

PRIMAL-DUAL ALGORITHM FOR SOLVING A CONVEX IMAGE DEJITTERING MODEL WITH HYBRID FINITE DIFFERENCES

WEIWEI DENG, JIE LIANG, WENXING ZHANG*

*School of Mathematical Sciences,
University of Electronic Science and Technology of China, Chengdu 611731, China*

Abstract. Jittering is a common phenomenon arising from the area of multimedia data compression and wireless video transmission. The visual abnormality of a jittered image is the jag in edge and loss of synchronization in latitudinal direction. Typically, the problem of intrinsic image dejittering is challenging to be tackled because of the ubiquitous noise in jittered data. In this paper, we develop a convex variational model for solving image dejittering problem by exerting high-order finite differences regularizer in objective function and exploiting linearization to constraints. Upon the recent progress in convex optimization community, the proposed model can be efficiently solved by the first-order primal-dual algorithm. Numerical simulations on recovering both noiseless and noisy jittered data demonstrate the compelling performance of the proposed model.

Keywords. Image dejittering; Independent jitter; Higher-order derivative; Convex programming; Primal-dual algorithm.

1. INTRODUCTION

1.1. Overview of the problem. Jittering usually occurs when a synchronization signal is severely corrupted by storage medium, or electromagnetically interfered during wireless video transmission. The problem of *intrinsic image dejittering* [21, 22] aims to retrieve an “ideal” image solely by recourse to the observed jittered data (i.e., without any a priori information on storage medium).

Let $\Omega := \Omega_x \times \Omega_y \subseteq \mathbb{R}^2$ be a bounded open set denoting the image domain and $u : \Omega \rightarrow \mathbb{R}$ be a function representing an ideal image. A *line jittered* image, denoted by u_s , is formalised by shifting each line of the ideal image u with a small displacement, i.e.,

$$u_s(x + s(y), y) = u(x, y), \quad \forall (x, y) \in \Omega, \quad (1.1)$$

where $s : \Omega_y \rightarrow \mathbb{R}$ is the jitter function indicating jittering process at y -coordinate. Typically, the jitter function s falls into two categories:

- (i) *Independent jitter.* The s is a stochastic function consisting of independent and identically distributed (i.i.d.) random variables, e.g., the Gaussian, uniform and binomial distributed jitter functions in [11, 19, 21, 31].

*Corresponding author.

E-mail address: zhangwx@uestc.edu.cn (W. Zhang).

Received October 4, 2019; Accepted December 11, 2019.

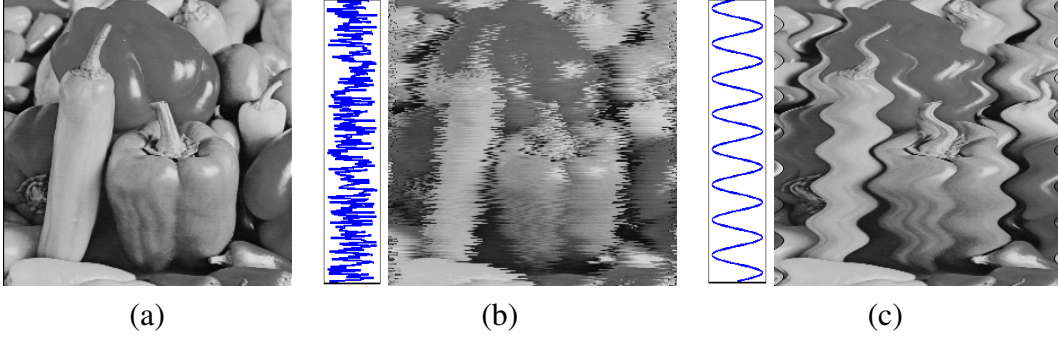


FIGURE 1. Ideal image and line jittered images. (a) Ideal image. (b) Independent jittered image. (c) Structured jittered image.

(ii) *Structured jitter*. The s is a waveform function exhibiting periodicity, e.g., the sinusoidal waveform jitter function in [23, 24, 25].

Fig. 1 displays an ideal image and its line jittered versions under different jitter functions, which illustrates the panoramic abnormality of images in latitudinal direction.

Moreover, due to ubiquitous intensity noises, a *noisy line jittered* image, denoted by u_0 , is produced by further contaminating the ideal image u via

$$u_0(x + s(y), y) = u(x, y) + \varepsilon(x, y), \quad \forall (x, y) \in \Omega, \quad (1.2)$$

where $\varepsilon : \Omega \rightarrow \mathbb{R}$ denotes the additive white noise.

1.2. Related work. There are abundant literatures on both theoretical analysis and practical manipulations for intrinsic image dejittering.

In the seminal work [22], Kokaram et al. proposed an autoregressive (AR) model — combining the multi-resolution scheme and filtering technique — to register shuffled lines in jittered image. Later on, an extension of AR model to 2D circumstance was devised in [21, Chapter 5] for intrinsic video dejittering by exploiting drift compensation skill. The coefficients of those AR models and the displacements of line jitter were simultaneously estimated in iterative manners.

Laborelli [23] explored a dynamic programming scheme for image dejittering. Concisely, given a registered line (i.e., a line without jitter) as reference, the optimal shift of the sequel line was the minimizer of a local cost function, which is defined as the differences between adjacent lines under l^1 -norm. However, as a reshuffled line would be a reference for the remaining lines in image, the error in estimating displacement of a reference line can be damageable to the whole image.

Based on the Bayesian machinery and bounded variations, Shen [31] innovated the first variational model for image dejittering

$$\min_{u, s} \alpha \text{TV}(u) + \frac{\mu}{2} \int_{\Omega_y} s^2 dy + \frac{\lambda}{2} \int_{\Omega} (u_s - u_0)^2 dx dy, \quad (1.3)$$

where $\text{TV}(u) := \int_{\Omega} |\nabla u| dx dy$ is the isotropic total variation of u ; and the triplets (α, μ, λ) are positive parameters to balance the total variation term, the Tikhonov regularizer, and the data fidelity term in objective function, respectively. The alternating minimization algorithm was

employed to acquire an optimum (possibly local) of model (1.3). At each iteration, the u -subproblem (which amounts to an image denoising problem) is solved by the gradient descent method, whilst the s -subproblem (which is a non-smooth non-linear least squares problem) is solved by Newton-Raphson method. When a jittered image contains oscillating textures or grainy patterns, the dummy optimum of model (1.3) may be derived due to the non-smoothness of s -subproblem.

To overcome the non-smoothness of model (1.3) on s -subproblem, Kang and Shen [19] developed a two-stage approach, named as “bake-and-shake”, to solve the image dejittering problem. Concisely, at the “bake” stage, the anisotropic diffusion technique [29] is employed to smooth the jittered image so that those non-smooth components and high frequency irregularities (e.g., the jags in edge, noise and outliers) can be melt away. At the “shake” stage, using the baked image as a prototype, the displacement along individual line is evaluated by solving a nonlinear least squares problem. Later on, Kang and Shen [20] initiated a “slicing moments” technique for image dejittering, whose efficiency is hinged by the maximal displacement in jittered image. As pointed in [25], it is somewhat inadvisable to “bake” the entire jittered image before the “shake” stage because the “bake” process may ruin important information (e.g., the edges) in the latitudinal vertical direction.

Nikolova [25] devised a series of tactful and nearly real-time image dejittering schemes. The jitter at y -th line is calibrated by minimizing the following non-smooth (possibly nonconvex) cost function

$$\min_{s \in \mathfrak{S}} \lim_{\delta \rightarrow 0^+} \frac{1}{2\delta} \int_{Y_y} \int_X \sum_{k=1}^2 \alpha_k \left| \frac{\partial^k}{\partial y^k} u_0(x+s, y) \right|^{p_k} dx dy, \quad (1.4)$$

where α_k ($k = 1, 2$) are positive parameters for balancing the first- and second-order derivatives in integrand; the p_k ($k = 1, 2$) are scalars in $[\frac{1}{2}, 1]$ for indicating norms associated to derivatives; the $\mathfrak{S} \subset \mathbb{R}$ denotes the range of jitter function s ; the X is the “interior” of Ω_x by removing some marginal regions in image domain; and the $Y_y = \mathcal{N}_\delta(y) \cap \Omega_y$ is the δ -neighbourhood of y within Ω_y (see also [11, 25] for details). By forsaking conventional partial differential equations (PDEs) solvers, the minimizer of model (1.4) is acquired by examining exhaustively all possible displacements in the interval \mathfrak{S} . Numerically, the model (1.4) can efficiently and accurately register a noiseless jittered image in discrete context.

Lenzen and Scherzer [24] devised the following PDEs based nonconvex energy functional for image dejittering

$$\min_u \alpha \text{TV}(u) + \frac{1}{2} \int_{\Omega} \left| \frac{\partial u}{\partial x} \right|^{-2} (u - u_0)^2 dx dy, \quad (1.5)$$

and deduced a self-contained theoretical analysis via semi-group theory (see e.g., [30, Chapter 6] for the account of semi-group theory). Therein, the jitter function s was supposed to obey zero-mean Gaussian distribution. Furthermore, they generalized the jittering process from *line jitters* (i.e., the displacements of pixels at the same line are identical) prototype to *line-pixel jitters* (i.e., the displacements of pixels at the same line can be different) and *pixel jitters* (i.e., the displacements of pixels occur in both horizontal and vertical directions) modes. Recently,

Herein, for ease of description, the dejittering schemes developed in [25] are presented in continuum context, see also [11].

Dong and Scherzer [12] further improved the model (1.5) by replacing total variation regularizer with a directional high-order one.

Dong et al. [11] reformulated the image dejittering schemes in [25] as PDEs framework and gave the following modified version

$$\min_{\|s\|_2 \leq \sigma} \int_{\Omega} \left| \frac{\partial^k}{\partial y^k} u_0(x+s, y) \right|^p dx dy, \quad (1.6)$$

where $k = \{1, 2\}$ are options for the first- or second-order derivative; and $p = \{1, 2\}$ are parameters to guarantee the convexity of objective function. Compared to (1.4), the model (1.6) minimizes globally and jointly the jitter function s over all lines whilst the model (1.4) seeks the displacement of each line in a sequential manner.

Basically, the methods for image dejittering can be categorized into *sequential* method (which means that the displacement of each line is retrieved sequentially by minimizing a merit function of adjacent lines, see, e.g., [21, 22, 23, 25]) and the *simultaneous* method (which means that the displacements of all lines are globally sought over the entire image domain by solving an optimization problem, see, e.g., [11, 19, 20, 24, 31]). Although the simultaneous methods are typically time-consuming (because they are usually involved in handling onerous non-smooth and/or non-convex optimization problems) compared to the sequential methods, they generally render more compelling results by deploying the coherence in entire jittered images. Pursuing the track of variational frameworks in [19, 31], we propose a convex variational model for image dejittering by exploiting high-order derivative to the objective function and linearized approximation to the constraint. Different to the non-convex model with sensitive property to the initialization, the proposed model can be robust to any initial points. Most importantly, the proposed model can be efficiently solvable by some state-of-the-art convex optimization algorithms, e.g., [4, 6, 7, 14, 15, 18], to name a few.

The rest of the paper is organized as follows. In Section 2, some basic notations and definitions are stated for the sequel discussion. In Section 3, the image dejittering model is explored by linearizing the image formation (1.1). In Section 4, the first-order primal-dual method for solving generic convex optimization is presented, followed by the implementation details on handling the proposed image dejittering model. In Section 5, numerical simulations are conducted to demonstrate the performance of the proposed model. Finally, some concluding remarks are drawn in Section 6.

2. NOTATIONS AND DEFINITIONS

For any vector $x = (x_1, x_2, \dots, x_n)^\top \in \mathbb{R}^n$, its l^p -norm is denoted by $\|x\|_p := (\sum_{i=1}^n |x_i|^p)^{1/p}$ ($0 < p < +\infty$) and its l^∞ -norm is expressed as $\|x\|_\infty := \max_{1 \leq i \leq n} |x_i|$. Given $\delta > 0$, the δ -neighbourhood of x is the set $\mathcal{N}_\delta(x) := \{z \in \mathbb{R}^n \mid \|z - x\|_2 < \delta\}$. We denote $\mathbf{0}$ the matrix with all elements being zero and \mathbf{I} the identity matrix with appropriate dimension. The $\text{diag}(x)$ represents a diagonal matrix whose diagonal elements are x_i 's.

Let $f : \mathbb{R}^n \rightarrow (-\infty, +\infty]$ be an extended real-valued function. The domain and the epigraph of f are denoted by

$$\text{dom}(f) := \{x \in \mathbb{R}^n \mid f(x) < +\infty\}$$

and

$$\text{epi}(f) := \{(x, r) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) \leq r\},$$

respectively. If the $\text{dom}(f)$ is nonempty, then f is said to be proper. If the $\text{epi}(f)$ is closed, then f is said to be closed. The conjugate of f is a closed convex function, defined by

$$f^*(\lambda) := \sup_{x \in \text{dom}(f)} \{x^\top \lambda - f(x)\}, \quad \forall \lambda \in \mathbb{R}^n. \quad (2.1)$$

Let $\Omega \subset \mathbb{R}^n$ be a measurable set. The $L^p(\Omega)$ ($1 \leq p \leq \infty$) denotes the space of all Lebesgue measurable functions $f : \Omega \rightarrow \mathbb{R}$ satisfying $\|f\|_{L^p(\Omega)} < \infty$, where

$$\|f\|_{L^p(\Omega)} := \begin{cases} (\int_{\Omega} |f(x)|^p dx)^{1/p}, & \text{if } 1 \leq p < \infty, \\ \text{ess sup}_{x \in \Omega} |f(x)|, & \text{if } p = \infty. \end{cases}$$

The support of f , denoted by $\text{spt } f$, is the closure of set in which f does not vanish. A non-negative function $\eta \in L^1(\mathbb{R}^n)$ is called mollifier if it satisfies $\int_{\mathbb{R}^n} \eta(x) dx = 1$ and $\text{spt } \eta \subset \mathcal{N}_1(\mathbf{0})$. Particularly, for any $\varepsilon > 0$, the scaled version of η , which is defined by

$$\eta_\varepsilon(x) := \frac{1}{\varepsilon^n} \eta\left(\frac{x}{\varepsilon}\right), \quad \forall x \in \mathbb{R}^n \quad (2.2)$$

is also a mollifier. In practice, the η is usually taken as the following function in $C^\infty(\mathbb{R}^n)$

$$\eta(x) = \begin{cases} c \exp\left(-\frac{1}{\|x\|_2^2 - 1}\right), & \text{if } \|x\|_2 < 1, \\ 0, & \text{otherwise,} \end{cases} \quad (2.3)$$

where $c > 0$ is a parameter satisfying $\int_{\mathbb{R}^n} \eta(x) dx = 1$. The following theorem states some properties of mollifier (see e.g., [1, Chapter 2]).

Lemma 2.1. *Let $f \in L^p(\mathbb{R}^n)$ with $p \in [1, \infty)$. Then, for any mollifier η_ε in (2.2), the convolution of f with η_ε satisfies: (i) $f \star \eta_\varepsilon \in L^p(\mathbb{R}^n)$; (ii) $\|f \star \eta_\varepsilon\|_{L^p(\mathbb{R}^n)} \leq \|f\|_{L^p(\mathbb{R}^n)}$; (iii) $\lim_{\varepsilon \rightarrow 0} \|f \star \eta_\varepsilon - f\|_{L^p(\mathbb{R}^n)} = 0$.*

The indicator function of a set $\Omega \subset \mathbb{R}^n$, denoted by ι_Ω , is defined as

$$\iota_\Omega(x) := \begin{cases} 0, & x \in \Omega, \\ +\infty, & \text{otherwise.} \end{cases}$$

For any $g \in \mathbb{R}^{n \times m}$ with g_{ij} as its (i, j) -th entry, the $|g|$ denotes a vector in \mathbb{R}^n with its i -th entry as

$$|g|_i := \left(\sum_{j=1}^m g_{ij}^2 \right)^{1/2}.$$

For any $\tau > 0$ and $g \in \mathbb{R}^{n \times m}$, the multi-dimensional soft-thresholding operator $\mathcal{S}_\tau : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{n \times m}$ is defined as (see e.g., [26, 33])

$$(\mathcal{S}_\tau(g))_{ij} := \frac{g_{ij}}{|g|_i} \max\{|g|_i - \tau, 0\}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, m, \quad (2.4)$$

where $\frac{g_{ij}}{|g|_i}$ is taken as 0 if $|g|_i = 0$. Let Ω be a non-empty closed convex set. The projection operator from \mathbb{R}^n onto Ω , denoted by Π_Ω , is defined as

$$\Pi_\Omega(x) := \underset{y \in \Omega}{\operatorname{argmin}} \|y - x\|_2, \quad \forall x \in \mathbb{R}^n.$$

The proximity of $f : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$, denoted by prox_f , is defined as

$$\text{prox}_f(x) := \underset{y \in \Omega}{\operatorname{argmin}} \left\{ f(y) + \frac{1}{2} \|y - x\|_2^2 \right\}, \quad \forall x \in \mathbb{R}^n. \quad (2.5)$$

Specially, the proximity of an indicator function ι_Ω is the projection operator, i.e., $\text{prox}_{\iota_\Omega} = \Pi_\Omega$; and the proximity of l^1 -norm is the soft-thresholding operator, i.e., $\text{prox}_{\tau \|\cdot\|_1} = \mathcal{S}_\tau$.

The interested reader is referred to, e.g., the monograph [2], for more properties of the conjugate, the projection and the proximity in Hilbert spaces.

3. CONVEX MODEL FOR IMAGE DEJITTERING

In this section, we present our image dejittering model by linearizing the image formation (1.1), followed by some remarks on its differences to the models in [19, 24, 25].

Under the differentiability assumption on the jittered image u_s along x -coordinate, the Taylor expansion of u_s with respect to variable x reads

$$u_s(x+s, y) = u_s(x, y) + \frac{\partial u_s(x, y)}{\partial x} s + R(x, y, s), \quad \forall (x, y) \in \Omega, \quad (3.1)$$

where $R(x, y, s) := \int_0^s \frac{\partial^2}{\partial x^2} u_s(x+t, y)(s-t) dt$ is the infinitesimal remainder. By subtracting (1.1) from (3.1), we obtain

$$u(x, y) = u_s(x, y) + \frac{\partial u_s(x, y)}{\partial x} s + R(x, y, s), \quad \forall (x, y) \in \Omega. \quad (3.2)$$

By ignoring the term R in (3.2), we have

$$u(x, y) \approx u_s(x, y) + \frac{\partial u_s(x, y)}{\partial x} s, \quad \forall (x, y) \in \Omega. \quad (3.3)$$

Thus, the error in the above approximation comes from the intrinsic finite difference approximation error (the $O(s^2)$ term). Finally, we propose the following convex model for registering noiseless jittered image

$$\min_{w, s} \int_{\Omega} \sum_{k=1}^2 \alpha_k \left| \frac{\partial^k w}{\partial y^k} \right| dx dy + \phi(s) \quad (3.4a)$$

$$\text{subject to} \quad \left\| w - u_s - \frac{\partial u_s}{\partial x} s \right\|_{L^p(\Omega)} \leq \delta, \quad w \in [0, 255], \quad (3.4b)$$

where the parameters $\alpha_k > 0$ ($k = 1, 2$) play the same roles as those of model (1.4); the scalar $p \in [1, +\infty]$ is used for measuring the linearized approximation of jittered image u_s ; and $\delta > 0$ restricts the deviation between ideal image and the first-order Taylor approximation of jittered image along x -coordinate. The abstract constraint $u \in [0, 255]$ in (3.4b) restricts the range of pixel values, which generally promotes better restorations in image processing (see, e.g., [8]). The integral term $\int_{\Omega} \sum_{k=1}^2 \alpha_k \left| \frac{\partial^k u}{\partial y^k} \right| dx dy$ and the ϕ in objective function are regularizers for image u and jitter s , respectively. Herein, the ϕ can be adaptively chosen by the characteristics of jitter function s . Specifically,

- (1) Independent jitter. The Tikhonov regularizer $\phi(s) := \int_{\Omega} |s|^2 dy$ can be applicable to the zero-mean randomly distributed jitters, e.g., the jitter function s obeys binomial, Gaussian or uniform distributions.

- (2) Structured jitter. As the jitter of this kind is typically sparse or compressible under some transforms (e.g., FFT, curvelets), the regularizer ϕ can be set as $\phi(s) := \int_{\Omega_y} |\Phi(s)| dy$, where Φ denotes the dictionary of transform. For example, the Φ can be taken as analysis operator of (Parseval) tight-frame [5] satisfying $\Phi\Phi^\top = c\mathbf{I}$ with $c > 0$.

Essentially, the δ in (3.4b) controls the magnitude of infinitesimal remainder R . The following proposition provides a lower bound of δ , which will be useful for the parametric selections on numerical simulations. Given any $(x, y) \in \Omega$, we define

$$h(t) := u_s(x+t, y). \quad (3.5)$$

Proposition 3.1. *Assume that the jitter function s satisfies $|s(y)| \leq M$ for all $y \in \Omega_y$. Then*

$$\delta \geq 2M \left\| \frac{\partial u_s}{\partial x} \right\|_{L^p(\Omega)}. \quad (3.6)$$

Proof. For the i -th row of jittered image u_s , it can be viewed as the rigid transform of the i -th row of ideal image u (regardless of boundary conditions). Accordingly, the $u_s(\cdot, y)$ and $u(\cdot, y)$ fall into the same functional space for any $y \in \Omega_y$. By combining the definition of $R(x, y, s)$ in (3.1), we derive

$$R(x, y, s) = \int_0^s h''(t)(s-t) dt = \int_0^s h'(t) dt - sh'(0), \quad \forall (x, y) \in \Omega, \quad (3.7)$$

where the last equality follows from the integration by parts formula. By the identity $|a+b|^p \leq 2^{p-1}(|a|^p + |b|^p)$ for any $a, b \in \mathbb{R}$ and $p \in [1, \infty)$, we have

$$\begin{aligned} |R(x, y, s)|^p &= \left| \int_0^s h'(t) dt - sh'(0) \right|^p \\ &\leq 2^{p-1} \left(\left| \int_0^s h'(t) dt \right|^p + |sh'(0)|^p \right), \quad \forall (x, y) \in \Omega. \end{aligned} \quad (3.8)$$

By further using the Hölder inequality, it follows that

$$\begin{aligned} \left| \int_0^s h'(t) dt \right| &\leq \left(\int_0^s 1 dt \right)^{1/q} \left(\int_0^s |h'(t)|^p dt \right)^{1/p} \\ &= s^{1/q} \left(\int_0^s |h'(t)|^p dt \right)^{1/p}, \quad \forall (x, y) \in \Omega, \end{aligned} \quad (3.9)$$

where $q > 0$ is a scalar satisfying $\frac{1}{p} + \frac{1}{q} = 1$. By substituting (3.9) into (3.8), we have

$$\begin{aligned} |R(x, y, s)|^p &\leq 2^{p-1} \left(s^{p/q} \int_0^s |h'(t)|^p dt + |sh'(0)|^p \right) \\ &= 2^{p-1} \left(s^{p-1} \int_0^s |h'(t)|^p dt + |sh'(0)|^p \right) \\ &\leq 2^{p-1} \left(M^{p-1} \int_0^M |h'(t)|^p dt + M^p |h'(0)|^p \right), \quad \forall (x, y) \in \Omega, \end{aligned} \quad (3.10)$$

where the last inequality is due to the assumption $|s(y)| \leq M$ for all $y \in \Omega_y$. Under the periodic boundary condition on image u_0 , we have

$$\|h'(t)\|_{L^p(\Omega)} = \|h'(0)\|_{L^p(\Omega)} = \left\| \frac{\partial u_s}{\partial x} \right\|_{L^p(\Omega)}, \quad \forall t \in \mathbb{R}. \quad (3.11)$$

By integrating (3.10) on Ω , we deduce that

$$\begin{aligned} \int_{\Omega} |R(x, y, s)|^p dx dy &\leq 2^{p-1} \left(M^{p-1} \int_0^M \int_{\Omega} |h'(t)|^p dx dy dt + M^p \int_{\Omega} |h'(0)|^p dx dy \right) \\ &= 2^{p-1} \left(M^{p-1} \int_0^M \|h'(t)\|_{L^p(\Omega)}^p dt + M^p \|h'(0)\|_{L^p(\Omega)}^p \right) \\ &= 2^{p-1} \left(M^{p-1} \int_0^M \left\| \frac{\partial u_s}{\partial x} \right\|_{L^p(\Omega)}^p dt + M^p \left\| \frac{\partial u_s}{\partial x} \right\|_{L^p(\Omega)}^p \right) \\ &= 2^p M^p \left\| \frac{\partial u_s}{\partial x} \right\|_{L^p(\Omega)}^p, \end{aligned}$$

where the second equation is due to (3.11). By rearranging terms of (3.2), we deduce that

$$\begin{aligned} \left\| u - u_s - \frac{\partial u_s}{\partial x} s \right\|_{L^p(\Omega)} &= \|R\|_{L^p(\Omega)} = \left(\int_{\Omega} |R(x, y, s)|^p dx dy \right)^{1/p} \\ &\leq 2M \left\| \frac{\partial u_s}{\partial x} \right\|_{L^p(\Omega)}, \end{aligned} \quad (3.12)$$

which indicates that the (3.4b) can be satisfied as soon as $\delta \geq 2M \left\| \frac{\partial u_s}{\partial x} \right\|_{L^p(\Omega)}$. \square

Remark 3.1. The above analysis is built upon the noiseless jittered image u_s . Strictly speaking, the noisy jittered image u_0 is generally non-differentiable because of the ubiquitous noise and outliers. When recovering noisy jittered image u_0 (i.e., the image formation (1.2)), the model (3.4b) will be not well-defined by replacing u_s with u_0 directly. Thus, the u_0 should be smoothed by, e.g., the mollifier in (2.2). Let $\tilde{u}_0 = \eta_\varepsilon \star u_0$ be a mollification of u_0 . Consequently, the modification of (3.4)

$$\min_{w, s} \int_{\Omega} \sum_{k=1}^2 \alpha_k \left| \frac{\partial^k w}{\partial y^k} \right| dx dy + \phi(s) \quad (3.13a)$$

$$\text{subject to } \left\| w - \tilde{u}_0 - \frac{\partial \tilde{u}_0}{\partial x} s \right\|_{L^p(\Omega)} \leq \delta, \quad u \in [0, 255], \quad (3.13b)$$

is well-defined for recovering noisy jittered image.

Remark 3.2. The approaches in [19, 24, 25] are representative paradigms to recover noisy jittered images.

- For Nikolova's method [25], it comprises three steps when dealing with a noisy jittered image (see also [25, Section 7.2] for details):
 - Step 1: Mollify “slightly” each line of a noisy jittered image by 1D signal denoising solver (e.g., the fast shrinkage estimator [10]);

- Step 2: Register the resultant image from Step 1 by a noiseless image dejittering solver (e.g., the Algorithm 1 in [25]);
- Step 3: Denoise the resultant image from Step 2 by a 2D image denoising solver (e.g., the enhanced-denoising program in CurveLab 2.1.2 toolbox).

The above image dejittering procedure involves several parameters to be selected at each step, which is somewhat complicated to manipulate.

- For Kang and Shen’s method [19], it is easy to manipulate (compared to Nikolova’s method in [25]) by only “bake” and “shake” stages. However, the non-convexity of image dejittering model in [19] (see also the model (1.3) in this paper) entails it unwarranted to acquire the global optima. Thus, it performs imperfectly on noisy jittered images with textures (see numerical experiments on Rose and Boat images in [19]). On the other hand, as pointed in [25], it is not suggested to “bake” the entire jittered image before the “shake” stage because the “bake” process may ruin important information in vertical direction.
- For Scherzer et al’s methods in [12, 24], by rewriting (1.1) into $u_s(x, y) = u(x - s, y)$ and deploying the linearization to $u(x - s, y)$ at (x, y) , it follows that $u_s \approx u - \frac{\partial u}{\partial x}s$, or equivalently, $\left(\frac{\partial u}{\partial x}\right)^{-1} (u - u_0) \approx s$ when $\frac{\partial u}{\partial x}$ is nonzero. Under the assumption of Gaussian distribution on jitter s , the non-convex fidelity term $\int_{\Omega} \left| \frac{\partial u}{\partial x} \right|^{-2} |u - u_0|^2 dx dy$ (see also (1.5) in this paper) was introduced. The non-convex models in [12, 24] were handled by solving the corresponding Euler-Lagrange equations, which essentially falls into the algorithmic framework of gradient descent methods.
- For our approach, the convexity of model (3.4) makes it easy to be handled by algorithms in convex optimization community (see Subsection 4.2 for details). The global optima can be always acquired in $\mathcal{O}(1/k)$ iterative complexity [7].

4. THE FIRST-ORDER PRIMAL-DUAL ALGORITHM

Consider the following separable convex optimization

$$\min_{x \in \mathbb{R}^n} g(\mathbf{x}) + f(B\mathbf{x}) \quad (4.1)$$

where $f : \mathbb{R}^m \rightarrow (-\infty, +\infty]$ and $g : \mathbb{R}^n \rightarrow (-\infty, +\infty]$ are proper closed convex functions; and the $B : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a linear operator.

A large family of examples (4.1) arise from the research of regularization-based optimization. Empirically, the first-order algorithms dominate the solvers for convex optimization (4.1) in compressive sensing, image processing and computer vision. Among them, the primal-dual algorithm (see, e.g., [6, 7, 14, 17, 18]) and the alternating direction method of multipliers (ADMM, see overviews [4, 15] and references therein) are two popular paradigms for solving separable convex optimization problems.

The first-order primal-dual algorithms allow us to independently cope with individual sub-function (i.e., f and g) in (4.1) by the aid of proximal operator (see (2.5) for definition). The iterative scheme of the first-order primal-dual algorithm proposed in [18, Algorithm 4] is shown as Algorithm 1 below.

Algorithm 1 First-order primal-dual algorithm for solving (4.1)

Require: Choose $\rho \in (0, 2)$, $\tau > 0$, $\sigma > 0$ satisfying $\tau\sigma\|\mathbf{B}^\top \mathbf{B}\|_2 < 1$, the initial point $(\mathbf{x}^0, \mathbf{y}^0) \in \mathbb{R}^n \times \mathbb{R}^m$, and the stopping tolerance $\varepsilon > 0$.

- 1: **repeat**
- 2: $\tilde{\mathbf{x}}^k = \text{prox}_{\tau g}(\mathbf{x}^k - \tau \mathbf{B}^\top \mathbf{y}^k)$;
- 3: $\tilde{\mathbf{y}}^k = \text{prox}_{\sigma f^*}(\mathbf{y}^k + \sigma \mathbf{B}(2\tilde{\mathbf{x}}^k - \mathbf{x}^k))$;
- 4: $\mathbf{x}^{k+1} = \mathbf{x}^k - \rho(\mathbf{x}^k - \tilde{\mathbf{x}}^k)$;
- 5: $\mathbf{y}^{k+1} = \mathbf{y}^k - \rho(\mathbf{y}^k - \tilde{\mathbf{y}}^k)$;
- 6: **until** $\max\{\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2, \|\mathbf{y}^k - \tilde{\mathbf{y}}^k\|_2\} < \varepsilon$.

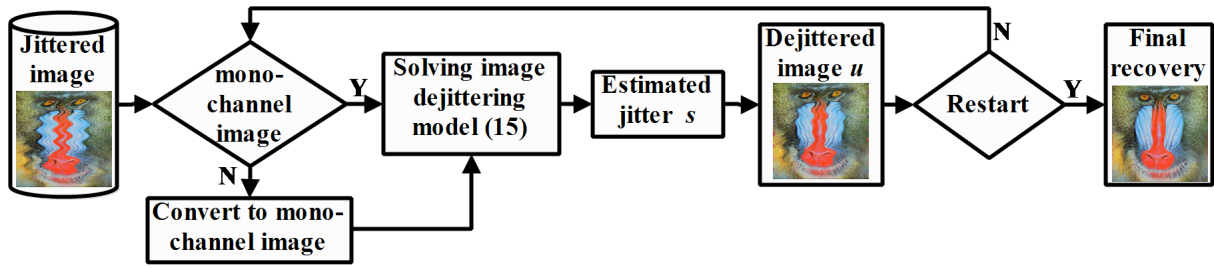


FIGURE 2. The flowchart of our image dejittering process.

Remark 4.1. Compared to the classical ADMM in [4, 15], the superiorities of the first-order primal-dual algorithm is that it suffices to calculate the proximity operators (more precisely, the $\text{prox}_{\tau g}$ and $\text{prox}_{\sigma f^*}$ in Algorithm 1) and the matrix-vector multiplications (i.e., the \mathbf{B} and \mathbf{B}^\top in Algorithm 1). This favourable algorithmic framework enables it fairly versatile to solve (4.1), in which the proximity operators are tractably solvable and the matrix \mathbf{B} is non-structural (e.g., $\mathbf{B}^\top \mathbf{B}$ can not be diagonalized by fast transform). The reader is referred to the recent literature [3, 16, 27] for more comparisons between the first-order primal-dual algorithm and the classical ADMM.

We now expatiate our approach for image dejittering, especially the implementation details on solving the proposed model (3.4) by Algorithm 1. The procedures of our approach have three main steps: (i) Pre-processing on jittered image, (ii) Implementation of Algorithm 1 on model (3.4), and (iii) Post-processing on dejittered image. Fig. 2 displays the flowchart of our image dejittering process.

4.1. Pre-processing on jittered image. In order to tackle the model (3.4) numerically, we digitalize the image domain Ω as a 2D Cartesian grid $\{1, 2, \dots, n_1\} \times \{1, 2, \dots, n_2\}$. Correspondingly, the continuous image $u(x, y)$ is divided into n_1 rows and n_2 columns. The reader is referred to, e.g., [30, Chapter 2], for details on reformulating a continuous image processing model into a discrete one. With a slight abuse of notations, we denote by $u \in \mathbb{R}^{n_1 \times n_2}$, $u_s \in \mathbb{R}^{n_1 \times n_2}$ and $u_0 \in \mathbb{R}^{n_1 \times n_2}$ the discrete version of the ideal image, noiseless jittered image and noisy jittered image, respectively. The pixel values of ideal image u are rescaled into $[0, 255]$. The periodic boundary condition is adopted for pixels beyond image domain Ω . The $s = (s_1, s_2, \dots, s_{n_2})^\top \in \mathbb{R}^{n_2}$ denotes the jitter vector with s_i indicating the displacement at the

i -th line. Accordingly, the image formation (1.1) can be discretized as

$$u_s(i, j) = u(i - s_j, j), \quad \forall (i, j) \in \Omega. \quad (4.2)$$

Note that if the entry of s , say s_j , is not integer, we resort to the spline interpolation to derive the pixel values of u at coordinate $(i - s_j, j)$.

Remark 4.2. For the multichannel jittered data (e.g., RGB images), under assumption that the jitter vectors s are equal for all channels in the same line (see also [25]), we can superimpose the multichannel data into monochannel data, or simply, choose only one channel as the target jittered data to estimate the jitter.

On the other hand, we can also concatenate horizontally all channels of multichannel data, and dejitter the concatenated “monochannel” image directly. For instance, given an RGB image $u \in \mathbb{R}^{n_1 \times n_2 \times 3}$, the image dejittering procedure can be carried out on $\tilde{u} = [u_R, u_G, u_B] \in \mathbb{R}^{n_1 \times (3n_2)}$. This way usually promotes better estimation of jitter vector s since the number of pixels in each row is enlarged.

4.2. Implementation details on (3.4) by Algorithm 1. For ease of notation, we reshape an n_1 -by- n_2 image as a vector in \mathbb{R}^n ($n = n_1 n_2$) by vectorizing its pixels in lexicographic order. Let $\nabla : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times 2}$ and $H : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times 4}$ denote respectively the first- and second-order derivative operators (see Appendix for details), i.e.,

$$\nabla = \begin{bmatrix} \nabla_x \\ \nabla_y \end{bmatrix} \quad \text{and} \quad H = \begin{bmatrix} \nabla_{xx} & \nabla_{xy} \\ \nabla_{yx} & \nabla_{yy} \end{bmatrix}. \quad (4.3)$$

Accordingly, the model (3.4) can be discretized as the following convex optimization problem

$$\min_{w, s} \quad \alpha_1 \|\nabla_y w\|_1 + \alpha_2 \|\nabla_{yy} w\|_1 + \phi(s) \quad (4.4a)$$

$$\text{subject to} \quad \|w - u_s - \text{diag}(\nabla_x u_s) s\|_p \leq \delta, \quad (4.4b)$$

$$0 \leq w \leq 255, \quad (4.4c)$$

where $\alpha_i > 0$ ($i = 1, 2$) and $\delta > 0$ are parameters defined in (3.4). The Tikhonov regularizer $\phi(s) = \frac{1}{2} \|s\|_2^2$ is used for dejittering images with independent jitter, whilst $\phi(s) = \|\Phi s\|_1$ with Φ being the dictionary (e.g., wavelets, curvelets, etc) is used for dejittering images with structured jitter.

Though many algorithms in convex optimization community can be readily applicable to problem (4.4), we pay our attention to a class of first-order algorithms. We herein adopt the primal-dual algorithm in [18] (i.e., Algorithm 1 in Section 4) to solve (4.4). First, problem (4.4) can be reformulated as

$$\min_{w, s} \quad \alpha_1 \|\nabla_y w\|_1 + \alpha_2 \|\nabla_{yy} w\|_1 + \iota_{\mathcal{W}}(w) + \phi(s) + \iota_{\mathcal{Z}}(w - [\nabla_x u_s] s), \quad (4.5)$$

where $\iota_{\mathcal{W}}$ and $\iota_{\mathcal{Z}}$ are indicator functions of closed convex sets $\mathcal{W} := \{w \in \mathbb{R}^n \mid 0 \leq w \leq 255\}$ and $\mathcal{Z} := \{z \in \mathbb{R}^n \mid \|z - u_s\|_p \leq \delta\}$, respectively. In essence, the problem (4.5) falls into the framework of (4.1) with the following specifications:

- The variables $\mathbf{x} := (w, s)$, $\mathbf{y} := (v_1, v_2, z)$;
- The objective functions are $g(\mathbf{x}) := \iota_{\mathcal{W}}(w) + \phi(s)$ and $f(\mathbf{y}) := \alpha_1 \|v_1\|_1 + \alpha_2 \|v_2\|_1 + \iota_{\mathcal{Z}}(z)$;

- The matrix

$$\mathbf{B} := \begin{pmatrix} \nabla_y & \mathbf{0} \\ \nabla_{yy} & \mathbf{0} \\ \mathbf{I} & -\text{diag}(\nabla_x u_s) \end{pmatrix}. \quad (4.6)$$

To apply Algorithm 1, we should compute the proximity operators $\text{prox}_{\sigma f^*}$ and $\text{prox}_{\tau g}$. Fortunately, both proximity operators admit explicitly analytical formulae. Specifically,

- The proximity operator $\text{prox}_{\sigma f^*}$ can be computed by exploiting the Moreau identity (see [2, Chapter 14]), i.e.,

$$\text{prox}_{\sigma f^*}(\mathbf{y}) = \mathbf{y} - \sigma \text{prox}_{\frac{f}{\sigma}}\left(\frac{\mathbf{y}}{\sigma}\right), \quad \forall \mathbf{y} \in \mathbb{R}^n, \quad (4.7)$$

which indicates that it suffices to deduce the $\text{prox}_{\frac{f}{\sigma}}(\frac{\mathbf{y}}{\sigma})$ in right-hand side of (4.7). By the separable property of f in variable $\mathbf{y} = (v_1, v_2, z)$, we have

$$\text{prox}_{\frac{f}{\sigma}}\left(\frac{\mathbf{y}}{\sigma}\right) = \begin{pmatrix} \text{prox}_{\frac{\alpha_1}{\sigma} \|\cdot\|_1}\left(\frac{v_1}{\sigma}\right) \\ \text{prox}_{\frac{\alpha_2}{\sigma} \|\cdot\|_1}\left(\frac{v_2}{\sigma}\right) \\ \text{prox}_{\iota_{\mathcal{Z}}}\left(\frac{z}{\sigma}\right) \end{pmatrix} = \begin{pmatrix} \mathcal{S}_{\frac{\alpha_1}{\sigma}}\left(\frac{v_1}{\sigma}\right) \\ \mathcal{S}_{\frac{\alpha_2}{\sigma}}\left(\frac{v_2}{\sigma}\right) \\ \Pi_{\mathcal{Z}}\left(\frac{z}{\sigma}\right) \end{pmatrix}, \quad (4.8)$$

where \mathcal{S} is the soft-thresholding operator defined in (2.4), and $\Pi_{\mathcal{Z}}$ is the projection operator onto \mathcal{Z} , which admits closed-form formula or can be easily solved up to high precisions. Specifically, the projection onto \mathcal{Z} can be computed as follows:

- If $p = \infty$, we have

$$[\Pi_{\mathcal{Z}}(z)]_i = \text{median}\{(u_s)_i - \delta, (u_s)_i, (u_s)_i + \delta\}, \quad i = 1, 2, \dots, n.$$

- If $p = 2$, we have

$$\Pi_{\mathcal{Z}}(z) = u_s + \min\left\{1, \frac{\delta}{\|z - u_s\|}\right\}(z - u_s). \quad (4.9)$$

- If $p = 1$, $\Pi_{\mathcal{Z}}(z)$ can be efficiently solved by some existing subroutines, e.g., [9, 13]. The interested reader is referred to <https://lts2.epfl.ch/unlocbox> for MATLAB package on calculating proximity and projection operators.

- Analogously, the proximity operator $\text{prox}_{\tau g}$ can be deduced by the separable property of g in variable $\mathbf{x} = (w, s)$

$$\text{prox}_{\tau g}(\mathbf{x}) = \begin{pmatrix} \Pi_{\mathcal{W}}(w) \\ \text{prox}_{\tau\phi}(s) \end{pmatrix}, \quad (4.10)$$

where $\Pi_{\mathcal{W}}$ is computed componentwisely by

$$[\Pi_{\mathcal{W}}(w)]_i = \text{median}\{0, w_i, 255\}, \quad i = 1, 2, \dots, n. \quad (4.11)$$

As for the $\text{prox}_{\tau\phi}$, when $\phi(s) = \frac{1}{2}\|s\|_2^2$, it can be derived by

$$\text{prox}_{\tau\phi}(s) = \frac{s}{\tau + 1}, \quad (4.12)$$

and when $\phi(s) = \|\Phi s\|_1$ with Φ being a (Parseval) tight-frame satisfying $\Phi\Phi^\top = c\mathbf{I}$, the $\text{prox}_{\tau\phi}$ can be computed by (see [28] for details)

$$\text{prox}_{\tau\phi}(s) = s - \frac{1}{c}\Phi^\top(\Phi s - \mathcal{S}_{\tau c}(\Phi s)). \quad (4.13)$$

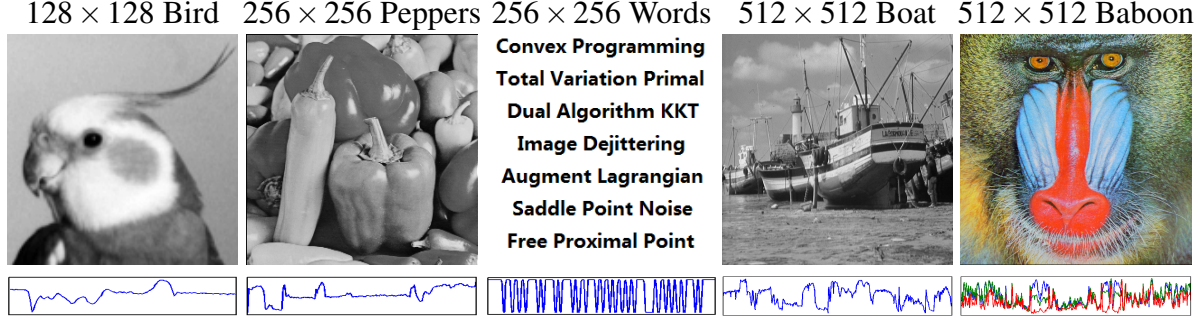


FIGURE 3. Ideal image (top) and the profile along a horizontal line (bottom).

Overall, the componentwise operations in (4.8) and (4.10) indicate that the computation of Algorithm 1 when solving (4.4) involves merely the matrix-vector multiplications.

4.3. Post-processing on dejittered image. Let (\bar{u}, \bar{s}) be the numerical solution of (4.4). There are two ways to obtain the dejittered image:

- (1) The \bar{u} can be intuitively perceived as the dejittered image. However, it is typically unattractive because \bar{u} is obtained by iteratively solving the u -subproblem and the pixels values of \bar{u} may be corrupted by the spline interpolation (recall that the spline interpolation is deployed when the displacement of a certain line, say s_j , is not an integer).
- (2) The \bar{s} can be used to retrieve the dejittered image via

$$\hat{u}(i, j) = u_s(i - \bar{s}_j, j), \quad \forall i = 1, \dots, n_1, j = 1, \dots, n_2, \quad (4.14)$$

where \hat{u} represents the dejittered image. If the \hat{u} contains noise, it should be further restored by some image denoising solvers.

On the other hand, an important technique for image dejittering (see also [19]) is to restart the iteration at every N (e.g., $N \approx 20$) steps by setting \bar{u} as initial point, which can improve the dejittered results substantially.

5. NUMERICAL EXPERIMENTS

In this section, we test the numerical performance of our approach on some image dejittering problems, compared to the Nikolova's method [25], Kang and Shen's method [19], and Lenzen and Scherzer's method [24]. All codes are written by MATLAB 7.9 and all experiments are conducted on a Lenovo personal computer with Intel Core (TM) CPU 2.30GHZ and 8G memory.

The test images and their profiles along some horizontal lines are shown in Fig. 3. By visualizing the profiles of those test images, it seems that the "Bird" and "Peppers" images are piecewise smooth (cartoon-dominant); the "Words" image is binary and piecewise constant; the "Boat" image is a natural image with many discontinuities (e.g., the regions with cloud and masts); and the "Baboon" image is an oscillating-dominant RGB image. Basically, the test images in Fig. 3 are representative samples of immense image databases.

To quantify the effectiveness of test methods, we adopt the following measurement metrics for the recovered image:

- (1) The peak signal-to-noise ratio (PSNR), which is widely used as an image quality assessment, is defined by

$$\text{PSNR} = 20 \log_{10} \frac{r\sqrt{n}}{\|\hat{u} - u\|_2}, \quad (5.1)$$

where u is the ideal image; \hat{u} is an approximation; and $r := \max_{1 \leq i \leq n} u_i - \min_{1 \leq i \leq n} u_i$ indicates the range of pixel values.

- (2) The structural similarity (SSIM), which perceived visual quality better than PSNR, is an alternative measurement for image quality assessment [32]. A MATLAB package for computing SSIM is available at <http://www.cns.nyu.edu/~lcv/ssim/>.

Note that we adopt the strategy in [25, Section 3] so as to eliminate the bias on PSNR or SSIM values posed by boundary noise (i.e., those ragged pixels at the left- and right-boundaries of de-jittered image, see e.g., the recovered “Bird” image by Nikolova’s method in Fig. 6). Concretely, we first shear l columns of $\hat{u} \in \mathbb{R}^{n_1 \times n_2}$ near to the left- and right-boundaries. The sheared version is denoted by $\hat{u}_e \in \mathbb{R}^{n_1 \times (n_2 - 2l)}$. Then, the PSNR value is essentially defined as the maximum PSNR values of \hat{u}_e compared to all possible ground-truth $u(:, j : j + n_2 - 2l - 1)$ for $j = 1, 2, \dots, 2l + 1$. In the upcoming numerical simulations, the l is taken as $1 + \|s\|_\infty$ with s being the jitter vector. The same process is conducted for computing those SSIM values.

5.1. Noiseless jittered data. We first conduct the numerical experiment on de-jittering noiseless jittered images. The test images are 256-by-256 “Peppers” and “Words” in Fig. 3. Both ideal images are degraded by some jitters via image formation (4.2) with $\varepsilon \equiv 0$. Concretely, the “Peppers” image is degraded by independent jitters with the jitter vector s obeying uniform distribution on $[-8, 8]$, and the “Words” image is degraded by a structured sinusoidal waveform jitter with the i -th entry of jitter vector s being $8 \sin(16\pi i)$. The ground truth jitter vectors and the corresponding jittered images are displayed in the first column of Fig. 4.

To apply our approach, we should settle down some parameters in the proposed model (4.4) and Algorithm 1. We take the trade-off parameters $(\alpha_1, \alpha_2) = (50, 0.01)$ and the scalar $\delta = 16 \|\nabla_x u_0\|_2$ for our model (4.4), where the δ is chosen based on the Proposition 3.1. To implement Algorithm 1, we choose the parameters $(\rho, \tau, \sigma) = (1.8, 0.01, 100)$ by trial-and-error. The initial point is taken as $(\mathbf{x}^0, \mathbf{y}^0) = (\mathbf{0}, \mathbf{0})$ and the stopping tolerance is set $\varepsilon = 3 \times 10^{-4} \|\mathbf{u}_0\|_2$.

For Nikolova’s method [25], the Algorithm 1(c) therein is exploited to recover the noiseless jittered images. The parameters for Algorithm 1(c) in [25] are taken as $\alpha = 1$ and $\beta = 5$. For the methods in [19, 24], the parameters for each test image are tuned by trial-and-error to maximize the PSNR values of de-jittered images. The de-jittered images by all test methods are listed in Fig. 4. Therein, the PSNR (in decibel) and the SSIM values of those de-jittered images are also reported for comparisons.

From this numerical experiment, we find that both the cartoon-dominant images (e.g., “Peppers”) and the oscillating-dominant images (e.g., “Words”) can be well de-jittered by our approach. The Nikolova’s method in [25] performs attractively (almost renders the ideal images) for de-jittering noiseless jittered images. The de-jittered images by our method is slightly superior to those results by the method in [25]. As the model (1.5) was designed under the assumption of Gaussian distribution on jitter s , the method [24] performs better on registering independent jitters than structured jitters (see e.g., the “Convex” and “Total” in recovered “Words” image by [24] are not left aligned).

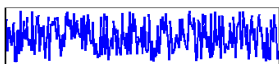
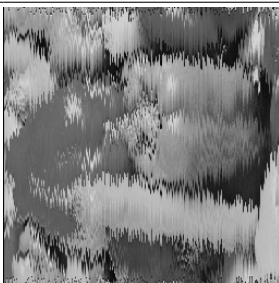




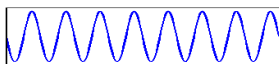
Independent jitter	Jitters and jittered image 16.24dB, 0.408	Kang & Shen [19] 23.54dB, 0.821	Lenzen & Scherzer [24] 25.61dB, 0.884	Nikolova [25] 30.13dB, 0.987	Ours 32.31dB, 0.992
	 				
Structured jitter	9.64dB, 0.522	10.76dB, 0.614	13.07dB, 0.783	14.28dB, 0.862	15.48dB, 0.886
	 <i>Convex Programming</i> <i>Total Variation Primal</i> <i>Dual Algorithm KKT</i> <i>Image DeJittering</i> <i>Augment Lagrangian</i> <i>Saddle Point Noise</i> <i>Free Proximal Point</i>	<i>Convex Programming</i> <i>Total Variation Primal</i> <i>Dual Algorithm KKT</i> <i>Image DeJittering</i> <i>Augment Lagrangian</i> <i>Saddle Point Noise</i> <i>Free Proximal Point</i>	<i>Convex Programming</i> <i>Total Variation Primal</i> <i>Dual Algorithm KKT</i> <i>Image DeJittering</i> <i>Augment Lagrangian</i> <i>Saddle Point Noise</i> <i>Free Proximal Point</i>	<i>Convex Programming</i> <i>Total Variation Primal</i> <i>Dual Algorithm KKT</i> <i>Image DeJittering</i> <i>Augment Lagrangian</i> <i>Saddle Point Noise</i> <i>Free Proximal Point</i>	<i>Convex Programming</i> <i>Total Variation Primal</i> <i>Dual Algorithm KKT</i> <i>Image DeJittering</i> <i>Augment Lagrangian</i> <i>Saddle Point Noise</i> <i>Free Proximal Point</i>

FIGURE 4. Noiseless jittered images recovery. Independent jitter: uniform distribution on $[-8, 8]$. Structured jitter: sinusoidal function $8 \sin(16\pi i)$.

5.2. Noisy jittered data. We now test the scenario of noisy jittered images. A noisy jittered image is produced by the image formation (1.2) with a certain jitter vector s , and then contaminated by additive white noise ε obeying the Gaussian distribution $\mathcal{N}(0, \mu^2)$.

We first test the performance of our approach on jittered images with different level of additive white noise, i.e., the Gaussian distribution $\mathcal{N}(0, \mu^2)$ with variant μ 's. The test image is “Peppers” image in Fig. 3. The ideal “Peppers” image is first degraded by an independent jitter uniformly distributed on $[-8, 8]$, and then contaminated by Gaussian noise $\mathcal{N}(0, \mu^2)$ with μ varying from 10 to 70. The noisy jittered “Peppers” images are shown in the top row of Fig. 5. The noisy jittered image u_0 is firstly smoothed as \tilde{u}_0 by a Gaussian filter with variance 0.2. We take the trade-off parameters $(\alpha_1, \alpha_2) = (1, 0.1)$ and the scalar $\delta = (16\|\nabla_x u_0\|_2^2 + \mu^2)^{1/2}$ for model (3.13). To implement Algorithm 1, we set the parameters as $(\rho, \tau, \sigma) = (1.5, 0.1, 10)$, the initial points $(\mathbf{x}^0, \mathbf{y}^0) = (\mathbf{0}, \mathbf{0})$ and the stopping tolerance $\varepsilon = 10^{-3}\|u_0\|_2$. Fig. 5 displays the dejittered images by our approach. Therein, the “Dejittered” row are the images \hat{u} produced by (4.14), whilst the “Denoised” row displays the output images after image denoising process on \hat{u} . Herein, the method in [18] is used as the image denoising solver. Fig. 5 shows that the proposed model performs well on dejittering noisy jittered images, even for the noise $\varepsilon \sim \mathcal{N}(0, \mu^2)$ with variance $\mu = 70$. When testing our approach, the numerical results show no difference with u_0 and \tilde{u}_0 as target jittered images in model (3.13).

We now test our approach on more noisy jittered images, compared with the methods in [25, 19, 24]. The test images are “Bird”, “Peppers”, “Boat” and “Baboon” images in Fig. 3. The noisy jittered images are listed in the first column of Fig. 6-7.

For our approach, we take the trade-off parameters $(\alpha_1, \alpha_2) = (1, 0.01)$ and the scalar $\delta = 20\|\nabla_x u_0\|_2$ for the proposed model (3.4). To implement Algorithm 1, we set the parameters as $(\rho, \tau, \sigma) = (1.8, 0.02, 1)$, the initial points $(\mathbf{x}^0, \mathbf{y}^0) = (\mathbf{0}, \mathbf{0})$ and the stopping tolerance $\varepsilon = 10^{-3}\|u_0\|_2$. For Nikolova’s method, we follow the procedures [25, Algorithm 3] to recover the noisy jittered images. The parameters are taken as $\alpha = 1$ and $\beta = 5$. The row-denoising step is realized by 1D wavelet thresholding with $T = 50$ (see also [10]) and the edge-preserving function ϕ is taken as the l^1 -norm. The obtained dejittered image is further denoised using image denoising solver in [18]. For the methods [19, 24], the parameters are tuned by trial-and-error to maximize the PSNR values of dejittered images. Figs. 6-7 illustrate the dejittered results by all test methods, which demonstrate the compelling performance of the proposed model.

6. CONCLUSION

In this paper, we focused on the problem of intrinsic image dejittering, which is a nontrivial task in image processing. A convex variational model was devised to register line jittered images (possible with noise). The proposed model can be easily solved by the first-order primal-dual algorithm. Preliminary numerical experiments demonstrated the compelling performance of our approach. As the line jitter can be extended to line-pixel jitter and pixel jitter, we will investigate efficient method for solving both settings as our pending work.


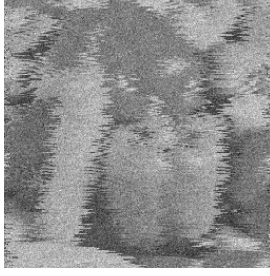
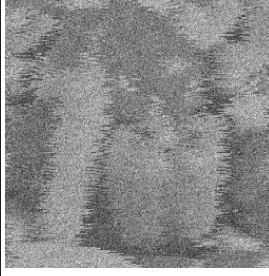
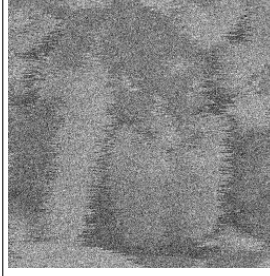
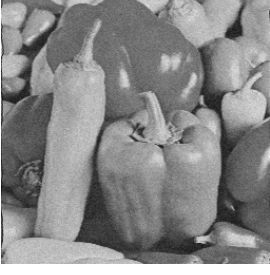
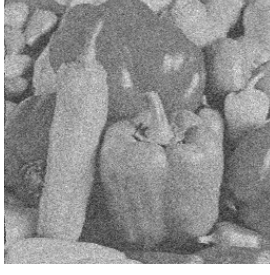
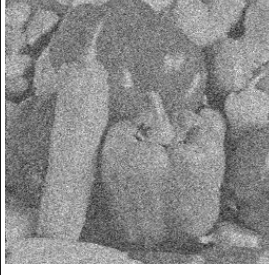
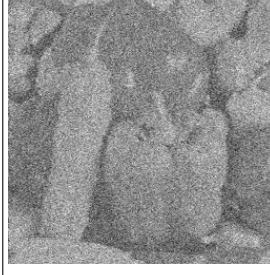




	$\mathcal{N}(0, 10^2)$	$\mathcal{N}(0, 30^2)$	$\mathcal{N}(0, 50^2)$	$\mathcal{N}(0, 70^2)$
Noisy, jittered	16.94dB, 0.267 	16.83dB, 0.109 	16.49dB, 0.061 	15.84dB, 0.038 
Dejittered				
Denoised	28.43dB, 0.888 	25.02dB, 0.785 	24.06dB, 0.724 	23.11dB, 0.661 

FIGURE 5. Performance of our method on dejittering noisy jittered image.

APPENDIX: THE FIRST- AND SECOND-ORDER DERIVATIVE OPERATORS

For an n_1 -by- n_2 image u , by defining the forward and backward differences as

$$(\nabla_x^+ u)_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < n_1, \\ u_{1,j} - u_{n_1,j} & \text{if } i = n_1, \end{cases} \quad (\nabla_y^+ u)_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < n_2, \\ u_{i,1} - u_{i,n_2} & \text{if } j = n_2, \end{cases}$$

and the backward differences

$$(\nabla_x^- u)_{i,j} = \begin{cases} u_{1,j} - u_{n_1,j} & \text{if } i = 1, \\ u_{i,j} - u_{i-1,j} & \text{if } i > 1, \end{cases} \quad (\nabla_y^- u)_{i,j} = \begin{cases} u_{i,1} - u_{i,n_2} & \text{if } j = 1, \\ u_{i,j} - u_{i,j-1} & \text{if } j > 1. \end{cases}$$

the operators in (4.3) can be specified as follows

$$\nabla_x = \nabla_x^+, \quad \nabla_y = \nabla_y^+,$$

$$\nabla_{xx} = \nabla_x^- \nabla_x^+, \quad \nabla_{yy} = \nabla_y^- \nabla_y^+, \quad \nabla_{xy} = \nabla_{yx} = \frac{1}{2}(\nabla_y^- \nabla_x^+ + \nabla_x^- \nabla_y^+).$$

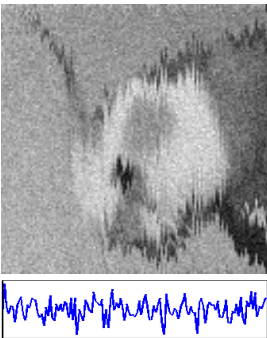




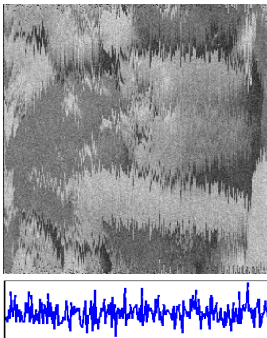




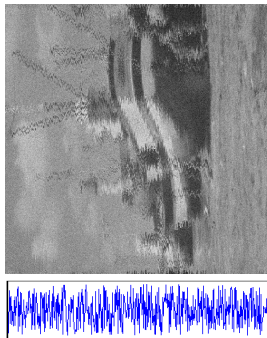




Noisy jittered image	Kang & Shen [19]	Lenzen & Scherzer [24]	Nikolova [25]	Ours
128 × 128 “Bird”, Gaussian jitter $\mathcal{N}(0, 3^2)$ and truncated on $[-8, 8]$, Gaussian noise $\mathcal{N}(0, 20^2)$				
20.60dB, 0.308 	26.89dB, 0.441 	27.17dB, 0.788 	27.25dB, 0.870 	29.47dB, 0.909 
256 × 256 “Peppers”, Binomial jitter in $[-14, 16]$, Gaussian noise $\mathcal{N}(0, 20^2)$				
17.06dB, 0.158 	21.20dB, 0.362 	21.94dB, 0.683 	22.08dB, 0.741 	23.40dB, 0.789 
512 × 512 “Boat”, Uniform jitter on $[-8, 8]$, Gaussian noise $\mathcal{N}(0, 15^2)$				
19.14dB, 0.171 	23.33dB, 0.322 	24.12dB, 0.543 	25.02dB, 0.711 	26.20dB, 0.750 

FIGURE 6. Performances of test methods on noisy jittered image recovery.

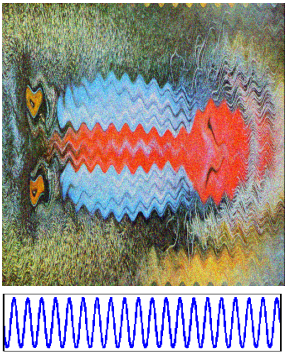




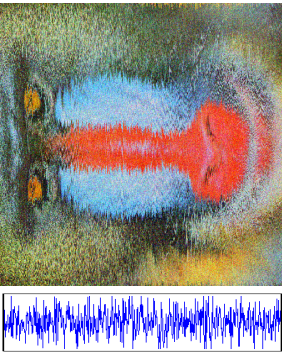




Noisy jittered image	Kang & Shen [19]	Lenzen & Scherzer [24]	Nikolova [25]	Ours
512×512 “Baboon”, Structured sinusoidal jitter $8 \sin(40\pi t)$, Gaussian noise $\mathcal{N}(0, 20^2)$ 18.45dB, 0.106 	19.49dB, 0.419 	18.52dB, 0.242 	21.03dB, 0.496 	23.07dB, 0.660 
512×512 “Baboon”, Gaussian jitter $\mathcal{N}(0, 6^2)$ and truncated on $[-12, 12]$, Gaussian noise $\mathcal{N}(0, 30^2)$ 18.31dB, 0.101 	19.17dB, 0.247 	19.73dB, 0.296 	20.01dB, 0.331 	22.31dB, 0.570 

FIGURE 7. Performance of test methods on RGB image dejittering.

Acknowledgments

The authors are grateful to Prof. Michael K. Ng (The University of Hong Kong), Prof. Pierre Weiss (CNRS-ITAV, Université de Toulouse) for fruitful suggestions, and Prof. Sung Ha Kang (School of Mathematics, Georgia Institute of Technology) for providing us with her MATLAB codes of reference [19]. This paper was supported by the NSFC (No. 11971003) and the Fundamental Research Funds for the Central Universities (No. ZYGX1029J090).

REFERENCES

- [1] H. Attouch, G. Buttazzo, G. Michaille, *Variational Analysis in Sobolev and BV Spaces: Applications to PDEs and Optimization*, SIAM and MPS, 2006.
- [2] H.H. Bauschke, P.L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York, 2011.
- [3] D. Boley, Local linear convergence of the alternating direction method of multipliers on quadratic or linear programs, *SIAM J. Optim.* 23 (2013), 2183-2207.
- [4] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, ‘Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Machine Learning* 3 (2010), 1-122.
- [5] J.F. Cai, R.H. Chan, Z.W. Shen, A framelet-based image inpainting algorithm, *Appl. Comput. Harmon. Anal.* 24 (2008), 131-149.
- [6] A. Chambolle, T. Pock, A first-order primal-dual algorithm for convex problems with applications to imaging, *J. Math. Imaging Vision*, 40 (2012), 120-145.
- [7] A. Chambolle, T. Pock, On the ergodic convergence rates of a first-order primal-dual algorithm, *Math. Program.* 159 (2016), 253-287.
- [8] R.H. Chan, M. Tao, X.M. Yuan, Constrained total variational deblurring models and fast algorithms based on alternating direction method of multipliers *SIAM J. Imag. Sci.* 6 (2013), 680-697.
- [9] L. Condat, Fast projection onto the simplex and the l_1 ball, *Math. Program.* 158 (2016), 575-585.
- [10] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, 1992.
- [11] G. Dong, A.R. Patrone, O. Scherzer, O. Öktem, Infinite dimensional optimization models and PDEs for deblurring, *Scale Space and Variational Methods in Computer Vision*, pp. 678-689, 2015.
- [12] G. Dong, O. Scherzer, Nonlinear flows for displacement correction and applications in tomography, *Scale Space and Variational Methods in Computer Vision*, pp. 283-294, 2017.
- [13] J. Duchi, S. Gould, D. Koller, Projected subgradient methods for learning sparse Gaussian, *Conference on Uncertainty in Artificial Intelligence*, 2008.
- [14] E. Esser, X. Zhang, T. F. Chan, A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science, *SIAM J. Imaging Sci.* 3 (2010), 1015-1046.
- [15] R. Glowinski, On alternating direction methods of multipliers: a historical perspective, In: *Modeling, Simulation and Optimization for Science and Technology Computational Methods in Applied Sciences*, vol. 34, pp. 59-82, 2014.
- [16] D.R. Han, X.M. Yuan, Local linear convergence of the alternating direction method of multipliers for quadratic programs, *SIAM J. Numer. Anal.* 51 (2013), 3446-3457.
- [17] B.S. He, Y.F. You, X.M. Yuan, On the convergence of primal-dual hybrid gradient algorithm, *SIAM J. Imaging Sci.* 7 (2014), 2526-2537.
- [18] B.S. He, X.M. Yuan, Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective, *SIAM J. Imaging Sci.* 5 (2012), 119-149.
- [19] S.H. Kang, J. Shen, Video deblurring by bake and shake, *Image Vision Comput.* 24 (2006), 143-152.
- [20] S.H. Kang, J. Shen, *Image deblurring based on slicing moments*, Springer Series on Mathematics and Visualization, pp. 35-55, 2007.
- [21] A. Kokaram, *Motion Picture Restoration*, Springer, 1998.
- [22] A. Kokaram, P.M. Roosmalen, P. Rayner, J. Biemond, Line registration of jittered video, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2553-2556, 1997.

- [23] L. Laborelli, Removal of video line jitter using a dynamic programming approach, In: Proceedings of IEEE ICIP, pp. 331-334, 2003.
- [24] F. Lenzen, O. Scherzer, Partial differential equations for zooming, deinterlacing and dejittering, *Int. J. Comput. Vision* 92 (2011), 162-176.
- [25] M. Nikolova, One-iteration dejittering of digital video images, *J. Vis. Commun. Image R.* 20 (2009), 254-274.
- [26] M.K. Ng, H.Y. Ngan, X.M. Yuan, W.X. Zhang, Lattice-based patterned fabric inspection by using total variation and sparsity with low-rank representations, *SIAM J. Imaging Sci.* 10 (2017), 2140-2164.
- [27] D. O'Connor, L. Vandenberghe, Primal-dual decomposition by operator splitting and applications to image deblurring, *SIAM J. Imaging Sci.* 7 (2015), 1724-1754.
- [28] N. Parikh, S. Boyd, Proximal algorithms, *Found. Trends Optim.* 1 (2013), 123-231.
- [29] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, *IEEE Trans. Pattern Anal. Mach. Intell.* 12 (1990), 629-639.
- [30] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, F. Lenzen, *Variational Methods in Imaging*, Springer, 2009.
- [31] J. Shen, Bayesian video dejittering by the BV image model, *SIAM J. Appl. Math.* 64 (2004), 1691-1708.
- [32] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: From error visibility to structural similarity, *IEEE Trans. Image Process* 13 (2004), 600-612.
- [33] J.F. Yang, Y. Zhang, W.T. Yin, An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise, *SIAM J. Sci. Comput.* 31 (2009), 2842-2865.