$$\mathbf{p}^{(k+1)} = \underset{\mathbf{p} \in A}{\operatorname{argmax}} \mathcal{L}_2(\widehat{\mathbf{f}}^{(k+1)}, \mathbf{y}, \mathbf{p}) - \frac{1}{2t} \left\| \mathbf{p} - \mathbf{p}^{(k)} \right\|_2^2. \quad (3.18)$$

Clear the algorithm requires $O(3mn)$ storage for the auxiliary variables $\mathbf{y} \in \mathbb{R}^{mn}$ and $\mathbf{p} \in \mathbb{R}^{2mn}$ (the variable $\widehat{\mathbf{f}}$ can be generated by the right hand side of (3.16) every time it is needed). It is easy to see that the minimization subproblem for the primal variable \mathbf{f} in (3.15) and the maximization subproblem for the dual variable \mathbf{p} in (3.18) are exactly the ones in (3.8) and (3.12), respectively. Thus we only need to consider the solution of the subproblem for the variable \mathbf{y} in (3.17). Its optimality condition is given by

$$\frac{\mathbf{g}}{1 + \mathbf{y}} - (H\widehat{\mathbf{f}}^{(k+1)} + \mathbf{b}) - \frac{1}{t} (\mathbf{y} - \mathbf{y}^{(k)}) = 0.$$

Denote $\boldsymbol{\beta}^{(k)} = \mathbf{y}^{(k)} - t(H\widehat{\mathbf{f}}^{(k+1)} + \mathbf{b})$. The above equation can be rewritten as

$$\mathbf{y}^2 - (\boldsymbol{\beta}^{(k)} - 1) \mathbf{y} - \boldsymbol{\beta}^{(k)} - t\mathbf{g} = 0.$$

We thus obtain the solution

$$\mathbf{y}^{(k+1)} = \frac{1}{2} \left(\boldsymbol{\beta}^{(k)} - 1 + \sqrt{(\boldsymbol{\beta}^{(k)} - 1)^2 + 4(\boldsymbol{\beta}^{(k)} + t\mathbf{g})} \right).$$

We close the section by remarking that both Algorithms 1 and 2 are realizations of Chambolle-Pock's algorithm for different minimax problems. Thus they enjoy the advantages of Chambolle-Pock's algorithm. Namely, they will converge with rate $O(1/k)$ when $\theta = 1$ and the step sizes satisfy $st < 1/\|K^T K\|$; and they are matrix inversion-free. Besides needing less storage than ADMM-type methods, we will see in the next section that they require less CPU time than ADMM-type methods too.

4 Numerical Results

We now illustrate the performance of our proposed algorithms for image restoration problem with Poisson noise. We compare our algorithms with ADMM-type methods in [21] and [35], see (2.4) and (2.5) respectively. For simplicity, we call them "ADMM-1" and "ADMM-2" respectively. Our codes are written in Matlab R2013a. The experiments were performed under Mac OS X10.9.4 on an iMac with a 3.4GHz Intel Core i5 processor and 8GB of RAM. The Signal-to-Noise Ratio (SNR) is used to measure

the quality of the restoration and is defined as: $\text{SNR} = 10 \log_{10} (\|\mathbf{f}^*\|_2^2 / \|\mathbf{f} - \mathbf{f}^*\|_2^2)$, where \mathbf{f}^* and \mathbf{f} are the original image and the restored image, respectively. We use the observed image \mathbf{g} as initial image for the algorithms.

The Matlab command `fspecial('gaussian',9,2)` is used to generate the Gaussian blur used in the experiment. We also tested other point spread functions such as uniform blurs and similar results are obtained; therefore they will not be reported here. Since Poisson noise is a data-dependent noise, the noise level of the observed images depends on the pixel intensity. To test different noise levels, we consider different peak intensities of the images with $M = 15, 30, 45, 60$, see (1.5) for the definition of M . To obtain the degraded image \mathbf{g} , we follow the model in (1.1). More precisely, the original image \mathbf{f}^* is first scaled with the peak intensities. Then the scaled image is convolved with the blur kernel H and then background value is added. Three background values with $b = 1, 5, 10$ are tested. Finally, the Poisson noise is generated to the blurred image by using the Matlab command `poissrnd`. In all tests, the regularization parameter $\gamma = 1/\lambda$ is chosen to satisfy the constraint $D_{KL}(H\mathbf{f}(\gamma), \mathbf{g}) = nm/2$, see [44].

In theory and in practice, the difference between successive iterates of the variable \mathbf{f} , i.e., $\|\mathbf{f}^{(k+1)} - \mathbf{f}^{(k)}\|_2^2$ is widely used to measure the convergence, see [25]. In our experiments, we apply the Relative Difference

$$\text{RD} = \frac{\|\mathbf{f}^{(k+1)} - \mathbf{f}^{(k)}\|_2^2}{\|\mathbf{f}^{(k+1)}\|_2^2} \quad (4.1)$$

as the measurement of convergence. We set the stopping criterion as $\text{RD} \leq 10^{-4}$ or when the iteration has reached 1000 iterations.

4.1 Gray-scale Images

We begin with gray-scale images. The test images are the cameraman image, the satellite image, and the MRI image, all with size 256×256 . The original images and the observed images with $M = 15, b = 1$ are shown in the left and middle columns of Figure 1 respectively.

As suggested in [21, 35], the parameter α in the ADMM is set to $50/\lambda$, and Chambolle's projection algorithm [11] is applied to solve the TV-denoising subproblem in ADMM-1. The number of iterations of Chambolle's algorithm is set to 20 and the final values for the dual variable of an inner iteration loop is used as the initial values for the next loop. We used the circulant boundary condition for H and ∇ , hence the matrix-inversions in ADMM can be done easily by fast fourier transform.

The SNR (in dB) and the CPU times (in seconds) of the algorithms are listed in Table 1, 3, 5. In theory, the SNR obtained by the algorithms should be the same since they are solving the same minimization problem. However, we find that there are slight differences due to the stopping criterion or computational errors. The analysis for the computation errors for different algorithms is still an open problem and is beyond the scope of this paper. Since there are only slight differences in the SNR of the restored images, we show only the restored images by Algorithm 1 in the right column of Figure 1. From the tables, we note however that our proposed algorithms out-perform the ADMM methods significantly in CPU times. We also list the structure similarity (SSIM) [45] of the algorithms in Table 2, 4, 6. The SSIM values are very similar.

Next we compare the convergence of the algorithms. The plots of RD (defined in (4.1)) versus CPU times are shown in Figure 2. The green solid line, the black dash-dot line, the blue dotted line, and the red dashed line represent Algorithm 1, Algorithm 2, ADMM-1 and ADMM-2 respectively. Note that the green solid line and the black dash-dot lines are almost indistinguishable from one another, indicating that the convergence of Algorithms 1 and 2 are almost the same. We observe that Algorithms 1 and 2 converge much faster than the two ADMM methods, and their convergence are very stable.

Finally, in Figure 3, we plot the SNR versus CPU times for the algorithms. We observe that the SNR obtained by the ADMM methods may stagnate after 2 seconds for the three images. Hence in the figure, we only show the results of the first 5 seconds. We observe that our proposed algorithms converge to a better SNR faster than the ADMM methods.

M	b	$\lambda = 1/\gamma$	SNR (dB)				CPU Time (s)			
			Alg.1	Alg.2	ADMM-1	ADMM-2	Alg.1	Alg.2	ADMM-1	ADMM-2
15	1	1.53e-01	16.64	16.64	16.62	16.59	3.02	2.89	18.12	13.24
30		1.25e-01	16.96	16.96	16.97	16.93	3.21	2.95	14.65	8.43
45		1.01e-01	17.25	17.25	17.29	17.23	3.32	3.11	13.43	7.58
60		7.16e-02	17.36	17.37	17.53	17.51	3.54	3.42	8.46	6.37
15	5	9.71e-02	16.24	16.23	16.13	16.20	3.34	3.12	12.47	13.71
30		1.00e-01	16.96	16.96	16.92	16.84	3.76	3.24	9.15	8.91
45		1.07e-01	17.07	17.05	17.15	17.11	4.06	3.16	7.20	6.60
60		7.68e-02	17.26	17.25	17.37	17.35	4.35	3.49	6.82	5.85
15	10	9.98e-02	15.90	15.91	15.87	15.85	3.99	3.29	11.01	12.98
30		6.75e-02	16.67	16.65	16.64	16.64	4.62	3.42	9.76	8.82
45		7.12e-02	17.09	17.07	17.11	17.07	5.12	3.59	7.70	6.78
60		6.23e-02	17.18	17.19	17.32	17.30	5.36	3.90	6.63	5.55
Average			16.88	16.88	16.91	16.88	3.97	3.30	10.45	8.73

Table 1. SNR and CPU time results for the cameraman image under Gaussian blur. The SNR values are similar. The best CPU times are in black.

M	b	Alg.1	Alg.2	ADMM-1	ADMM-2
15	1	0.9793	0.9793	0.9790	0.9792
30		0.9510	0.9510	0.9507	0.9509
45		0.9243	0.9244	0.9244	0.9252
60		0.9002	0.9005	0.9027	0.9026
15	5	0.9762	0.9762	0.9757	0.9760
30		0.9482	0.9482	0.9481	0.9483
45		0.9212	0.9211	0.9219	0.9222
60		0.8972	0.8970	0.8994	0.9000
15	10	0.9743	0.9744	0.9741	0.9743
30		0.9468	0.9466	0.9467	0.9468
45		0.9191	0.9187	0.9200	0.9203
60		0.8966	0.8959	0.8989	0.8989
Average		0.9362	0.9361	0.9368	0.9371

Table 2 SSIM results for the cameraman image under Gaussian blur.

M	b	$\lambda = 1/\gamma$	SNR (dB)				CPU Time (s)			
			Alg.1	Alg.2	ADMM-1	ADMM-2	Alg.1	Alg.2	ADMM-1	ADMM-2
15	1	1.65e-01	12.52	12.52	12.46	12.52	3.09	3.11	32.74	23.88
30		1.43e-01	12.96	12.96	12.88	12.90	3.55	3.39	32.09	15.00
45		1.32e-01	13.08	13.08	13.04	13.05	3.93	3.78	14.71	7.11
60		1.23e-01	13.24	13.24	13.20	13.21	4.11	4.25	12.32	5.13
15	5	9.51e-02	12.33	12.33	12.24	12.31	3.97	3.86	31.75	20.01
30		8.98e-02	12.88	12.88	12.83	12.84	3.94	3.71	22.31	13.80
45		7.81e-02	13.12	13.12	13.08	13.12	4.38	3.97	18.52	11.21
60		7.31e-02	13.27	13.28	13.26	13.27	4.49	4.19	10.93	8.49
15	10	7.33e-02	11.86	11.87	11.77	11.84	4.97	4.20	30.18	18.32
30		5.93e-02	12.73	12.75	12.63	12.67	4.83	4.35	19.23	12.36
45		5.39e-02	13.07	13.06	12.99	13.00	4.99	4.20	17.22	11.07
60		5.53e-02	13.23	13.22	13.25	13.28	5.70	4.41	15.55	8.68
Average			12.86	12.86	12.80	12.83	4.33	3.95	21.46	12.92

Table 3. SNR and CPU time results for the satellite image under Gaussian blur. The SNR values are similar. The best CPU times are in black.

M	b	Alg.1	Alg.2	ADMM-1	ADMM-2
15	1	0.9810	0.9810	0.9810	0.9811
30		0.9563	0.9563	0.9562	0.9567
45		0.9360	0.9360	0.9370	0.9368
60		0.9198	0.9198	0.9204	0.9207
15	5	0.9769	0.9769	0.9763	0.9764
30		0.9518	0.9519	0.9511	0.9516
45		0.9324	0.9325	0.9327	0.9327
60		0.9169	0.9171	0.9202	0.9180
15	19	0.9727	0.9728	0.9722	0.9722
30		0.9462	0.9463	0.9458	0.9461
45		0.9269	0.9269	0.9269	0.9270
60		0.9121	0.9122	0.9134	0.9131
Average		0.9441	0.9441	0.9444	0.9444

Table 4 SSIM results for the satellite image under Gaussian blur.

M	b	$\lambda = 1/\gamma$	SNR (dB)				CPU Time (s)			
			Alg.1	Alg.2	ADMM-1	ADMM-2	Alg.1	Alg.2	ADMM-1	ADMM-2
15	1	1.57e-01	13.11	13.11	13.10	13.10	2.71	2.56	30.83	23.05
30		1.42e-01	13.60	13.59	13.56	13.53	2.98	2.82	14.06	14.92
45		1.25e-01	13.87	13.86	13.74	13.75	3.40	3.29	12.30	6.54
60		1.19e-01	14.03	14.02	13.87	13.87	3.55	3.42	10.33	5.79
15	5	8.87e-02	12.75	12.76	12.71	12.75	3.38	3.13	30.69	21.90
30		9.00e-02	13.60	13.60	13.49	13.49	3.74	3.39	19.75	14.23
45		7.74e-02	13.94	13.94	13.81	13.78	4.22	3.83	15.55	10.76
60		6.94e-02	14.26	14.26	14.07	14.05	4.21	3.77	9.11	8.52
15	10	8.15e-02	12.33	12.34	12.30	12.35	4.06	3.54	31.80	17.32
30		5.94e-02	13.45	13.48	13.36	13.35	4.75	4.10	18.17	12.61
45		5.48e-02	13.93	13.95	13.82	13.80	5.07	4.14	15.80	9.93
60		5.08e-02	14.23	14.25	14.05	14.05	5.44	4.29	13.22	8.93
Average			13.59	13.60	13.49	13.49	3.96	3.52	18.47	12.88

Table 5. SNR and CPU time results for the MRI image under Gaussian blur. The SNR values are similar. The best CPU times are in black.

M	b	Alg.1	Alg.2	ADMM-1	ADMM-2
15	1	0.9830	0.9830	0.9831	0.9831
30		0.9580	0.9580	0.9585	0.9578
45		0.9354	0.9353	0.9353	0.9352
60		0.9114	0.9113	0.9103	0.9102
15	5	0.9788	0.9788	0.9784	0.9786
30		0.9546	0.9547	0.9536	0.9538
45		0.9324	0.9325	0.9334	0.9311
60		0.9120	0.9121	0.9119	0.9102
15	10	0.9750	0.9751	0.9747	0.9748
30		0.9517	0.9519	0.9505	0.9508
45		0.9270	0.9273	0.9255	0.9255
60		0.9084	0.9087	0.9073	0.9061
Average		0.9440	0.9441	0.9435	0.9431

Table 6 SSIM results for the MRI image under Gaussian blur.

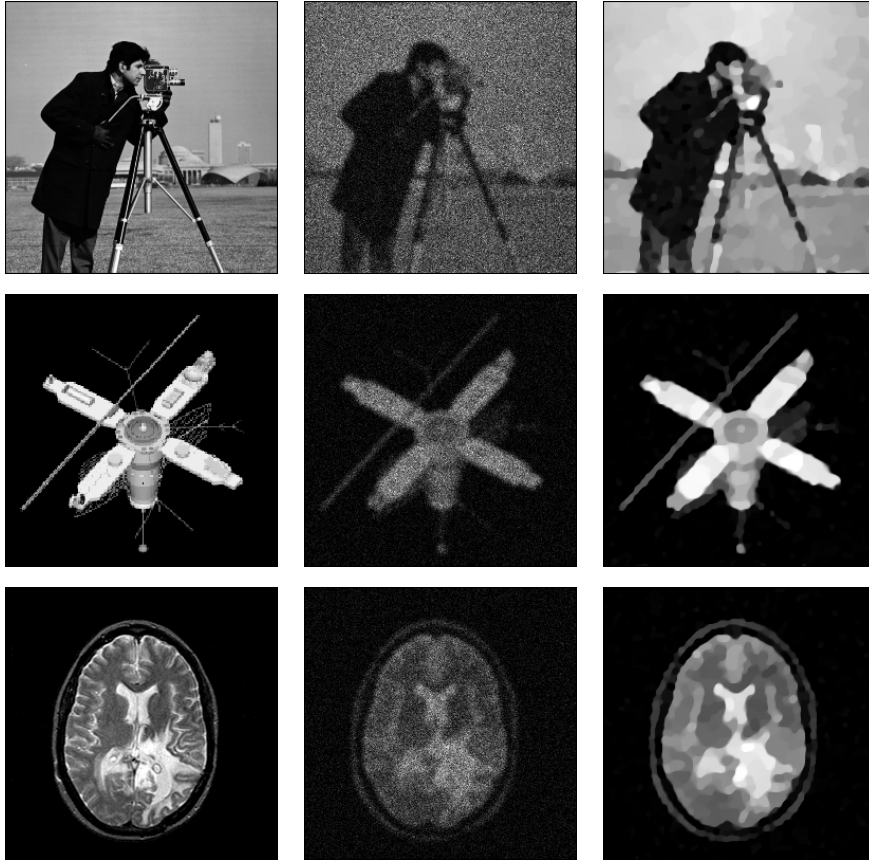


Figure 1. Original images (left), the degraded images with $M = 15, b = 1$ (middle), and the restored images by Algorithm 1 (right).

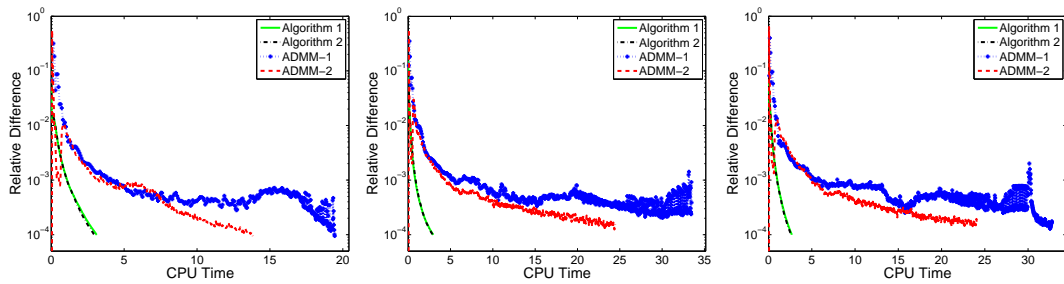


Figure 2. Relative difference (RD) versus CPU time for the cameraman image, the satellite image and the MRI image (from left to right) with $M = 15, b = 1$.

4.2 Color Images

Next we extend our proposed algorithms to color images. The extension can easily be performed to each color component independently when no cross-channel degradation is presented. However, such a process produces false colors in the restored images since the signals in different channels are often correlated and there exist spaces of perceptual redundancy in color images. One solution to overcome this is to replace the gray-scale TV defined in (1.3) by the color TV proposed by Blomgren and Chan [6]. We tested this extension of color TV with the onion image of size 135×198 and the cells image of size 512×512 . The original images, the degraded images with $M = 30, b = 1$ and the restored images by Algorithm 1 are shown in Figure 4.

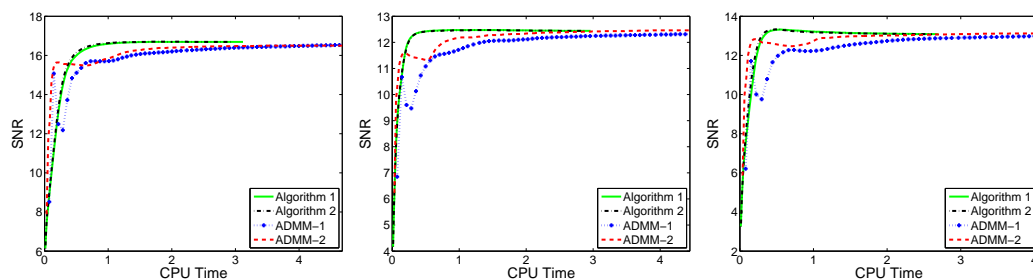


Figure 3. SNR versus CPU time for the cameraman image, the satellite image and the MRI image (from left to right) with $M = 15, b = 1$.

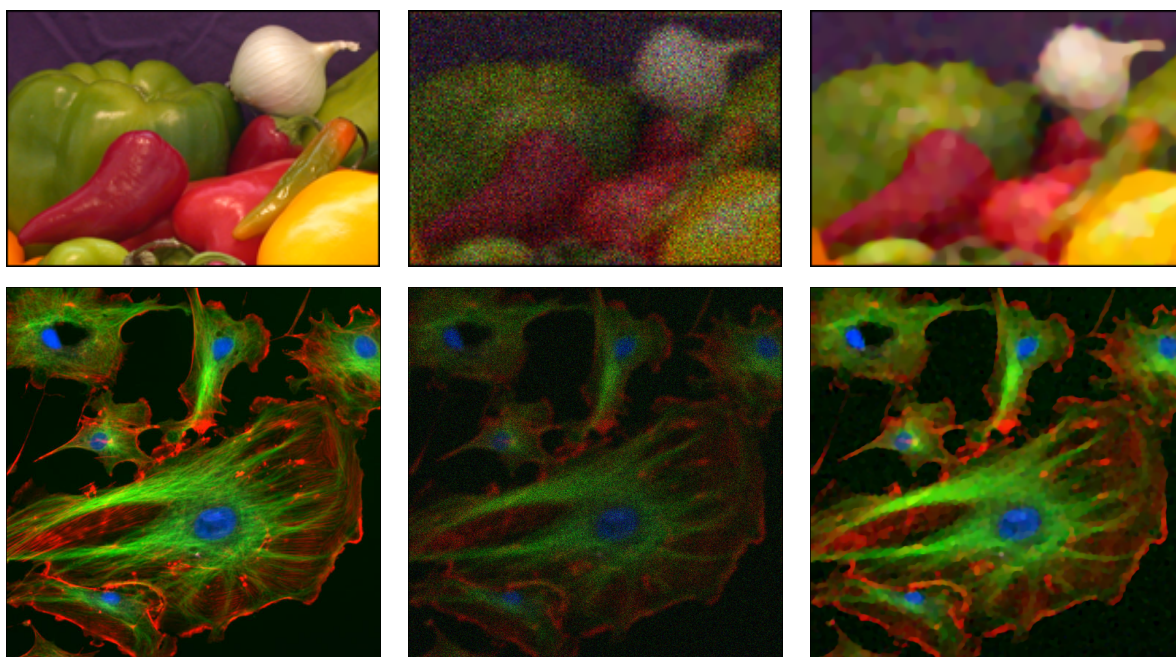


Figure 4. Original images (left), degraded images with $M = 30, b = 1$ (middle) and the restored images by Algorithm 1 (right).

The SNR and CPU times of the algorithms for three background values ($b = 1, 5, 10$) are listed in Table 7 and 9. The SSIM values are listed in Table 8 and 10. The convergence history (RD versus CPU time) for difference algorithms are shown in Figure 5. Once again, we observe that our proposed algorithms out-perform the ADMM methods significantly in CPU times and the convergence is very stable.

5 Conclusion

We have proposed two algorithms for solving image restoration problem with Poisson noise. We reformulated the problem into two minimax problems and then applied Chambolle-Pock's first-order primal-dual algorithm to compute the saddle point of the minimax problems. Compared to existing ADMM algorithms, our algorithms require less memory and no need to solve any linear systems. Numerical results show that our algorithms are stable and fast.

References

- 1 H. Andrew and B. Hunt. *Digital Image Restoration*. Prentice-Hall, Englewood Cliffs, NJ, 1977.

M	b	$\lambda = 1/\gamma$	SNR (dB)				CPU Time (s)			
			Alg.1	Alg.2	ADMM-1	ADMM-2	Alg.1	Alg.2	ADMM-1	ADMM-2
15	1	1.59e-01	19.33	19.34	18.62	18.17	3.88	3.82	28.66	24.82
30		1.50e-01	20.37	20.37	20.27	19.00	4.66	4.44	17.20	16.54
45		1.65e-01	20.61	20.61	21.01	19.38	5.04	4.76	11.74	12.37
60		1.40e-01	20.65	20.65	21.12	19.65	5.51	5.46	9.52	10.07
15	5	1.54e-01	18.67	18.66	18.08	17.46	4.35	4.03	12.84	24.81
30		1.57e-01	20.05	20.00	20.09	18.50	5.34	4.76	9.09	15.48
45		1.56e-01	20.32	20.21	20.64	19.06	5.98	5.08	7.71	12.03
60		1.87e-01	20.47	20.36	21.00	19.05	6.10	5.06	6.59	9.23
15	10	1.20e-01	17.75	17.73	16.78	16.65	5.81	4.64	12.45	23.54
30		1.15e-01	19.62	19.52	19.44	18.29	6.96	5.36	9.12	14.96
45		1.14e-01	20.26	20.08	20.44	19.00	7.63	5.51	7.56	11.10
60		1.12e-01	20.46	20.30	20.83	19.33	8.13	5.67	7.01	9.09
Average			19.88	19.82	19.86	18.63	5.78	4.90	11.62	15.34

Table 7. SNR and CPU time results for the onion image under Gaussian blur. The SNR values are similar except ADMM-2. The best CPU times are in black.

M	b	Alg.1	Alg.2	ADMM-1	ADMM-2
15	1	0.9904	0.9905	0.9885	0.9894
30		0.9797	0.9797	0.9790	0.9759
45		0.9662	0.9662	0.9660	0.9601
60		0.9537	0.9538	0.9541	0.9447
15	5	0.9876	0.9877	0.9855	0.9862
30		0.9778	0.9778	0.9760	0.9741
45		0.9649	0.9649	0.9644	0.9578
60		0.9506	0.9505	0.9518	0.9410
15	10	0.9854	0.9854	0.9822	0.9843
30		0.9757	0.9756	0.9739	0.9718
45		0.9612	0.9612	0.9616	0.9522
60		0.9472	0.9471	0.9489	0.9352
Average		0.9700	0.9700	0.9693	0.9644

Table 8 SSIM results for the onion image under Gaussian blur.

M	b	$\lambda = 1/\gamma$	SNR (dB)				CPU Time (s)			
			Alg.1	Alg.2	ADMM-1	ADMM-2	Alg.1	Alg.2	ADMM-1	ADMM-2
15	1	1.61e-01	13.67	13.67	13.54	13.35	45.87	44.96	825.91	481.97
30		1.55e-01	14.47	14.47	14.59	13.94	56.19	53.06	792.13	469.45
45		1.47e-01	14.81	14.80	14.99	14.16	59.65	57.09	579.43	361.81
60		1.37e-01	15.04	15.04	15.21	14.35	72.72	62.76	498.85	297.42
15	5	1.14e-01	12.83	12.84	12.36	12.70	60.36	53.83	471.94	453.79
30		1.16e-01	14.15	14.16	14.11	13.68	69.52	59.54	416.19	326.14
45		1.13e-01	14.66	14.66	14.75	14.07	76.91	64.30	318.35	261.95
60		1.12e-01	14.94	14.95	15.05	14.28	82.25	67.42	272.25	215.88
15	10	9.43e-02	12.26	12.28	11.63	12.25	80.83	61.37	390.01	447.99
30		8.66e-02	13.80	13.82	13.61	13.46	94.91	72.06	319.05	305.16
45		8.33e-02	14.50	14.53	14.46	14.01	102.29	77.74	278.00	245.69
60		8.64e-02	14.86	14.89	14.90	14.25	106.64	79.78	255.16	201.57
Average			14.17	14.18	14.10	13.71	75.68	62.83	451.44	339.07

Table 9. SNR and CPU time results for the cells image under Gaussian blur. The SNR values are similar. The best CPU times are in black.

M	b	Alg.1	Alg.2	ADMM-1	ADMM-2
15	1	0.9905	0.9905	0.9896	0.9906
30		0.9801	0.9801	0.9799	0.9791
45		0.9676	0.9676	0.9677	0.9649
60		0.9543	0.9543	0.9547	0.9501
15	5	0.9865	0.9865	0.9837	0.9878
30		0.9766	0.9767	0.9749	0.9765
45		0.9643	0.9644	0.9630	0.9627
60		0.9509	0.9510	0.9503	0.9480
15	10	0.9820	0.9821	0.9774	0.9849
30		0.9725	0.9726	0.9690	0.9739
45		0.9601	0.9603	0.9571	0.9604
60		0.9472	0.9475	0.9450	0.9458
Average		0.9694	0.9695	0.9677	0.9687

Table 10 SSIM results for the cells image under Gaussian blur.

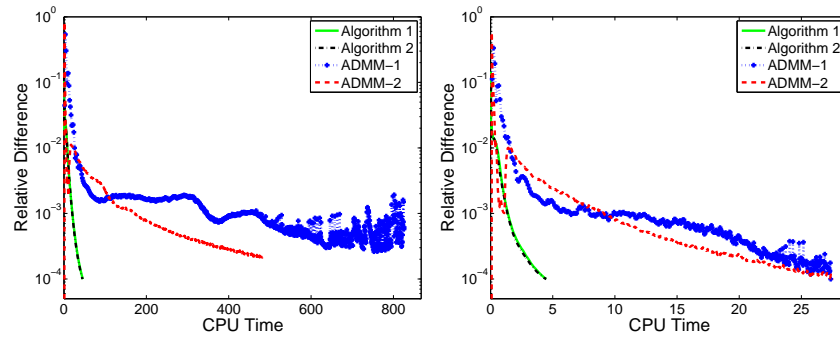


Figure 5. Relative difference (RD) versus CPU time the onion image (left) and the cells image (right) with $M = 15, b = 1$.

- 2 J. Bardsley and J. Goldes. Regularization parameter selection methods for ill-posed Poisson maximum likelihood estimation. *Inverse Problems*, 25:095005, 2009.
- 3 J. Bardsley and J. Goldes. An iterative method for edge-preserving MAP estimation when data-noise is Poisson. *SIAM Journal on Scientific Computing*, 32(1):171–185, 2010.
- 4 M. Bertero, P. Boccacci, G. Desiderà, and G. Vicidomini. Image deblurring with poisson data: from cells to galaxies. *Inverse Problems*, 25(12):123006, 2009.
- 5 D. Bertsekas. *Convex optimization theory*. Athena Scientific Belmont, MA, 2009.
- 6 P. Blomgren and T. Chan. Color TV: total variation methods for restoration of vector-valued images. *IEEE Trans. Image Process.*, 7(3):304–309, March 1998.
- 7 A. Bovik. *Handbook of image and video processing*. Academic Press, 2010.
- 8 S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
- 9 C. Brune, A. Sawatzky, and M. Burger. Bregman-EM-TV methods with application to optical nanoscopy. In *Scale Space and Variational Methods in Computer Vision*, pages 235–246. Springer, 2009.
- 10 C. Brune, A. Sawatzky, and M. Burger. Primal and dual Bregman methods with application to optical nanoscopy. *International Journal of Computer Vision*, 92(2):211–229, 2011.
- 11 A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imag. Vision*, 20(1-2):89–97, 2004.
- 12 A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.
- 13 R. Chan, H. Yang, and T. Zeng. A two-stage image segmentation method for blurry images with poisson or multiplicative gamma noise. *SIAM Journal on Imaging Sciences*, 7(1):98–127, 2014.
- 14 R. Chan and K. Chen. Multilevel algorithm for a Poisson noise removal model with total-variation regularization. *Int. J. Comput. Math.*, 84(8):1183–1198, 2007.
- 15 G. Chen and M. Teboulle. A proximal-based decomposition method for convex minimization problems. *Math. Programming, Ser. A*, 64(1):81–101, 1994.
- 16 P. Combettes and J. Pesquet. A proximal decomposition method for solving convex variational inverse problems. *Inverse Problems*, 24(6):Article No. 065014, 2008.
- 17 F. Dupe, J. Fadili, and J. Starck. A proximal iteration for deconvolving poisson noisy images using sparse representations. *IEEE Trans. Image Process.*, 18(2):310–321, 2009.
- 18 J. Eckstein and D. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Programming, Ser. A*, 55(3):293–318, 1992.
- 19 J. Fessler, S. Lee, V. Olafsson, H. Shi, and D. Noll. Toeplitz-based iterative image reconstruction for mri with correction for magnetic field inhomogeneity. *Signal Processing, IEEE Transactions on*, 53(9):3393–3402, Sept 2005.
- 20 J. Fessler and S. Booth. Conjugate-gradient preconditioning methods for shift-variant pet image reconstruction. *Image Processing, IEEE Transactions on*, 8(5):688–699, 1999.
- 21 M. Figueiredo and J. Bioucas-Dias. Restoration of Poissonian images using alternating direction optimization. *IEEE Transactions on Image Processing*, 19(12):3133–3145, 2010.
- 22 D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Comput. Math. Appl.*, 2(1):17–40, 1976.
- 23 T. Goldstein and S. Osher. The split Bregman method for L1 regularized problems. *SIAM J. Imaging Sci.*, 2:323–343, 2009.
- 24 P. Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*, volume 4. Siam, 1998.

- 25 B. He and X. Yuan. On the $o(1/n)$ convergence rate of the Douglas-Rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.
- 26 T. Le, R. Chartrand, and T.J. Asaki. A variational approach to reconstructing images corrupted by Poisson noise. *Journal of Mathematical Imaging and Vision*, 27(3):257–263, 2007.
- 27 L. Lucy. An iterative technique for the rectification of observed distributions. *Astron. J.*, 79:745–754, 1974.
- 28 F. Murtagh and J. Starck. *Astronomical image and data analysis*. Springer, 2006.
- 29 M. Ng, R. Chan, and W. Tang. A fast algorithm for deblurring models with Neumann boundary conditions. *SIAM J. Sci. Comput.*, 21(3):851–866, 1999.
- 30 R. Puetter, T. Gosnell, and A. Yahil. Digital image reconstruction: deblurring and denoising. *Annu. Rev. Astron. Astrophys.*, 43:139–194, 2005.
- 31 W. Richardson. Bayesian-based iterative method of image restoration. *J. Opt. Soc. Am.*, 62:55–59, 1972.
- 32 R. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. Oper. Res.*, 1(2):97–116, 1976.
- 33 R. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control Optimization*, 14(5):877–898, 1976.
- 34 L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
- 35 S. Setzer, G. Steidl, and T. Teuber. Deblurring Poissonian images by split Bregman techniques. *Journal of Visual Communication and Image Representation*, 21(3):193–199, 2010.
- 36 S. Setzer, G. Steidl, and T. Teuber. Infimal convolution regularizations with discrete l_1 -type functionals. *Comm. Math. Sci*, 9(3):797–872, 2011.
- 37 D. Snyder, A. Hammoud, and R. White. Image recovery from data acquired with a charge-coupled-device camera. *JOSA A*, 10(5):1014–1023, 1993.
- 38 X. Tai and C. Wu. Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model. In *Scale space and variational methods in computer vision*, volume 5567 of *Lecture Notes in Computer Science*, pages 502–513. Springer, 2009.
- 39 K. Timmermann and R. Nowak. Multiscale modeling and estimation of poisson processes with application to photon-limited imaging. *IEEE Transactions on Information Theory*, 45(3):846–862, Apr 1999.
- 40 P. Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM J. Control Optim.*, 29(1):119–138, 1991.
- 41 M. Wernick and J. Aarsvold. *Emission tomography: the fundamentals of PET and SPECT*. Academic Press, 2004.
- 42 R. Willett and R. Nowak. Platelets: a multiscale approach for recovering edges and surfaces in photon-limited medical imaging. *IEEE Transactions on Medical Imaging*, 22(3):332–350, March 2003.
- 43 W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for ℓ_1 -minimization with applications to compressed sensing. *SIAM Journal on Imaging Sciences*, 1(1):143–168, 2008.
- 44 R. Zanella, P. Boccacci, L. Zanni, and M. Bertero. Efficient gradient projection methods for edge-preserving removal of poisson noise. *Inverse Problems*, 25:045010, 24, 2009.
- 45 W. Zhou, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.
- 46 M. Zhu. *Fast Numerical Algorithms for Total Variation Based Image Restoration*. PhD thesis, University of California, Los Angeles, 2008.
- 47 M. Zhu and T. Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report*, 08-34, 2007.