

# An Efficient Primal-Dual Hybrid Gradient Algorithm For Total Variation Image Restoration

Mingqiang Zhu      Tony Chan

## Abstract

We propose a simple yet efficient algorithm for total variation (TV) minimizations with applications in the image processing realm. This descent-type algorithm alternates between the primal and dual formulations and exploit the information from both the primal and dual variables. It converges significantly faster than some popular existing methods as demonstrated in our experiments. This approach is to some extent related to projection type methods for solving variational inequalities. It can be applied to solve other TV model and  $L^1$  minimization problem.

## 1 Introduction

### 1.1 Background

Variational models have been extremely successful in a wide variety of image restoration problems, and remain one of the most active areas of research in mathematical image processing and computer vision. The most fundamental image restoration problem is perhaps denoising. It forms a significant preliminary step in many machine vision tasks such as object detection and recognition. Total variation based image restoration models were first introduced by Rudin, Osher, and Fatemi(ROF) in their pioneering work [17]. It was designed with the explicit goal of preserving sharp discontinuities (edges) in an image while removing noise and other unwanted fine scale detail. It was formulated as the following minimization problem.

$$\min_u \int_{\Omega} |\nabla u| \quad \text{s.t.} \quad \|u - f\|_2^2 \leq |\Omega| \sigma^2. \quad (1)$$

Here,  $\Omega$  denotes the image domain with its area being  $|\Omega|$ , the function  $f : \Omega \rightarrow \mathbb{R}$  represents the given observed image and  $\sigma^2$  is an estimate of the variance of the noise in the image  $f$ .

Rather than solving the constrained minimization problem (1), ROF and sub-sequent researchers also considered the unconstrained minimization

problem which uses TV as a Tikhonov regularization

$$\min_u \int_{\Omega} |\nabla u| + \frac{\lambda}{2} \|u - f\|_2^2 \quad (2)$$

The above problem (2) yields the same solution as (1) for a suitable choice of the Lagrange multiplier  $\lambda$  (see [5]).

More recently, the dual formulation of unconstrained ROF model (2) has also been studied and numerical algorithms are proposed correspondingly (see [6], [2], [4]). It was shown that the desired clean image  $u^*$ , or the global minimizer of primal ROF (2), can be obtained by

$$u^* = f + \frac{1}{\lambda} \nabla \cdot w^*, \quad (3)$$

where  $w^*$  solves the following dual problem

$$\min_{|w| \leq 1} \|\nabla \cdot w + \lambda f\|_2^2. \quad (4)$$

From the computational point of view, the primal and dual formulations pose different challenges for computing their optimality solutions (see Table 1). The total variation term in the primal formulation is non-smooth at where  $|\nabla u| = 0$ , which makes the derivative-based methods impossible without an artificial smoothing parameter. The dual formulation imposes constraints which usually require extra effort compared to unconstrained optimizations. Being quadratic, the dual energy is less nonlinear than the primal energy, but the rank-deficient operator  $\nabla \cdot$  makes the dual minimizers possibly non-unique. Finally, they share the same problem of spatial stiffness due to the global couplings in their energy functions, which presents a challenge to any algorithm in order to control the computational complexity that scales reasonably bounded with the number of pixels.

| Primal Problem (2)                      | Dual Problem (4)                                      |
|---|---|
| · Nondifferentiable at $ \nabla u  = 0$ | · Non-uniqueness due to rank-deficient $\nabla \cdot$ |
| · Highly nonlinear                      | · Extra constraints                                   |
| · Spatial stiffness                     | · Spatial stiffness                                   |

Table 1: Computational Challenges for Primal and Dual ROF

Over the years, the ROF model has been extended to many other image processing tasks, including inpaintings, blind-deconvolutions, cartoon-texture decompositions and vector valued images. It has also been modified in a variety of ways to improve its performance. (Interested reader can see [3] and the reference therein for recent developments in TV based image processing.) In this paper, we shall focus on solving the original ROF restoration problem, but we point out that our idea can be naturally extended to other relevant models.

## 1.2 Existing Algorithms

A great amount of algorithms have been developed, aiming to solve efficiently either the primal formulation (2) or the dual formulation (4) of the ROF model. We shall review some of the numerical methods here and refer the interested readers to [7] for a more comprehensive survey.

In their original paper [17], the authors proposed a time marching scheme to solves the associated Euler-Lagrange equation of (2). The method is very slow due to the CFL stability constraints (see [16]). In fact the time step constraint is near zero when the solution develop flat regions(where  $|\nabla u| \approx 0$ ). Hence this scheme is only of practical use to provide low-accurate solutions, and sometimes it may even require substantial cost to just compute a visually satisfactory image. In [18], Vogel etc. proposed to solve the same Euler-Lagrange equation of (2) via a fixed point iteration method. The method, requiring to solve a linear system at each iteration, is proved to be global convergent and asymptotically much faster than the explicit time marching scheme.

The idea of duality was first introduced in [6], where the authors applied Newton's method to solve the primal-dual system of the ROF model. The primal-dual Newton or CGM method was shown to have a fast locally quadratic convergence rate. Therefore, it can be used to generate high-accurate benchmark solutions.

The dual problem (4) alone was tackled by Chambolle in [4], where he proposed a semi-implicit gradient descent algorithm based on some original observation he made on the Lagrange multipliers. The method is global convergent with suitable stepsize. It has become popular for its simplicity and fast convergence to medium-accurate visually satisfactory solutions.

Goldfarb and Yin proposed in [8] a unified approach that reformulates both the primal model (1) and dual model (4) to Second-Order Cone Programming(SOCP) and solved them by corresponding algorithms (i.e. primal-dual interior-point method with MOSEK package). This approach essentially need to solve the update direction at each step by Newton's linearization and hence has the similar convergence properties to the CGM method.

More recently, Wang etc. introduced a model [19] that splits the TV term into two parts and hereby constructed an iterative procedure of alternately solving a pair of easy subproblems associated with an increasing sequence of penalty parameter values. Their algorithm focus on solving the more general denoising/deblurring problem. Later, Goldstein and Osher [9] proposed a method based on the same splitting technique but uses Bregman distance to deal with the artificial constraints. Both of these methods were shown to converge very fast to a visually satisfactory solution.

### 1.3 Notations and Discretizations

From now on, and until the end of the paper, we will restrict our attention to the discrete setting. Let us fix our main notations before get into numerical algorithms.

For the sake of simplicity, We assume that the image domain  $\Omega$  is a square, and define a regular  $n \times n$  grid of pixels, indexed as  $(i, j)$ , for  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, n$ . We represent images as two-dimensional matrices of dimension  $n \times n$ , where  $\mathbf{u}_{i,j}$  represents the value of the function  $u$  at pixel  $(i, j)$ . (Adaption to less regular domains is not difficult in principle.) To define the discrete total variation, we introduce a discrete gradient operator, whose two components at each pixel  $(i, j)$  are defined as follows:

$$\begin{aligned} (\nabla u)_{i,j}^1 &= \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < n \\ 0 & \text{if } i = n \end{cases} \\ (\nabla u)_{i,j}^2 &= \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < n \\ 0 & \text{if } j = n \end{cases} \end{aligned}$$

(Thus  $\nabla u \in \mathbb{R}^{n \times n \times 2}$ .) The discrete TV of  $u$  is then defined by

$$\text{TV}(u) = \sum_{1 \leq i,j \leq n} \|(\nabla u)_{i,j}\|$$

where  $\|\cdot\|$  is the Euclidean ( $\ell_2$ ) norm in  $\mathbb{R}^m$  (in this case,  $\mathbb{R}^2$ ), and we shall use this notation for the rest of the paper. Note that this norm is not a smooth function of its argument.

To describe the problem in matrix algebra language, we reorder the image matrix  $u$  (resp.  $f$ ) row-wisely into a vector  $y$  (resp.  $z$ ), associating the  $(i, j)$  element of the two-dimensional structure with the element  $(j-1)n+i$  of the vector structure, as follows:

$$y_{(j-1)n+i} = u_{i,j}, \quad 1 \leq i, j \leq n.$$

We have  $y \in \mathbb{R}^N$ , where  $N = n^2$ . The  $(i, j)$  component of the gradient  $\nabla u$  can thus be represented as a multiplication of the vector  $y \in \mathbb{R}^N$  by a matrix  $A_k \in \mathbb{R}^{N \times 2}$ , for  $k = 1, 2, \dots, N$ :

$$A_l^T y = \begin{cases} (y_{l+1} - y_l, y_{l+n} - y_l)^T & \text{if } l \bmod n \neq 0 \text{ and } l \leq N-n \\ (0, y_{l+n} - y_l)^T & \text{if } l \bmod n = 0 \text{ and } l \leq N-n \\ (y_{l+1} - y_l, 0)^T & \text{if } l \bmod n \neq 0 \text{ and } l > N-n \\ (0, 0)^T & \text{if } l \bmod n = 0 \text{ and } l > N-n. \end{cases}$$

Using this notation, the discrete version of the primal ROF model (2) can be written as

$$\min_y P(y) := \sum_{l=1}^N \|A_l^T y\| + \frac{\lambda}{2} \|y - z\|^2, \quad (5)$$

## 1.4 Duality

As we see in the previous part of this paper, duality theory plays an important role in TV based image restoration models, especially in developing fast numerical algorithms to compute their optimality solutions. In this section, we discuss about duality, and first we shall derive the dual formulation of the discrete primal ROF model (5).

For any vector  $b$ , we have

$$\|a\| = \max_{\|b\| \leq 1} a^T b, \quad (6)$$

which is a direct consequence from the Cauchy-Schwartz inequality.

Using the above equation, the discrete TV of  $y$  can be written as:

$$\begin{aligned} \sum_{l=1}^N \|A_l^T y\| &= \max_{\|x_l\| \leq 1} \sum_{l=1}^N (A_l^T y)^T x_l \\ &= \max_{\|x_l\| \leq 1} y^T \sum_{l=1}^N A_l x_l \\ &= \max_{x \in X} y^T Ax, \end{aligned} \quad (7)$$

where

$$A = [A_1, A_2, \dots, A_N] \in \mathbb{R}^{N \times 2N}, \quad x_l = \begin{bmatrix} x_l^1 \\ x_l^2 \end{bmatrix} \in \mathbb{R}^2, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^{2N}. \\ X = \{x : x \in \mathbb{R}^{2N}, \|x_l\| \leq 1 \text{ for } l = 1, 2, \dots, N\} \quad (8)$$

From equation (7), we can reform the primal ROF model (5) to the following min-max or max-min problem:

$$\min_y \max_{x \in X} \Phi(y, x) := y^T Ax + \frac{\lambda}{2} \|y - z\|^2 \quad (9a)$$

$$= \max_{x \in X} \min_y y^T Ax + \frac{\lambda}{2} \|y - z\|^2, \quad (9b)$$

where the equality follows from the min-max theorem (see [11, Chapter VII, Theorem 4.3.1]).

The inner minimization problem in (9b) can be solved exactly as

$$y = z - \frac{1}{\lambda} Ax. \quad (10)$$

Substituting (10) back to (9b) yields the following dual problem

$$\max_{x \in X} D(x) := \frac{\lambda}{2} \left[ \|z\|^2 - \left\| \frac{1}{\lambda} Ax - z \right\|^2 \right], \quad (11)$$

or equivalently

$$\min_{x \in X} \|Ax - \lambda z\|^2. \quad (12)$$

Problem (12) is the discrete counterpart of problem (4). (The matrix  $A$  is the discrete version of the negative divergence operator  $-\nabla \cdot$  and  $x$  is the discrete dual variable.)

For primal-dual feasible pair  $(y, x)$ , the duality gap  $G(y, x)$  is defined as the difference between the primal and dual objectives:

$$\begin{aligned} G(y, x) &= P(y) - D(x) \\ &= \sum_{l=1}^N \|A_l^T y\| + \frac{\lambda}{2} \|y - z\|^2 - \frac{\lambda}{2} \left[ \|z\|^2 - \left\| \frac{1}{\lambda} Ax - z \right\|^2 \right] \\ &= \sum_{l=1}^N \left( \|A_l^T y\| - x_l^T A_l^T y \right) + \frac{\lambda}{2} \left\| y - z + \frac{1}{\lambda} Ax \right\|^2 \end{aligned} \quad (13)$$

The duality gap is a measure of closeness of the primal-dual pair  $(y, x)$  to the primal-dual solution, and also bounds the differences  $P(y) - P^*$  and  $D^* - D(x)$ , where  $P^* = D^*$  is the (common) optimal objective value for the primal and dual problems, respectively. Therefore,  $G(y, x)$  can be useful in designing the stopping criterion for a numerical algorithm.

If  $y$  and  $x$  are feasible, then it is clear from (13) that  $G(y, x) \geq 0$ , and that  $G(y, x) = 0$  iff.

$$\begin{cases} \|A_l^T y\| - x_l^T A_l^T y = 0 & \text{for } l = 1, \dots, N \\ y - z + \frac{1}{\lambda} Ax = 0 \end{cases} \quad (14)$$

which is also equivalent to

$$\begin{cases} \|A_l^T y\| x_l - A_l^T y = 0 & \text{for } l = 1, \dots, N \\ y - z + \frac{1}{\lambda} Ax = 0 \end{cases} \quad (15)$$

(15) is referred as the *primal-dual optimality system*.

## 2 Primal-Dual hybrid gradient Algorithm

### 2.1 Motivations

Most existing numerical algorithms to solve ROF models (5) or (11) can be loosely divided into two categories: those that need to solve a linear system

of equations at each iteration (implicit) and those that require only a matrix-vector multiplication in the discrete setting (explicit). Generally speaking, the implicit methods(e.g. CGM and SOCP) have fast asymptotical convergence rate and can provide highly accurate benchmark solutions. However, explicit methods are preferred in many situations for their simplicity and their fast initial convergence to medium-accurate and visually satisfactory results. Their low memory requirements make them even more attractive to large-scale problems. As an example for the high memory requirement of implicit schemes, we note that an image of size  $512 \times 512$  is close to the limit of what the SOCP solver Mosek can handle on a workstation with 2GB of memory.

In this paper, we shall develop some simple yet efficient algorithms. They are explicit so the memory requirement is low and each iteration only takes  $O(N)$  operations. They converge very fast to visually satisfactory solutions and also have much improved asymptotical convergence rate compared with existing explicit methods. Our proposed algorithms shall also exploit some use of the dual variable since a pure primal formulation usually requires a numerical smoothing parameter that would prevent the resulting algorithm from converging to the true optimizer.

Previous developed gradient descent type methods are the primal time marching method in [17] and Chambolle's duality based semi-implicit gradient descent type method in [4]. In this chapter, we refer to these two methods as the *primal gradient descent* algorithm and *dual gradient descent* algorithm. They are showed briefly as follows.

Primal ROF:

$$\min_{y \in \mathbb{R}^N} \sum_{l=1}^N \|A_l^T y\| + \frac{\lambda}{2} \|y - z\|^2, \quad (16)$$

*Primal gradient descent* algorithm (smoothed with  $\beta$ ):

$$y^{k+1} = y^k - \theta_k \left( \frac{1}{\lambda} \sum_{l=1}^N \frac{A_l A_l^T y^k}{\sqrt{\|A_l^T y^k\|^2 + \beta}} + y^k - z \right), \quad (17)$$

*Primal gradient descent* algorithm (unsmoothed subgradient):

$$y^{k+1} = y^k - \theta_k \left( \frac{1}{\lambda} \sum_{l=1}^N A_l x_l^k + y^k - z \right), \quad (18)$$

where

$$x_l^k = \begin{cases} \frac{A_l^T y_l^k}{\|A_l^T y_l^k\|}, & \text{if } A_l^T y_l^k \neq 0 \\ \text{any element in the unit ball } B(0, 1) \subset \mathbb{R}^2, & \text{else} \end{cases} \quad (19)$$

Dual ROF:

$$\max_{x \in X} \|Ax - \lambda z\|^2$$

where  $X = \{x : x \in \mathbb{R}^{2N}, \|x_l\| \leq 1 \text{ for } l = 1, 2, \dots, N\}$  (20)

*Dual gradient descent* algorithm (Chambolle):

$$x_l^{k+1} = \frac{x_l^k - \tau_k A_l^T (Ax^k - \lambda z)}{1 + \tau_k \|A_l^T (Ax^k - \lambda z)\|}. \quad (21)$$

We realize that the above methods are based exclusively either on the primal formulation (5) or the dual formulation (11). Our approach is, however, to unify the primal and dual formulations into one framework and develop a gradient descent type method based on both formulations. Therefore, at each step, the updates will exploit the information in both the primal and dual states and we expect it to improve the convergence speed. We notice that the primal problem and the dual problem has different computational challenges. The primal problem is difficult to solve and has slow convergence at the nonsmooth pixels where  $\|A_i^T y\| = 0$ . The dual problem is difficult to solve and has slow convergence at the pixels where constraints are active :  $\|x_i\| = 1$ . These two difficulties are almost exclusive (not strictly exclusive, which corresponding to the strict complementarity condition of the dual problem), hence by combining these two formulation into one algorithm framework, we might be able to solve each difficulty using the help from the other.

## 2.2 The Proposed Algorithm

Our approach can be most effectively illustrated under the setting of the primal-dual formulation (9), which we rewrite here:

$$\min_{y \in \mathbb{R}^N} \max_{x \in X} \Phi(y, x) := y^T Ax + \frac{\lambda}{2} \|y - z\|^2$$

Given any intermediate solution  $(y^k, x^k)$  at iteration step  $k$ , the proposed algorithm updates the solution as follows.

### 1. DUAL STEP

Fix  $y = y^k$ , apply one step of (projected) gradient ascent method to the maximization problem

$$\max_{x \in X} \Phi(y^k, x). \quad (22)$$

The ascent direction  $\nabla_x \Phi(y^k, x) = A^T y^k$ , so we update  $x$  as

$$x^{k+1} = P_X(x^k + \tau_k \lambda A^T y^k), \quad (23)$$

where  $\tau_k$  is the (dual) stepsize and  $P_X$  denotes the projection onto the set  $X$ :

$$P_X(z) = \arg \min_{x \in X} \|z - x\|.$$

The projection can be simply computed in our case (see remark 2 in section 2.3). The factor  $\lambda$  is used in (23) so that the stepsize  $\tau_k$  will not be sensitive to different problems or scales of gray levels, which also explains the same situation in (25).

## 2. PRIMAL STEP

Fix  $x = x^{k+1}$ , apply one step of gradient descent method to the minimization problem

$$\min_{y \in \mathbb{R}^N} \Phi(y, x^{k+1}). \quad (24)$$

The ascent direction is  $\nabla_y \Phi(y, x^{k+1}) = Ax^{k+1} + \lambda(y^k - z)$  and therefore the update is

$$y^{k+1} = y^k - \theta_k \left( \frac{1}{\lambda} Ax^{k+1} + y^k - z \right), \quad (25)$$

where  $\theta_k$  is the (primal) stepsize.

Put them all together, we have the following algorithm.

### Algorithm PDHG

**Step 0.** *Initialization.* Pick  $y^0$  and a feasible  $x^0 \in X$ , set  $k \leftarrow 0$ .

**Step 1.** Choose stepsize  $\tau_k$  and  $\theta_k$ .

**Step 2.** *Updating.*

$$x^{k+1} = P_X(x^k + \tau_k \lambda A^T y^k) \quad (26a)$$

$$y^{k+1} = (1 - \theta_k)y^k + \theta_k(z - \frac{1}{\lambda}Ax^{k+1}) \quad (26b)$$

**Step 3.** Terminate if a stopping criterion is satisfied; otherwise set  $k \leftarrow k + 1$  and return to step 1.

The above algorithm is very simple. Notice here  $A^T y$  and  $Ax$  are just  $\nabla u$  and  $-\nabla \cdot w$  and can be computed efficiently using difference instead of matrix-vector multiplication.

### 2.3 Remarks

1. The hybrid gradient algorithm (26) can also be developed as a (primal-dual) proximal-point method:

$$x^{k+1} = \arg \max_{x \in X} \Phi(y^k, x) - \frac{1}{2\lambda\tau_k} \|x - x^k\|^2 \quad (27a)$$

$$y^{k+1} = \arg \min_{y \in \mathbb{R}^N} \Phi(y, x^{k+1}) + \frac{\lambda(1 - \theta_k)}{2\theta_k} \|y - y^k\|^2 \quad (27b)$$

The idea here is that when using dual variable to update the primal variable, since the dual variable is not optimal yet, we do not want to solve the primal minimization problem exactly, instead, we add a penalty term to force the new update close to the previous value, vice versa for the dual step.

2. The projection  $P_X$  in (27a) can be computed in the following straightforward way:

$$(P_X(x))_l = \frac{x_l}{\max\{\|x_l\|, 1\}}, \quad l = 1, 2, \dots, N. \quad (28)$$

Here, since  $X$  is a Cartesian product of unit Euclidean balls, the above operation (28) actually projects each  $2 \times 1$  subvector of  $x$  separately onto the unit ball in  $\mathbb{R}^2$ .

3. Both problem (22) and (24) can be solved exactly, which would yield the following updating formula (taking  $\tau_k = \infty$  and  $\theta_k = 1$  in (26)):

$$x_l^{k+1} = \frac{A_l^T y^k}{\|A_l^T y^k\|}, \quad \text{for } l = 1, \dots, N \quad (29a)$$

$$y^{k+1} = z - \frac{1}{\lambda} A x^{k+1}. \quad (29b)$$

However, we choose not to do so since the above algorithm does not converge.

As a special case, if we only solve subproblem (22) exactly (taking  $\tau_k = \infty$  in (26)), the resulting algorithm would be

$$y^{k+1} = y^k - \theta_k \left( \frac{1}{\lambda} \sum \frac{A_l A_l^T y^k}{\|A_l^T y^k\|} + y^k - z \right), \quad (30)$$

The above algorithm is exactly a subgradient descent method for the primal formulation (5).

Another special case is that we solve subproblem (24) exactly (taking  $\theta_k = 1$  in (26)) and still apply gradient ascent method to (22). The resulting algorithm is as follows

$$P_X \left( x^k - \tau_k A^T (Ax^k - \lambda z) \right), \quad (31)$$

which is a (projected) gradient descent method for dual problem (12).

Hence, the *primal subgradient descent method* and the *dual projected gradient descent method* are two special cases of our algorithm, which correspond to taking special stepsizes  $\tau_k = \infty$  and  $\theta_k = 1$  respectively in (26).

4. The convergence of the PDHG is (empirically) observed for a variety of suitable stepsize pairs  $(\tau, \theta)$ . Moreover, the choice of stepsizes are insensitive to different problems or scales of gray levels. In some range, the stability constraint on the stepsizes seems more relevant to the product  $\tau\theta$  rather than to  $\tau$  or  $\theta$  individually. For example, our experiments show that convergence can be obtained for the stepsize pair  $(2, 0.2)$  as well as for  $(4, 0.1)$ . Finally, the numerical results also reveal that a pair of relatively small  $\tau$  and large  $\theta$  gives faster initial convergence rate and the opposite choice gives faster asymptotic convergence. Therefor, we can optimize the performance of the algorithm through some strategy of choosing  $(\tau_k, \theta_k)$ , although simple fixed stepsizes might already give satisfactory results.

## 2.4 Extensions

Our algorithm can be naturally extended to other TV image restoration models without any major modifications.

### 2.4.1 Constrained ROF Model

The original constrained ROF model (1) has the following discrete form

$$(\mathbf{P}) \quad \min_{y \in Y} \sum_{l=1}^N \|A_l^T y\|, \quad (32)$$

where  $Y \equiv \{y \in \mathbb{R}^N : \|y - z\| \leq n\sigma\}$ .

Using the same idea in section (1.4), we can obtain the primal-dual and dual formulation of (32) as follows

$$(\mathbf{PD}) \quad \min_{y \in Y} \max_{x \in X} y^T A x \quad (33)$$

$$(\mathbf{D}) \quad \max_{x \in X} -n\sigma \|Ax\| + z^T A x \quad (34)$$

Both the primal problem (32) and dual problem (34) are difficult to solve since their objectives are non-smooth. However, our proposed approach based on the primal-dual formulation (33) are simply

$$x^{k+1} = P_X \left( x^k + \frac{\tau_k}{\sigma} A^T y^k \right) \quad (35a)$$

$$y^{k+1} = P_Y \left( y^k - \sigma \theta_k A x^{k+1} \right), \quad (35b)$$

where the projection  $P_Y$  is given by

$$P_Y(y) = z + \frac{y - z}{\max \{\|y - z\|/(n\sigma), 1\}} \quad (36)$$

The full *primal-dual hybrid gradient method* for *constrained ROF model* is shown as follows:

Algorithm PDHG-C

**Step 0.** *Initialization.* Pick  $y^0$  and a feasible  $x^0 \in X$ , set  $k \leftarrow 0$ .

**Step 1.** Choose stepsize  $\tau_k$  and  $\theta_k$ .

**Step 2.** *Updating.*

$$x^{k+1} = P_X(x^k + \frac{\tau_k}{\sigma} A^T y^k) \quad (37a)$$

$$y^{k+1} = P_Y(y^k - \sigma \theta_k A x^{k+1}), \quad (37b)$$

**Step 3.** Terminate if a stopping criterion is satisfied; otherwise set  $k \leftarrow k + 1$  and return to step 1.

#### 2.4.2 TV Deblurring Model

The total variation based image restoration model (2) can be extended to recover blurry and noisy image  $f$  by solving the following problem:

$$(\mathbf{P}) \quad \min_u \int_{\Omega} |\nabla u| + \frac{\lambda}{2} \|Ku - f\|_2^2 \quad (38)$$

where  $K$  is a given linear blurring operator and every other term is defined the same as in (2). In this model,  $f$  is formulated as the sum of a Gaussian noise  $v$  and a blurry image  $Ku$  resulting from the linear blurring operator  $K$  acting on the clean image  $\bar{u}$ , i.e.,  $f = Ku + v$ .

Among all linear blurring operators, many are shift-invariant and can be expressed in the form of convolution:

$$(Ku)(x) = (h * u)(x) = \int_{\Omega} h(x - y) u(y) dy, \quad (39)$$

where  $h$  is the given *point spread function* (PSF) associated with  $K$ .

The discrete form of model (38) is

$$\min_y \sum_{l=1}^N \|A_l^T y\| + \frac{\lambda}{2} \|By - z\|^2, \quad (40)$$

where  $B$  is the discretization of the blurring operator  $K$ .

The primal-dual and dual formulation of (40) can be obtained in the same way as in section (1.4) and are shown as follows

$$(\mathbf{PD}) \quad \min_y \max_{x \in X} y^T A x + \frac{\lambda}{2} \|By - z\|^2 \quad (41)$$

$$(\mathbf{D}) \quad \max_{x \in X} -\frac{1}{2\lambda} \|B^{-1} A x - \lambda z\| + \frac{\lambda}{2} \|z\|^2. \quad (42)$$

However, the blurring matrix  $B$  is highly ill-posed (non-invertible in some cases), making it difficult if not impossible to compute the inverse  $B^{-1}$ . Therefore the dual formulation (42) is of little use in practice and sometimes may not even exist. On the other hand, our *primal-dual hybrid gradient descent* algorithm based on formulation (41) still works well. The core part of the algorithm are given as follows

$$x^{k+1} = P_X(x^k + \tau_k A^T y^k) \quad (43a)$$

$$y^{k+1} = y^k - \theta_k (A x^{k+1} + \lambda B^T (By^{k+1} - z)). \quad (43b)$$

Note the primal update step (43b) here is semi-implicit. The motivation here is that since  $B$  is ill-posed, the explicit gradient descent will have slow asymptotic convergence. Since  $B$  is the matrix representation of a convolution operator  $K$ , the Fourier transform of matrix multiplication by  $B$  becomes point-wise multiplication in the frequency domain. Hence, step (43b) can be efficiently solved by FFT and inverse FFT:

$$y^{k+1} = \mathcal{F}^{-1} \left[ \frac{\mathcal{F}(y^k - \theta_k A x^{k+1}) + \theta_k \lambda \mathcal{F}(K)^* \odot \mathcal{F}(z)}{1 + \theta_k \lambda \mathcal{F}(K)^* \odot \mathcal{F}(K)} \right], \quad (44)$$

where  $\mathcal{F}(\cdot)$  and  $\mathcal{F}^{-1}(\cdot)$  are FFT and inverse FFT operators,  $*$  denotes the complex conjugate and  $\odot$  is the pointwise multiplication operator.

The full *hybrid gradient descent* method for deblurring (PDHG-D) is defined as follows:

#### Algorithm PDHG-D

**Step 0.** *Initialization.* Pick  $y^0$  and a feasible  $x^0 \in X$ , set  $k \leftarrow 0$ .

**Step 1.** Choose stepsize  $\tau_k$  and  $\theta_k$ .

**Step 2.** *Updating.*

$$x^{k+1} = P_X(x^k + \tau_k A^T y^k) \quad (45a)$$

$$y^{k+1} = \mathcal{F}^{-1} \left[ \frac{\mathcal{F}(y^k - \theta_k A x^{k+1}) + \theta_k \lambda \mathcal{F}(K)^* \odot \mathcal{F}(z)}{1 + \theta_k \lambda \mathcal{F}(K)^* \odot \mathcal{F}(K)} \right] \quad (45b)$$

**Step 3.** Terminate if a stopping criterion is satisfied; otherwise set  $k \leftarrow k + 1$  and return to step 1.

### 3 Theoretical Connections

Our method is related to projection type methods existing in the literature for finding saddle points and, more generally, solutions to variational inequalities. In this section, we shall discuss very briefly about the framework of projection methods for solving variational inequalities and point out the connections and difference between our method and previous work. We refer interested readers to the survey papers [10] and [20] for the background of this area.

Let  $H$  be a real Hilbert space (in our case,  $\mathbb{R}^n$ ), whose inner product and norm are denoted by  $\langle \cdot, \cdot \rangle$ , and  $\|\cdot\|$  respectively. Let  $K$  be a closed convex set in  $H$  and  $F$  be a mapping from  $H$  into itself. We now consider the problem of finding  $v^* \in K$  such that

$$\langle v - v^*, F(v^*) \rangle \geq 0, \quad \forall v \in K. \quad (46)$$

The above problem is called a *variational inequality* problem with  $v^*$  being one of its solution. We denote the above variational inequality problem by  $\text{VI}(K, F)$ . In most real applications,  $K$  is convex and  $F$  satisfy some monotonicity and Lipschitz continuity properties, which we defined as follows:

**Definition 1**  $F$  is said to be

- (i) *monotone* if  $\langle u - v, F(u) - F(v) \rangle \geq 0 \quad \forall u, v \in H$ .
- (ii) *strongly monotone* if  $\exists \nu > 0$  s.t.  $\langle u - v, F(u) - F(v) \rangle \geq \nu \|u - v\|^2 \quad \forall u, v \in H$ .
- (iii) *pseudomonotone* if  $\langle u - v, F(v) \rangle > 0 \Rightarrow \langle u - v, F(u) \rangle \geq 0 \quad \forall u, v \in H$ .
- (iv) *Lipschitz continuous* if  $\exists L > 0$  s.t.  $\|F(u) - F(v)\| \leq L \|u - v\| \quad \forall u, v \in H$ .

Finding a saddle point  $(y^*, x^*)$  to the min-max problem

$$\min_{y \in Y} \max_{x \in X} \Phi(y, x)$$

can be written as a special case of the variational inequality problem:

$$\text{find } v^* \in K \quad \text{s.t.} \quad \langle v - v^*, F(v^*) \rangle \geq 0 \quad \forall v \in K, \quad (47)$$

where

$$v = \begin{bmatrix} y \\ x \end{bmatrix} \quad F(v) = \begin{bmatrix} \Phi_y(x, y) \\ -\Phi_x(x, y) \end{bmatrix} \quad \text{and } K = Y \times X.$$

In particular Our ROF problem (9) and (33) can both be transformed into a variational inequality problem  $\text{VI}(K, F)$  in (47) with  $F$  and  $K$  defined as follows.

For unconstrained ROF (9):

$$F(v) = \begin{bmatrix} Ax + \lambda(y - z) \\ -A^T y \end{bmatrix} \quad \text{and } K = \mathbb{R}^N \times X.$$

For constrained ROF (33):

$$F(v) = \begin{bmatrix} Ax \\ -A^T y \end{bmatrix} \quad \text{and } K = Y \times X.$$

Variational inequality problem is closely related to the fixed-point problem. The fixed-point theory has played an important role in the development of various algorithms for solving variational inequalities. In fact we have the following well-known result (see, e.g. [1, pp. 267]):

**Lemma 1**  *$v^*$  is a solution of  $VI(K, F)$  if and only if*

$$v^* = P_K(v^* - \alpha F(v^*)) \quad \text{for any } \alpha > 0.$$

The fixed-point formulation in the above lemma suggests the simple iterative algorithm of solving for  $u^*$ .

### VI Algorithm 1

$$v^{k+1} = P_K(v^k - \alpha_k F(v^k)). \quad (48)$$

The convergence of the above algorithm requires  $F$  to be strongly monotone and Lipschitz continuous, which is too restrictive in many cases. An alternative approach is to consider the following ‘implicit’ iterative scheme

### VI Algorithm 2

$$v^{k+1} = P_K(v^k - \alpha_k F(v^{k+1})). \quad (49)$$

The convergence of this new algorithm only requires monotonicity of  $F$  but it is often difficult to solve the implicit update at each iteration, making it less practical.

To overcome the drawbacks of the projection methods defined in (48) and (49), Korpelevich [12] first proposed a modified method called *extragradient algorithm*. It consists two projections at each iteration: a predictor step and a corrector step.

### VI Algorithm 3

$$\bar{v}^k = P_K(v^k - \alpha_k F(v^k)) \quad (50a)$$

$$v^{k+1} = P_K(v^k - \alpha_k F(\bar{v}^k)) \quad (50b)$$

Global convergence is proved for the above algorithm if  $F$  is pseudomonotone or Lipschitz continuous or the problem satisfy some local error bound (see [20]), provided the step size  $\alpha_k$  is small enough to satisfy

$$\alpha_k \|F(v^k) - F(\bar{v}^k)\| \leq \mu \|v^k - \bar{v}^k\| \quad \text{for some fixed } \mu \in (0, 1),$$

which can be obtained by simple Armijo type line search.

There are many other variants of the original extragradient algorithm with different predictor search rule and corrector step size aiming to improve performance (see [15] and [20]). New related developments in this direction can also be found in [13] and [15], where the final solution is obtained by averaging along the solution path.

Our ROF problem (9) and (33) can both be transformed into a variational inequality problem  $\text{VI}(K, F)$  with a monotone and Lipschitz continuous mapping  $F$ . Some of the existing algorithms can be applied directly with proved global convergence. However, numerical experiments show that none of these existing methods has comparable performance to our algorithm. There are many possible explanations for this. First of all, in the variational inequality setting the variables  $y$  and  $x$  are combined as one variable  $u$  and have to be updated in one step with same steplength; while in our approach the primal  $y$  and dual  $x$  are updated alternatively in a Gauss-seidal type of way with freedom to choose their own step sizes. More importantly, all the existing algorithms are developed to solve variational inequalities as a general class; while our method exploits the particular information of the problem, including the bilinear function  $F$  and special structure of the set  $K$ , which allow us to choose optimal step size to improve the performance. On the other hand, our approach is lack of a global convergence proof which would be useful to provide some benchmark rules and can help us better understand how the algorithm works.

## 4 Numerical Experiments

We report on computational experiments for three test problems in image denoising and four test problems in image deblurring. All the programs are run in an IBM T61 Notebook PC with Intel Core 2 Duo 2.5G CPU. All methods are coded in MATLAB. It is expected that the performance can be improved by recoding in C or C++, but we believe that improvements would be fairly uniform across all the algorithms.

### 4.1 Experiments on Image Denoising

We test three problems on image denoising. The original clean images and the input noisy images are shown in Figure(1). The size of the two test problems are  $128 \times 128$ ,  $256 \times 256$ , and  $512 \times 512$  respectively. The noisy images are generated by adding Gaussian noises with standard deviation  $\sigma = 20$  to the original clean images.

The parameter  $\lambda$  in the unconstrained ROF model (2) is inverse related to the noise level  $\sigma$  and usually need to be tuned for each individual image. In our case,  $\lambda$  is chosen in the following way. We first compute the constrained ROF model by algorithm PDHG-C for the optimality solution  $(y^*, x^*)$ . Then

the particular  $\lambda$  that will make the unconstrained ROF model (2) equivalent to the constrained model (1) is given by

$$\lambda = \frac{\|Ax^*\|}{n\sigma}. \quad (51)$$

For our test problems, the parameters  $\lambda$  obtained in the above way are 0.0415, 0.053 and 0.0485 respectively.

We tested the following algorithms for denoising problems:

- Chambolle's semi-implicit gradient descent method [4];
- Split Bregman method of [9]
- The CGM method of [6].
- Primal-dual hybrid gradient methods proposed in Section 2;

Although suitable constant stepsizes will give fast convergence results already, the power of our proposed algorithm shall be most exploit with some optimal strategy of choosing stepsizes  $(\tau_k, \theta_k)$ . Throughout the experiments, we use the following stepsize strategy:

**Algorithm PDHG**     $\tau_k = 0.2 + 0.08k, \quad \theta_k = (0.5 - \frac{5}{15+k})/\tau_k;$

**Algorithm PDHG-C**     $\tau_k = 0.2 + 0.08k \quad \theta_k = 0.5/\tau_k.$

In Chambolle's method, we take the time step to be 0.248 for its near optimal performance. In the CGM implementation, we used a direct solver for the linear system at each iteration by using sparse Choleskey factorization with symmetric approximate minimum degree ordering (the built-in `symamd` and `chol` functions in MATLAB). We note the conjugate gradient iterative solver was slower on these examples. The smooth parameter  $\beta$  is dynamically updated based on duality gap rather than fixed. In particular we take  $\beta^{(0)} = 100$  and let  $\beta^{(k)} = \beta^{(k-1)} \cdot \left(\frac{G^{(k)}}{G^{(k-1)}}\right)^2$ . We noticed that this simple strategy of updating  $\beta$  borrowed from interior-point methods outperforms the classical CGM measured by the decrease of duality gap.

The decision about when an approximate solution is of sufficiently high quality to terminate the algorithm can be difficult for general constrained optimization problems. Often, we wish the approximate solution  $x$  to be close to a global minimizer  $x^*$  and/or the function value  $F(x)$  be close to  $F(x^*)$ . In the denoising case, the duality gap provides a reliable and easily calculated stopping criterion. We terminate our program whenever the relative duality gap  $\frac{G(y^k, x^k)}{D(x^k)}$  reaches a pre-specified tolerance threshold `TOL`.

Tables 3, 4 and 5 report numbers of iterations and CPU times required by two primal-dual hybrid gradient algorithms as well as by Chambolle's algorithm and CGM method for the relative duality gap to achieve certain

threshold. In all codes, we used the same starting point  $(y^0, x^0) = (z, 0)$ . (Convergence does not depend on initial conditions though.) We vary the threshold TOL from  $10^{-2}$  to  $10^{-6}$ , producing results of increasingly high accuracy as TOL is decreased. Note we have not put the split Bregman method here because this method can not generate duality gap as a stop criterion.

The results in the tables demonstrates that our proposed approaches is very competitive to existing methods. They are the winners for all tests with different stopping criterions  $\text{TOL} = 10^{-2}, 10^{-4}, 10^{-6}$ . It is significantly faster than Chambolle's method to obtain medium-high accurate solutions and significantly faster than CGM method to obtain low-medium accurate solutions.

Figure 2 shows the denoised images obtained at different values of TOL. Note that visually there is little difference between the results obtained with two tolerance values  $10^{-2}$  and  $10^{-4}$ . Smaller values of TOL do not produce further visual differences.

Figure 3 plots the relative duality gap against the CPU time cost for Chambolle's method, CGM method as well as the *PDHG* algorithm.

Figure 4 plots the relative  $L^2$  error and  $L^\infty$  against the CPU time cost. The errors are defined as

$$\begin{aligned}\text{Relative } L^2\text{error} &= \frac{\|y^k - y^*\|}{\|y^*\|} \\ L^\infty\text{error} &= \|y^k - y^*\|_\infty,\end{aligned}$$

where the exact solution  $u^*$  is computed using CGM method with  $\frac{G(y^*, x^*)}{D(x^*)}$  reaches  $\text{TOL} = 10^{-12}$ .

Figure 5 plots the “visual error” against the CPU time cost. The visual error is defined as the ratio of pixel that differer from the exact solution visually. More specifically, if we use  $[a]$  to denote the closed integer to number  $a$ , then the visual error is defined as

$$\text{VE}(u) = \frac{\#\{(i, j) : [u_{i,j}] \neq [u_{i,j}^*]\}}{n^2}$$

From all four different measures in Figure 3, 4 and 5, we see that the primal-dual hybrid gradient method is the best performer in all situations in terms of efficiency. We also confirmed that the Split-Bregman algorithm have very fast visual convergence as the authors claimed in [9]. The overall comparison results in different scopes of accuracies are listed as follows:

1. Low-accuracy, visually acceptable solution. PDHG, Split-Bregman, and Chambolle's method are all competitive, with PDHG slightly faster than the other two.

2. medium accuracy, visually satisfactory solution. Chambolle’s method starts losing its competency. PDHG and Split-Bregman are still competitive, with PDHG being from slightly faster to much faster than Split-Bregman.
3. medium-high accurate, visually convergent (i.e. visual error =0) solution. Split-Bregman starts losing its competency. PDHG is significantly faster than Split-Bregman and Split-Bregman is significantly faster than Chambolle’s method. CGM is catching to catch up but PDHG is still the fastest.
4. super-high accurate benchmark solution. High-order implicit method like CGM finally wins.

## 4.2 Experiments on Image Deblurring/Denoising

The original images France ( $512 \times 512$ ) and Man ( $1024 \times 1024$ ) are shown in Figure (). The intensities of the original images are scaled into the range between 0 and 1. We tested two typical types of blurring effects: motion and Gaussian. Specifically, we used the functions `fspecial` and `imfilter` from the MATLAB Image Processing Toolbox with the types “motion” and “gaussian”, and then added to the resulting blurry images the Gaussian white noise with a mean 0 and a standard deviation  $10^{-3}$  by MATLAB function `imnoise`. With the “motion” blurring, we set the angle parameter “theta” to 135 and the motion distance parameter “len” to two different values 21 and 91 corresponding to the medium and severe levels of blurring, respectively. With the “medium” Gaussian blurring, we set the blurring window size “hsize” equal to 21 and the standard deviation “sigma” equal to 5, and with the “severe” Gaussian blurring, 41 and 10, respectively. We summarize the information on the four test images in Table 2.

Table 2: Information on Test Deblurring Problems

| Test Problem | Original Image | Size               | Blurring Type     |
|--------------|----------------|--------------------|-------------------|
| 1            | France         | $512 \times 512$   | medium “motion”   |
| 2            | France         | $512 \times 512$   | severe “motion”   |
| 3            | Man            | $1024 \times 1024$ | medium “Gaussian” |
| 4            | Man            | $1024 \times 1024$ | severe “Gaussian” |

We tested our PDHG-D algorithm proposed in section 2.4 and the FTVd algorithm in [19]. The parameter  $\lambda$  is chosen as  $\lambda = \min\{0.2/\sigma^2, 2 \cdot 10^{-11}\}$ . For algorithm PDHG-D, we terminate the algorithm when the stop criterion

$$\|y^k - y^{k-1}\|_\infty \leq \text{TOL} = 10^{-3}$$

is satisfied. (Note the intensity is scaled in the range  $[0, 1]$ ). One can also use a smaller TOL to achieve better solutions.

Although suitable constant stepsize works well in FTVd, some adaptive stepsize rule works even better. We use the following strategy for stepsize:

$$\tau_k = 10 + 40k, \quad \theta_k = (1 - \frac{0.2}{k})/\tau_k.$$

Note the place of the parameter  $\lambda$  is different for stepsize in PDHG algorithm (26) and PDHG-D algorithm (45).

Figure 7 and 8 showed the blurry images and the deblurring results by our our algorithm . Table 6 reports the average computational costs of PDHG-D as well as of FTVd algorithm. It shows that the new proposed algorithm is very competitive to existing fast methods.

## 5 Conclusions

We have proposed a primal-dual hybrid gradient method to solve the total variation based image restoration model of Rudin, Osher and Fatemi (ROF) [17]. The algorithm tries to improve performance by alternating between the primal and dual variable and exploit information from both variables. We compare our method with two popular existing approaches proposed by Chambolle [4], Goldstein and Osher [9] and Chan, Golub, and Mulet [6] and show our method is consistently faster than earlier approaches in all experiments with different stopping criterions. Our algorithm can be applied to solve both the unconstrained ROF and constrained ROF model and, in theory, it can be applied to solve other TV minimization model or  $L^1$  minimization problem by transforming it to the min-max form. We also pointed out that our algorithm is related to existing projection type methods for solving variational inequalities.

## 6 Acknowledgements

This work is supported by NSF grant DMS-0610079 and ONR grant N00014-06-1-0345. The work was performed while T. Chan was on leave at the National Science Foundation as Assistant Director of the Directorate for Mathematics.

## References

- [1] Bertsekas, D. P. & Tsitsiklis, J. N., *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [2] J. L. Carter. Dual method for total variation-based image restoration, *UCLA CAM Report* 02-13, 2002.

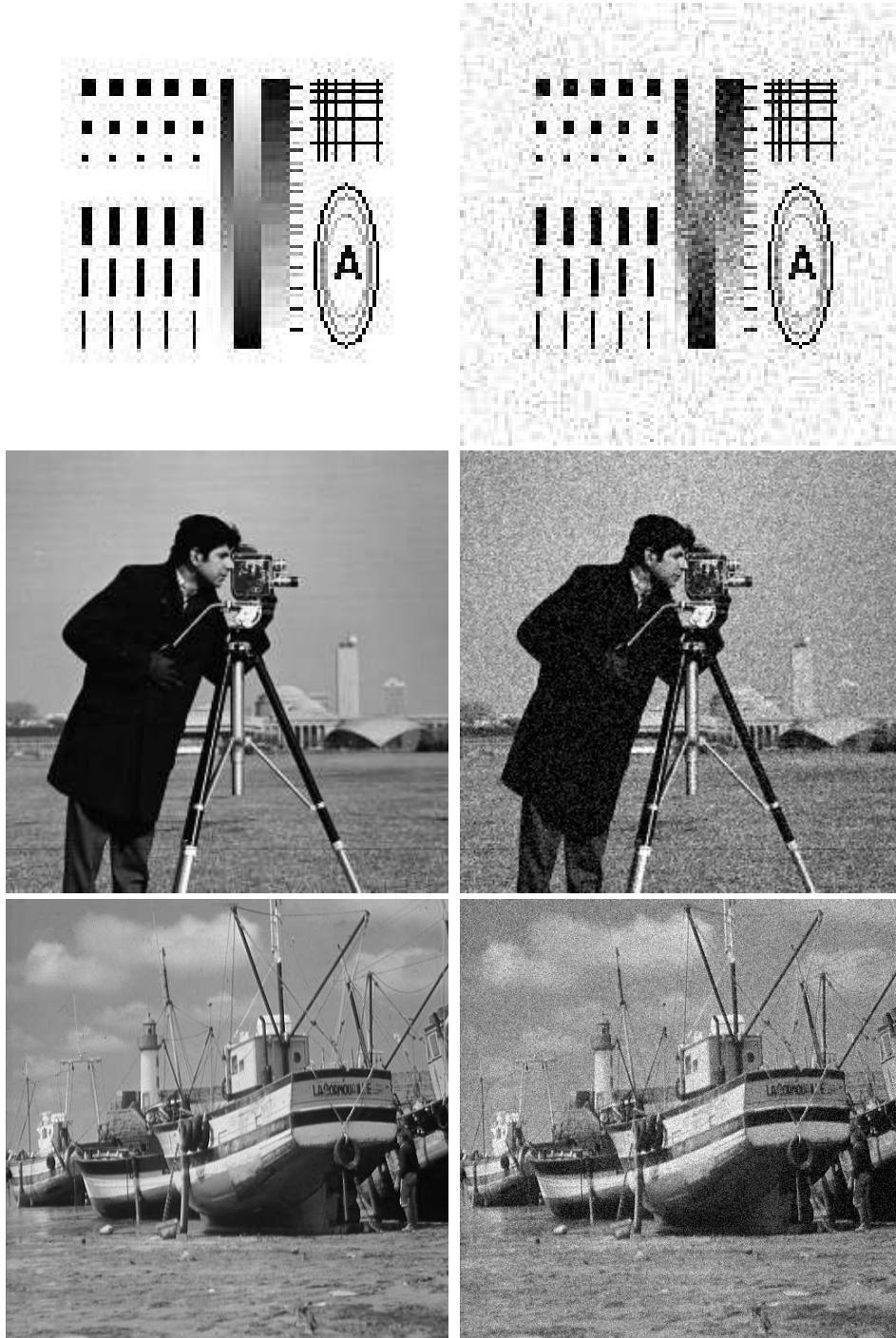


Figure 1: Denoising test problems. Left: original clean image. Right: noisy image with Gaussian noise ( $\sigma = 20$ ). First row: test problem 1,  $128 \times 128$  *shape*. Middle row: test problem 2,  $256 \times 256$  *cameraman*. Bottom row: test problem 3,  $512 \times 512$  *boat*

Table 3: Iterations and CPU times. Problem 1,  $128 \times 128$ ,  $\lambda = 0.0415$ 

| Algorithms | $TOL = 10^{-2}$ |         | $TOL = 10^{-4}$ |         | $TOL = 10^{-6}$ |         |
|------------|-----------------|---------|-----------------|---------|-----------------|---------|
|            | Iter            | CPU (s) | Iter            | CPU (s) | Iter            | CPU (s) |
| Chambolle  | 37              | 0.19    | 2074            | 5.8     | 36876           | 107     |
| PDHG       | 14              | 0.11    | 106             | 0.33    | 456             | 1.2     |
| PDHG-C     | 14              | 0.13    | 100             | 0.36    | 446             | 1.3     |
| CGM        | 5               | 1.20    | 14              | 3.5     | 19              | 4.7     |

Table 4: Iterations and CPU times. Problem 2,  $256 \times 256$ ,  $\lambda = 0.053$ 

| Algorithms | $TOL = 10^{-2}$ |         | $TOL = 10^{-4}$ |         | $TOL = 10^{-6}$ |         |
|------------|-----------------|---------|-----------------|---------|-----------------|---------|
|            | Iter            | CPU (s) | Iter            | CPU (s) | Iter            | CPU (s) |
| Chambolle  | 45              | 1.2     | 1213            | 31      | 22597           | 579     |
| PDHG       | 14              | 0.28    | 73              | 1.5     | 328             | 6.6     |
| PDHG-C     | 14              | 0.34    | 75              | 1.6     | 336             | 6.8     |
| CGM        | 6               | 7.9     | 14              | 19      | 19              | 26      |

Table 5: Iterations and CPU times. Problem 3,  $512 \times 512$ ,  $\lambda = 0.0485$ 

| Algorithms | $TOL = 10^{-2}$ |         | $TOL = 10^{-4}$ |         | $TOL = 10^{-6}$ |         |
|------------|-----------------|---------|-----------------|---------|-----------------|---------|
|            | Iter            | CPU (s) | Iter            | CPU (s) | Iter            | CPU (s) |
| Chambolle  | 61              | 7.5     | 1218            | 150     | 21925           | 2715    |
| PDHG       | 16              | 1.5     | 72              | 6.8     | 320             | 30      |
| PDHG-C     | 16              | 1.6     | 71              | 6.9     | 316             | 31      |
| CGM        | 7               | 51      | 14              | 101     | 20              | 143     |

Table 6: Deblurring: Comparison of PDHG-D and FTVd algorithm.

| Problem | CPU Time (s) |      | SNR (dB) |      | Objective Function |        |
|---------|--------------|------|----------|------|--------------------|--------|
|         | PDHG-D       | FTVd | PDHG-D   | FTVd | PDHG-D             | FTVd   |
| 1       | 1.4          | 4.6  | 22.8     | 22.3 | 27330              | 29272  |
| 2       | 3.3          | 5.8  | 13.3     | 14.2 | 36374              | 36102  |
| 3       | 12.2         | 25.5 | 14.3     | 14.3 | 120287             | 119894 |
| 4       | 13.0         | 26.9 | 11.7     | 12.2 | 118962             | 117675 |

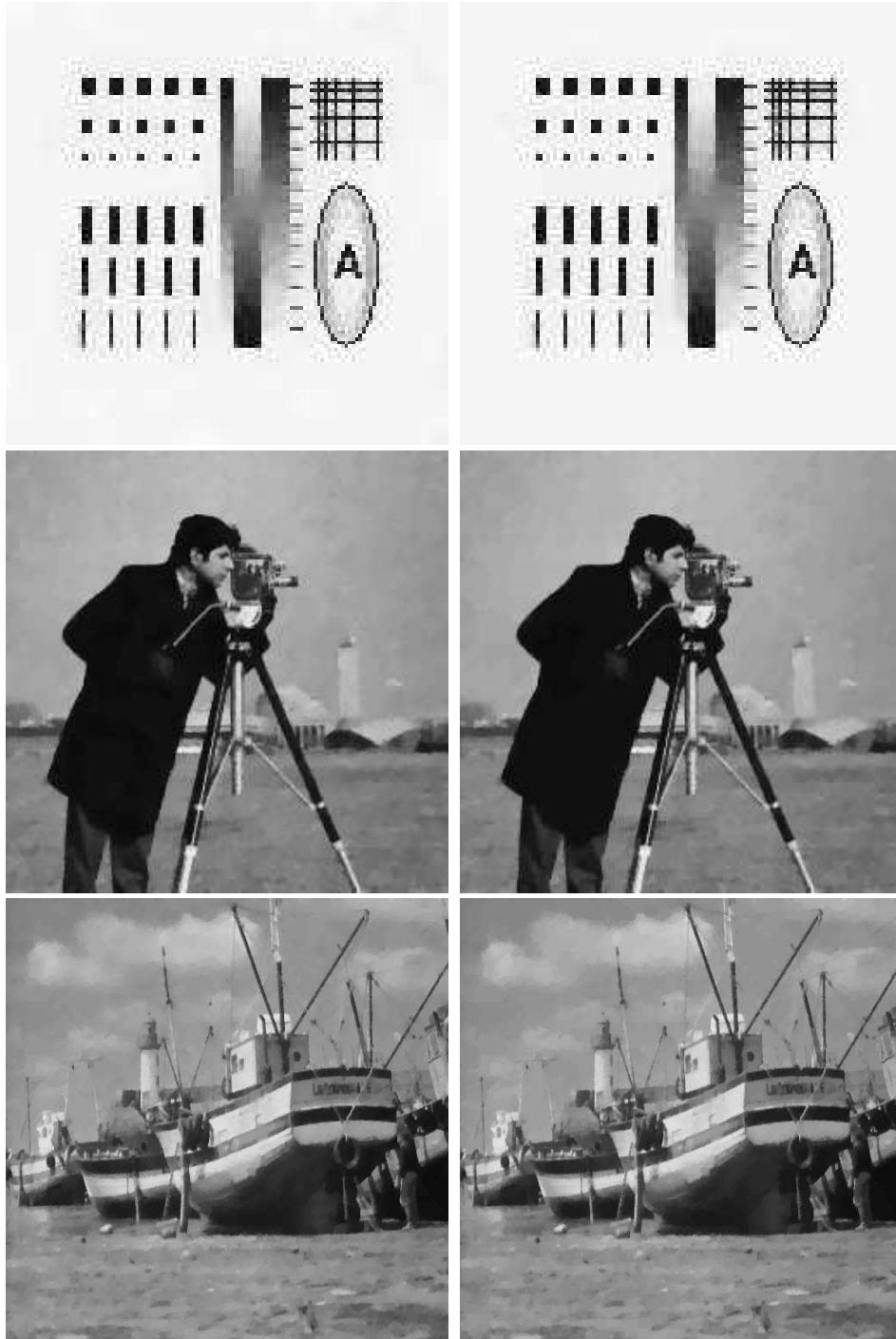


Figure 2: The denoised images with different level of termination criterions.  
left column:  $\text{TOL} = 10^{-2}$ , right column:  $\text{TOL} = 10^{-4}$ .

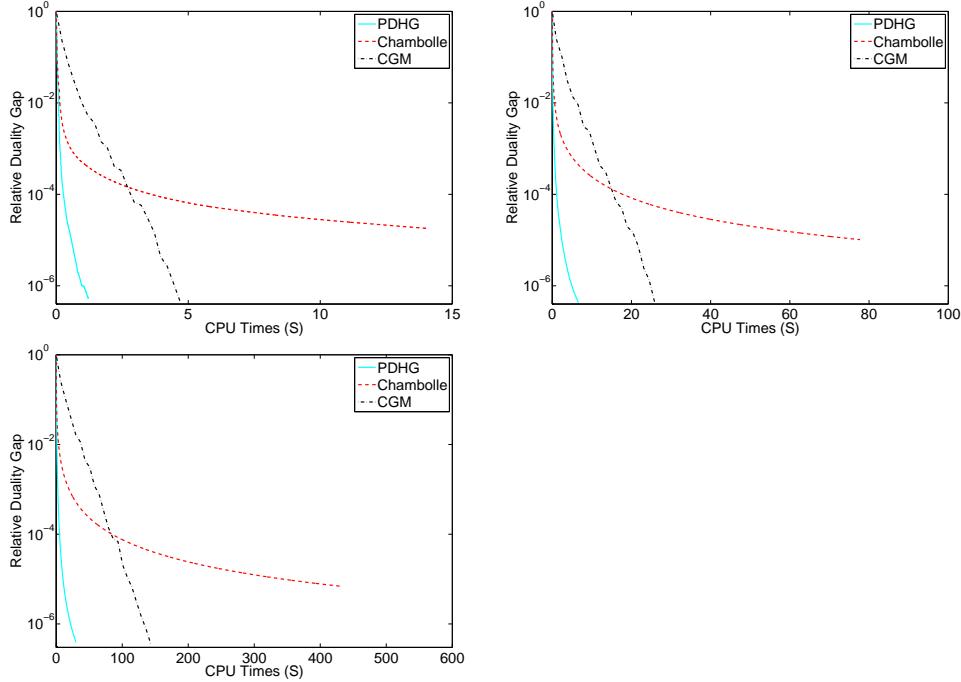


Figure 3: Plot of relative duality gap  $\frac{G(y^k, x^k)}{G(y^0, x^0)}$  v.s. CPU time. Top left: test problem 1. Top right: test problem 2. Bottom: test problem 3.

- [3] T. F. Chan, S. Esedoglu, F. Park & A. Yip. Total variation image restoration: overview and recent developments. In *Handbook of Mathematical Models in Computer Vision*. Springer Verlag, 2005. Edt. by: N. Paragios, Y. Chen, O. Faugeras.
- [4] A. Chambolle. An algorithm for total variation minimization and applications, *J. Math. Imag. Vis.*, 20 (2004), 89-97.
- [5] A. Chambolle & P. L. Lions, Image recovery via total variation minimization and related problems, *Numer. Math.*, 76 (1997), 167-188.
- [6] T. F. Chan, G. H. Golub & P. Mulet. A nonlinear primal dual method for total variation based image restoration, *SIAM J. Sci. Comput.*, 20 (1999), 1964-1977.
- [7] T. F. Chan, M. Zhu. Fast Algorithms for Total Variation-Based Image Processing. To appear in: *The Proceedings of 4th ICCM*, Hangzhou, China 2007.
- [8] D. Goldfarb & W. Yin. Second-order cone programming methods for total variation-based image restoration, *SIAM J. Sci. Comput.*, 27 (2005), 622-645.

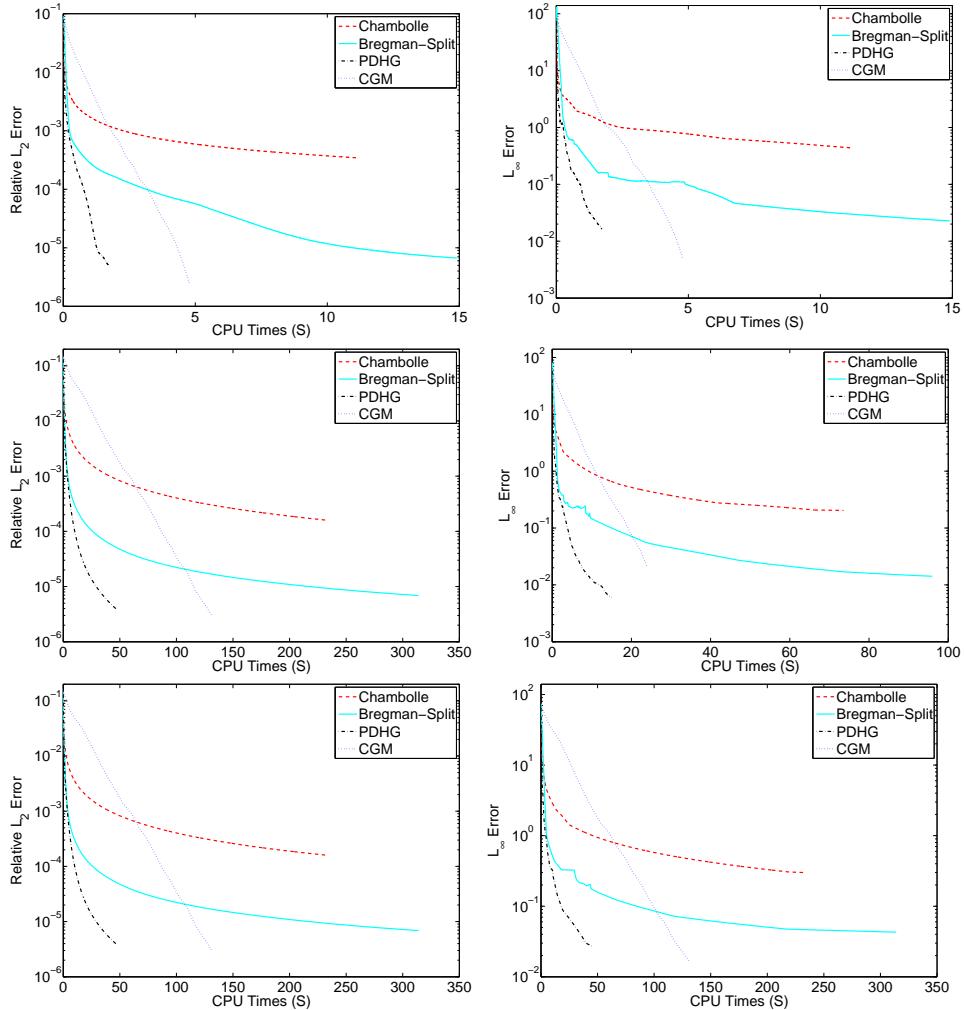


Figure 4: Plot of relative  $L^2$  error and  $L^\infty$  error v.s. CPU time.  
Left: relative  $L^2$  error  $\frac{\|y^k - y^*\|}{\|y^*\|}$ , Right:  $L^\infty$  error  $\|y^k - y^*\|_\infty$ .  
Top: problem 1. Middle: problem2. Bottom: problem 3.

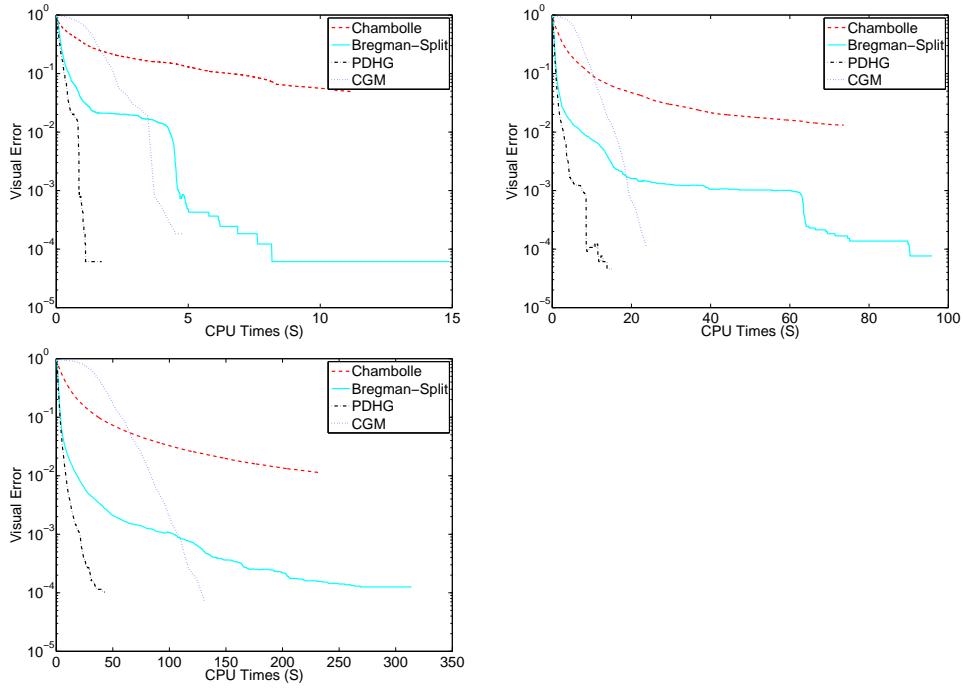


Figure 5: Plot of visual error VE v.s. CPU Time. Top left: test problem 1. Top right: test problem2. Bottom: test problem 3.

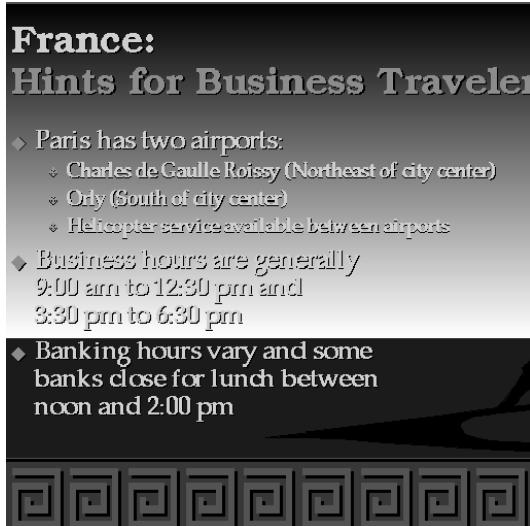


Figure 6: Original test images for delurring. Left:  $512 \times 512$  France. Right:  $1024 \times 1024$  man.

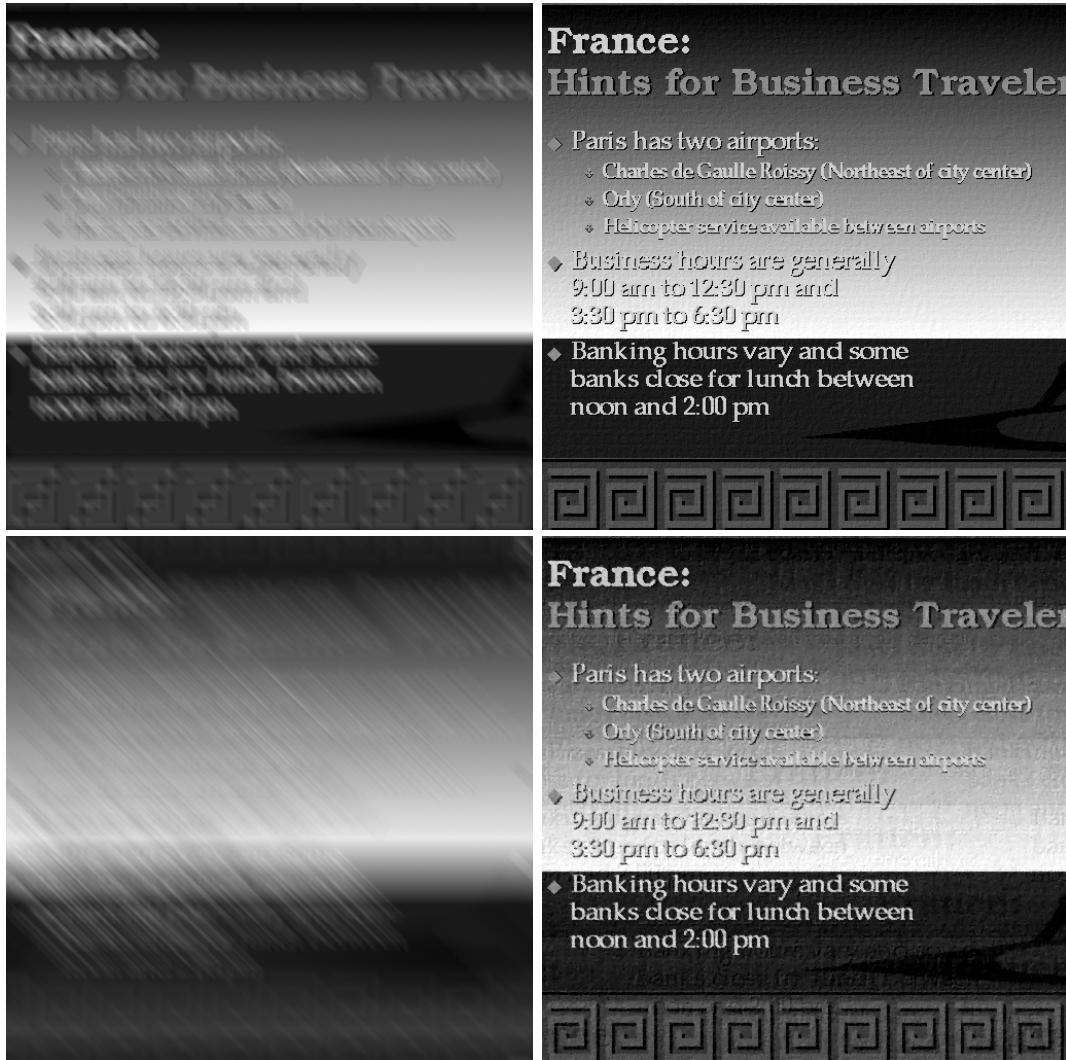


Figure 7: Deblurring problems 1 and 2. Left: blurry image, Right: delurring results. Top: medium “motion” blur, Bottom: severe “motion” blur.



Figure 8: Deblurring problems 3 and 4. Left: blurry image, Right: delurring results. Top: medium “Gaussian” blur, Bottom: severe “Gaussian” blur.

- [9] T. Goldstein & S. Osher. The split Bregman algorithm for  $L^1$  Regularized problems. *UCLA CAM Report*, 08-29 (2008).
- [10] P.T. Harker & J.S. Pang. Finite-dimensional variational inequality and nonlinear complementarity problems: a survey of theory, algorithms and applications. *Math. Programming* 48 (1990), 161-220.
- [11] J.-B. Hiriart-Urruty & C. Lemaréchal. *Convex Analysis and Minimization Algorithms, Volume I*, Springer-Verlag, Berlin & Heidelberg, 1993.
- [12] G. Korpelevich. The extragradient method for finding saddle points and other problems. *Ekonomika i Matematicheskie Metody* 12 (1976), 747-756.
- [13] A. Nemirovski. Prox-Method with Rate of Convergence  $O(1/t)$  for Variational Inequalities with Lipschitz Continuous Monotone Operators and Smooth Convex-Concave Saddle Point Problems. *SIAM Journal on Optimization* 15 (2005), 229-251.
- [14] Yu. Nesterov. Dual extrapolation and its applications for solving variational inequalities and related problems. *Math. Program., Ser. B* 109 (2007), 319-344.
- [15] M. A. Noor. New extragradient-type methods for general variational inequalities. *Journal of Math. Anal. Appl.* 277 (2003), 379-394.
- [16] S. Osher & A. Marquina. Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal, *SIAM J. Sci. Comput.*, 22 (2000), 387-405.
- [17] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms, *Physica D*, 60 (1992), 259–268.
- [18] C. R. Vogel & M. E. Oman. Iterative methods for total variation denoising, *SIAM J. Sci. Stat. Comput.*, 17 (1996), 227-238.
- [19] Y. Wang, W. Yin & Y. Zhang. A Fast Algorithm for Image Deblurring with Total Variation Regularization. Rice University CAAM Technical Report TR07-10, 2007.
- [20] N. Xiu & J. Zhang. Some recent advances in projection-type methods for variational inequalities. *J. of Comp. and Applied Math.* 152 (2003), 559-585.