MATHEMATICS OF COMPUTATION
Volume 83, Number 289, September 2014, Pages 2263–2291
S 0025-5718(2014)02829-9
Article electronically published on April 1, 2014

AN AUGMENTED LAGRANGIAN BASED PARALLEL SPLITTING METHOD FOR SEPARABLE CONVEX MINIMIZATION WITH APPLICATIONS TO IMAGE PROCESSING

DEREN HAN, XIAOMING YUAN, AND WENXING ZHANG

Abstract. This paper considers the convex minimization problem with linear constraints and a separable objective function which is the sum of many individual functions without coupled variables. An algorithm is developed by splitting the augmented Lagrangian function in a parallel way. The new algorithm differs substantially from existing splitting methods in alternating style which require solving the decomposed subproblems sequentially, while it remains the main superiority of existing splitting methods in that the resulting subproblems could be simple enough to have closed-form solutions for such an application whose functions in the objective are simple. We show applicability and encouraging efficiency of the new algorithm by some applications in image processing.

1. Introduction

We consider the convex minimization problem with linear constrains and a separable objective function which is the sum of many individual functions but without coupled variables:

$$\min \left\{ \sum_{i=1}^{m} \theta_i(x_i) \;\middle|\; \sum_{i=1}^{m} A_i x_i = b,\ x_i \in \mathcal{X}_i,\ i = 1, \cdots, m \right\}, \tag{1.1}$$

where $\theta_i : \mathcal{R}^{n_i} \to \mathcal{R}$ are closed proper convex functions (not necessarily smooth); $A_i \in \mathcal{R}^{l \times n_i}$ and $\mathcal{X}_i \subseteq \mathcal{R}^{n_i}$ are closed and convex nonempty sets; $b \in \mathcal{R}^l$ and $\sum_{i=1}^m n_i = n$. Throughout, the solution set of (1.1) is assumed to be nonempty, and A_i's are all assumed to be full column-rank. We would emphasize that weaker convergence results similar to those in [33, 59, 60] can be easily derived for the method to be proposed without this full column-rank assumption on A_i's. Moreover, we use the notation $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_m$ and $x = (x_1, x_2, \cdots, x_m) \in \mathcal{R}^n$, and denote by x_{-i} the subvector of x excluding only x_i, i.e.,

$$x_{-i} := (x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_m) \in \mathcal{R}^{n-n_i}, \quad i = 1, \cdots, m.$$

Received by the editor February 16, 2012 and, in revised form, December 1, 2012.
2010 *Mathematics Subject Classification.* Primary 90C06, 90C25, 94A08.
Key words and phrases. Augmented Lagrangian method, convex programming, splitting method, parallel, image processing.
The first author was supported by NSFC Grants 11071122, 11171159, and 20103207110002 from MOE of China.
The second author was supported by the General Research Fund from Hong Kong Research Grants Council: HKBU203311.

Note that although our discussion focuses on the case of (1.1) with vector variables, our analysis can be easily extended to the case with matrix variables (see Section 5.1).

In the literature there are many articles discussing the special case of (1.1) with $m = 2$. But, we are also interested in the general case of (1.1) with $m \geq 3$; see e.g. [4, 8, 9, 51, 54] for some applications. If we regard (1.1) as a generic convex minimization problem with linear constraints, we can apply the classical augmented Lagrangian method (ALM) in [34, 46] and obtain the iterative scheme

$$\left\{\begin{array}{l} (x_1^{k+1},\cdots,x_m^{k+1}) = \arg\min\limits_{x\in\mathcal{X}}\Big\{\sum\limits_{i=1}^m \theta_i(x_i) - (\lambda^k)^T\Big(\sum\limits_{i=1}^m A_i x_i - b\Big) \\ \qquad\qquad\qquad\qquad\qquad\qquad + \frac{\beta}{2}\|\sum\limits_{i=1}^m A_i x_i - b\|^2\Big\}, \\ \lambda^{k+1} = \lambda^k - \beta\Big(\sum\limits_{i=1}^m A_i x_i^{k+1} - b\Big), \end{array}\right. \tag{1.2}$$

where $\lambda^k \in \mathcal{R}^l$ is the Lagrange multiplier and $\beta > 0$ is a penalty parameter. The scheme (1.2), however, treats the well-structured problem (1.1) as a generic purpose and ignores completely the favorable separable structure in (1.1). Thus, this straightforward application of ALM to (1.1) is not recommended. On the other hand, (1.2) provides the possibility of developing customized algorithms with consideration of the specific structure of (1.1). Taking a close look at the minimization problem in (1.2), we find that the minimization tasks over all variables x_i are coupled only by the quadratic term $\frac{\beta}{2}\|\sum_{i=1}^m A_i x_i - b\|^2$. Therefore, we can split the minimization subproblem in (1.2) into m easier and smaller subproblems by applying a Gauss-Seidel or Jacobian decomposition to this quadratic term. Accordingly, existing augmented Lagrangian based methods for (1.1) can be classified into two categories: alternating splitting methods which decompose this quadratic term in a Gauss-Seidel way and thus the decomposed subproblems should be solved sequentially, and parallel splitting methods which decompose this quadratic term in a Jacobian way and thus the decomposed subproblems can be solved simultaneously. With either of these decomposition strategies, the purpose is to exploit each of the θ_i's properties individually and thus to generate subproblems that could be easy enough to have closed-form solutions if the θ_i's are simple. Thus, the common idea of these two kinds of algorithms for solving (1.1) is combining the generic ALM with customized decompositions in accordance with the special separable structure in (1.1).

Among alternating splitting augmented Lagrangian based methods for (1.1), there are a number of celebrated methods such as the Douglas-Rachford alternating direction method of multipliers (ADM for short) in [23] (see also [22]), split linearized Bregman schemes in [11, 12, 26, 36, 44, 58] and the split inexact Uzawa method in [59, 60]. For example, the ADM scheme for (1.1) with $m = 2$ is

$$\left\{\begin{array}{l} x_1^{k+1} = \arg\min\limits_{x_1\in\mathcal{X}_1}\big\{\theta_1(x_1) - (\lambda^k)^T(A_1x_1 + A_2x_2^k - b) \\ \qquad\qquad\qquad\qquad + \frac{\beta}{2}\|A_1x_1 + A_2x_2^k - b\|^2\big\}, \\ x_2^{k+1} = \arg\min\limits_{x_2\in\mathcal{X}_2}\big\{\theta_2(x_2) - (\lambda^k)^T(A_1x_1^{k+1} + A_2x_2 - b) \\ \qquad\qquad\qquad\qquad + \frac{\beta}{2}\|A_1x_1^{k+1} + A_2x_2 - b\|^2\big\}, \\ \lambda^{k+1} = \lambda^k - \beta(A_1x_1^{k+1} + A_2x_2^{k+1} - b). \end{array}\right. \tag{1.3}$$

Obviously, the scheme (1.3) is capable of exploiting the properties of θ_1 and θ_2 individually, making the subproblems much easier and sometimes easy enough to have closed-form solutions. We refer to [9] and the references therein for applications of ADM. Naturally, one may be inclined to extend the scheme (1.3) to the general case of (1.1) with $m \geq 3$, obtaining the scheme

$$(1.4)\quad \begin{cases} x_i^{k+1} = \arg\min_{x_i\in\mathcal{X}_i} \Big\{\theta_i(x_i) - (\lambda^k)^T\big(\sum_{j=1}^{i-1} A_j x_j^{k+1} + A_i x_i + \sum_{j=i+1}^{m} A_j x_j^k - b\big) \\ \qquad\qquad + \frac{\beta}{2}\big\|\sum_{j=1}^{i-1} A_j x_j^{k+1} + A_i x_i + \sum_{j=i+1}^{m} A_j x_j^k - b\big\|^2\Big\},\ i = 1,\cdots,m; \\ \lambda^{k+1} = \lambda^k - \beta\Big(\sum_{i=1}^{m} A_i x_i^{k+1} - b\Big). \end{cases}$$

Without further assumptions, the convergence of this ADM's direct extension is still unproved in the literature even for $m = 3$ (We refer to [45, 53] where the numerical efficiency of (1.4) has been verified empirically; and [27, 35] where the convergence of (1.4) is established under further assumptions on θ_i's and/or β). This lack of convergence thus has inspired some alternating splitting augmented Lagrangian based methods in [31, 33] in the prediction-correction framework, where the output of (1.4) (i.e., the predictor) is corrected by certain correction steps. We also refer to [32] for a relevant splitting method for (1.1) whose subproblems originated from not ALM, but the proximal-point algorithm in [41].

On the contrary, parallel splitting augmented Lagrangian based methods for (1.1) decompose the ALM subproblem in (1.2) in a certain parallel way and these decomposed subproblems can be solved simultaneously. This parallel consideration makes particular sense in the era where magnitude of data increases explosively and intensive computing infrastructure treating mass data (such as parallel and distributed computing facilities) becomes more and more advanced and popular; see e.g. [6, 9]. For existing methods in this regard, we refer to [17, 39] for the case $m = 2$; [29, 31] for the case $m = 3$; and [30] for the case with general m. In particular, the parallel splitting augmented Lagrangian method in [29] is for monotone variational inequalities with two or three separable operators and it does not apply to the case where $m > 3$; and the splitting methods proposed in [30] additionally require using a certain "asymmetric proximal term" to regularize the decomposed subproblems.

In this paper, we present a parallel splitting augmented Lagrangian based method for (1.1) with a general m, where the decomposed subproblems are completely tailored for simultaneous computation. Compared to those of alternating splitting augmented Lagrangian based methods such as (1.4), the subproblems at each iteration of the new algorithm are of the same level of difficulty. In addition to proposing a new parallel splitting augmented Lagrangian based method for (1.1), another purpose of this paper is to show that the model (1.1) captures some interesting applications in the area of image processing. When the new method is applied to solve these applications, the resulting subproblems in (3.1) are all easy enough to have closed-form solutions or can be easily solved up to high precisions. Thus, the implementation of the new method is very easy for these imaging applications. We shall compare the new method numerically with some existing splitting methods, and report the numerical results.

Finally, we mention some interesting methods for solving some other models relevant to (1.1). For example, the alternating linearization approach for the sum of two convex functions without constraints in [24] and for the sum of many convex functions without constraints in [25]; the projective splitting method in [20] and the Jacobian-like method in [52] for finding zero points of the sum of many maximal monotone operators without constraints; and a parallel splitting method in [3] for coupled monotone inclusions.

The rest of the paper is organized as follows. Some necessary preliminaries are provided in Section 2. In Section 3, we present the new method for (1.1), and its convergence is proved in Section 4. Then, we apply the new method to solve some imaging applications and report the numerical results in Section 5. Some conclusions are made in Section 6.

2. Preliminaries

In this section, we summarize some basic concepts and their properties that will be useful later.

2.1. Some definitions.

Let $\|\cdot\|_p$ denote the standard definition of l^p-norm, and in particular, let $\|\cdot\| := \|\cdot\|_2$ denote the Euclidean norm. For a positive definite matrix H, we denote by $\|\cdot\|_H$ the H-norm, i.e., $\|x\|_H = \sqrt{x^T Hx}$.

Let Ω be a nonempty closed convex subset of $\mathcal{R}^n$. The indicator function of Ω is

$$\chi_\Omega(x) := \begin{cases} 0, & \text{if} \quad x \in \Omega, \\ +\infty, & \text{if} \quad x \notin \Omega. \end{cases} \tag{2.1}$$

2.2. Variational reformulation.

Since our convergence analysis for the new algorithm is essentially based on some variational properties of (1.1), we derive here a variational reformulation of (1.1).

For a convex function $\theta : \mathcal{R}^n \to \mathcal{R}$, its subdifferential is the set-valued operator given by

$$\partial\theta : \mathcal{R}^n \to 2^{\mathcal{R}^n} : x \mapsto \left\{\xi \in \mathcal{R}^n \mid (y-x)^T\xi + \theta(x) \le \theta(y),\ \forall y \in \text{dom}\ \theta\right\}.$$

Let $\mathcal{U} := \mathcal{X} \times \mathcal{R}^l$ and $\mathbf{u} = (x_1, x_2, \cdots, x_m, \lambda)$. Then, by invoking its first-order optimality condition, (1.1) can be characterized by the following variational problem: Find $\mathbf{u}^* \in \mathcal{U}$ and $\mathbf{F}_i(x_i^*) \in \partial\theta_i(x_i^*)$ $(i = 1, 2, \cdots, m)$ such that the inequalities

$$\begin{cases} (x_1 - x_1^*)^T\{\mathbf{F}_1(x_1^*) - A_1^T\lambda^*\} \ge 0, \\ (x_2 - x_2^*)^T\{\mathbf{F}_2(x_2^*) - A_2^T\lambda^*\} \ge 0, \\ \qquad\cdots\cdots \\ (x_m - x_m^*)^T\{\mathbf{F}_m(x_m^*) - A_m^T\lambda^*\} \ge 0, \\ (\lambda - \lambda^*)^T(\sum_{i=1}^m A_i x_i^* - b) \ge 0, \end{cases} \quad \forall\, \mathbf{u} \in \mathcal{U}, \tag{2.2}$$

are satisfied. Note that $\partial\theta_i$ is maximal monotone (see [49]). We denote by $\mathcal{U}^*$ the set of such $\mathbf{u}^*$ that satisfies (2.2). Then, according to [21], $\mathcal{U}^*$ is nonempty under the aforementioned nonempty assumption on the solution set of (1.1).

3. The new method

In this section, we present a new parallel splitting augmented Lagrangian based method for (1.1) and prove some relevant properties.

Algorithm 3.1 (An augmented Lagrangian based parallel splitting (ALBPS) method for (1.1)).

Step 0. Given $\gamma \in (0, 2)$, a symmetric positive definite matrix $H \in \mathcal{R}^{l\times l}$, $\epsilon > 0$ and the initial iterate $(x^0, \lambda^0) \in \mathcal{X} \times \mathcal{R}^l$ chosen arbitrarily. Set $k := 0$.

Step 1. The parallel splitting augmented Lagrangian step: solving the following convex programs for $i = 1, \cdots, m$ (possibly in parallel),

$$\tilde{x}_i^k = \arg\min_{x_i \in \mathcal{X}_i} \left\{ \theta_i(x_i) - (\lambda^k)^T p_i(x_i, x_{-i}^k) + \tfrac{1}{2}\|p_i(x_i, x_{-i}^k)\|_H^2 \right\}, \tag{3.1}$$

where $p_i : \mathcal{R}^{n_i} \times \mathcal{R}^{(n-n_i)} \to \mathcal{R}^l$ is defined as

$$p_i(x_i, x_{-i}) := \sum_{j=1}^{i-1} A_j x_j + A_i x_i + \sum_{j=i+1}^{m} A_j x_j - b. \tag{3.2}$$

Step 2. The Lagrange multiplier update:

$$\tilde{\lambda}^k = \lambda^k - H\Big(\sum_{i=1}^{m} A_i \tilde{x}_i^k - b \Big). \tag{3.3}$$

Step 3. Convex combination step to generate the new iterate

$$\mathbf{u}^{k+1} = \mathbf{u}^k - \gamma \alpha_k (\mathbf{u}^k - \tilde{\mathbf{u}}^k), \tag{3.4}$$

where

$$\alpha_k := \frac{\varphi(x^k, \tilde{x}^k)}{\psi(x^k, \tilde{x}^k)}, \tag{3.5}$$

$$\varphi(x^k, \tilde{x}^k) := \sum_{i=1}^{m} \|A_i x_i^k - A_i \tilde{x}_i^k\|_H^2 + \Big\| \sum_{i=1}^{m} A_i x_i^k - b \Big\|_H^2, \tag{3.6}$$

$$\psi(x^k, \tilde{x}^k) := (m+1)\Big(\sum_{i=1}^{m} \|A_i x_i^k - A_i \tilde{x}_i^k\|_H^2 \Big) + \Big\| \sum_{i=1}^{m} A_i \tilde{x}_i^k - b \Big\|_H^2. \tag{3.7}$$

Step 4. Termination step: If

$$\max_{1 \le i \le m} \max \left\{ \|A_i x_i^k - A_i \tilde{x}_i^k\|, \|p_i(\tilde{x}_i^k, \tilde{x}_{-i}^k)\| \right\} < \epsilon,$$

then terminate the iteration with $\tilde{\mathbf{u}}^k = (\tilde{x}^k, \tilde{\lambda}^k)$, where $\tilde{x}^k$ is an acceptable approximate solution of (1.1). Otherwise, set $k := k+1$ and go to Step 1.

Remark 3.1. Unlike (1.4), the $\tilde{x}_j^k$'s ($j = 1, \cdots, i-1$) are not required by the $\tilde{x}_i^k$-subproblem in (3.1). Thus, these $\tilde{x}_i^k$-subproblems are fully eligible for parallel computation. This treatment is particularly preferable for the cases where m is large and advanced parallel computing infrastructure is available.

Remark 3.2. In the convex combination step (3.4), α_k can be viewed as the step size along the direction $-(\mathbf{u}^k - \tilde{\mathbf{u}}^k)$ and the strategy (3.5) for determining α_k will be delineated later. Moreover, the parameter γ in the convex combination step (3.4) can be regarded as a relaxation factor which is beneficial for accelerating convergence empirically. The reason for restricting it to $(0, 2)$ will become clear in Lemma 4.3.

Remark 3.3. According to the optimality condition of the ith subproblem in (3.1), there exist $\tilde{x}_i^k \in \mathcal{X}_i$ and $\mathbf{G}_i(\tilde{x}_i^k) \in \partial\theta_i(\tilde{x}_i^k)$ such that

$$(x_i' - \tilde{x}_i^k)^T \left\{ \mathbf{G}_i(\tilde{x}_i^k) - A_i^T[\lambda^k - H p_i(\tilde{x}_i^k, x_{-i}^k)] \right\} \ge 0, \ \forall x_i' \in \mathcal{X}_i. \tag{3.8}$$

The following lemma verifies that the stopping criterion used in Algorithm 3.1 is reasonable.

Lemma 3.1. *If* $p_i(\tilde{x}_i^k, \tilde{x}_{-i}^k) = 0$ *and* $A_i x_i^k = A_i \tilde{x}_i^k$ $(i = 1, \cdots, m)$, *then* $(\tilde{x}^k, \tilde{\lambda}^k)$ *is a solution of* (2.2).

Proof. From the definition of $p_i(x_i, x_{-i})$ in (3.2), the assumption $p(\tilde{x}_i^k, \tilde{x}_{-i}^k) = 0$ means that

$$\sum_{i=1}^m A_i \tilde{x}_i^k = b. \tag{3.9}$$

Thus, (3.2) and (3.8) together imply that for any $i = 1, \cdots, m$, there exists $\mathbf{G}_i(\tilde{x}_i^k) \in \partial\theta_i(\tilde{x}_i^k)$ such that

$$(x_i' - \tilde{x}_i^k)^T \Big\{ \mathbf{G}_i(\tilde{x}_i^k) - A_i^T \tilde{\lambda}^k + \sum_{j \neq i} A_i^T H A_j (x_j^k - \tilde{x}_j^k) \Big\} \geq 0, \; \forall x_i' \in \mathcal{X}_i. \tag{3.10}$$

When $A_i x_i^k = A_i \tilde{x}_i^k,\; i = 1, \cdots, m$, we have

$$(x_i' - \tilde{x}_i^k)^T \{\mathbf{G}_i(\tilde{x}_i^k) - A_i^T \tilde{\lambda}^k\} \geq 0, \quad \forall x_i' \in \mathcal{X}_i, \;\; i = 1, \cdots, m. \tag{3.11}$$

Recall the variational characterization (2.2). Then, (3.9) and (3.11) imply that $(\tilde{x}^k, \tilde{\lambda}^k)$ is a solution of (2.2). □

From the above lemma, it is easy to see that both $\varphi(x^k, \tilde{x}^k)$ defined in (3.6) and $\psi(x^k, \tilde{x}^k)$ defined in (3.7) are positive whenever the current iterate is not a solution of (2.2). Consequently, the step size α_k defined in (3.5) is also positive for all iterations generated by Algorithm 3.1. In the following lemma, we show that the sequence $\{\alpha_k\}$ is uniformly bounded away from zero, i.e., there exists a constant $\alpha_{\min} > 0$, such that $\alpha_k \geq \alpha_{\min}$ for all $k \geq 0$.

Lemma 3.2. *Let $\{\alpha_k\}$ be the sequence generated by Algorithm* 3.1. *Then, there exists a positive constant $\alpha_{\min} > 0$, such that $\alpha_k \geq \alpha_{\min}$ for all $k \geq 0$.*

Proof. First, for arbitrary vectors u and v in $\mathcal{R}^n$, it holds that

$$\|u + v\|_H^2 \leq 2(\|u\|_H^2 + \|v\|_H^2).$$

Second, for given vectors $a_i \in \mathcal{R}^l$ $(i = 1, \cdots, m)$, let $a \in \mathcal{R}^{ml}$ be the stacking vector $(a_1^T, a_2^T, \cdots, a_m^T)^T$ and $M := [I_l, I_l, \cdots, I_l] \in \mathcal{R}^{l \times ml}$ where $I_l \in \mathcal{R}^{l \times l}$ denote the identity matrix. Then, we have

$$\begin{aligned} \left\| \sum_{i=1}^m a_i \right\|^2 &= \|Ma\|^2 = a^T M^T M a \leq \lambda_{\max}(M^T M) \left(\sum_{i=1}^m \|a_i\|^2 \right) \\ &\leq m \left(\sum_{i=1}^m \|a_i\|^2 \right). \end{aligned} \tag{3.12}$$

Then, setting $u := \sum_{i=1}^m A_i x_i^k - b$ and $v := -\sum_{i=1}^m (A_i x_i^k - A_i \tilde{x}_i^k)$, we derive that

$$\begin{aligned} \Big\| \sum_{i=1}^m A_i \tilde{x}_i^k - b \Big\|_H^2 &= \Big\| \Big(\sum_{i=1}^m A_i x_i^k - b \Big) - \sum_{i=1}^m (A_i x_i^k - A_i \tilde{x}_i^k) \Big\|_H^2 \\ &\leq 2 \Big\| \sum_{i=1}^m A_i x_i^k - b \Big\|_H^2 + 2 \Big\| \sum_{i=1}^m (A_i x_i^k - A_i \tilde{x}_i^k) \Big\|_H^2 \\ &\leq 2 \Big\| \sum_{i=1}^m A_i x_i^k - b \Big\|_H^2 + 2m \sum_{i=1}^m \|A_i x_i^k - A_i \tilde{x}_i^k\|_H^2. \end{aligned}$$

It then follows that

$$\begin{aligned}\psi(x^k,\tilde{x}^k) &= (m+1)\Big(\sum_{i=1}^m \|A_ix_i^k - A_i\tilde{x}_i^k\|_H^2\Big) + \Big\|\sum_{i=1}^m A_i\tilde{x}_i^k - b\Big\|_H^2\\ &\le (3m+1)\Big(\sum_{i=1}^m \|A_ix_i^k - A_i\tilde{x}_i^k\|_H^2\Big) + 2\Big\|\sum_{i=1}^m A_ix_i^k - b\Big\|_H^2\\ &\le (3m+1)\Big[\sum_{i=1}^m \|A_ix_i^k - A_i\tilde{x}_i^k\|_H^2 + \Big\|\sum_{i=1}^m A_ix_i^k - b\Big\|_H^2\Big]\\ &\equiv (3m+1)\varphi(x^k,\tilde{x}^k).\end{aligned}$$

Consequently,

$$\alpha_k = \frac{\varphi(x^k,\tilde{x}^k)}{\psi(x^k,\tilde{x}^k)} \ge \frac{1}{3m+1} =: \alpha_{\min},$$

which completes the proof. □

4. Convergence analysis

In this section, we prove the global convergence for the proposed algorithm. Note that the sequences generated by the splitting type methods in [29, 31, 32] are contractive with respect to their corresponding solution sets. Thus, the convergence of these methods can be easily established by following the analytic framework of contractive type methods (see [7] for details). For the sequence $\{\mathbf{u}^k\}$ generated by Algorithm 3.1, it is not necessarily contractive with respect to $\mathcal{U}^*$. Hence, standard techniques of contractive type methods are not applicable for establishing the convergence of the sequence $\{\mathbf{u}^k\}$. Our proof framework is as follows:

1) Find a lower bound of the progress made by the new iterate on the proximity to the solution set of (2.2); this is achieved by Lemma 4.2.
2) Prove the monotonicity and boundedness of the sequence $\big\{\|u^k - u^*\|_M^2 + \|\sum_{i=1}^m A_ix_i^k - b\|_H^2\big\}$, where M is a positive definite matrix to be defined later (see (4.5)), and this is completed by Lemma 4.3.
3) Establish the convergence in Theorem 4.1.

We first prove an important inequality which is used in our analysis.

Lemma 4.1. *Let* (x^*,λ^*) *be an arbitrary solution of* (2.2). *For the iterates generated by Algorithm* 3.1, *we have*

$$\begin{aligned}(\lambda^k-\lambda^*)^T\Big(\sum_{i=1}^m A_i\tilde{x}_i^k - b\Big) &\ge \Big(\sum_{i=1}^m (A_ix_i^k - A_i\tilde{x}_i^k)\Big)^T H\Big(\sum_{i=1}^m A_i\tilde{x}_i^k - b\Big)\\ &\quad + \Big\|\sum_{i=1}^m A_i\tilde{x}_i^k - b\Big\|_H^2 + \sum_{i=1}^m (A_i\tilde{x}_i^k - A_ix_i^*)^T H(A_i\tilde{x}_i^k - A_ix_i^k).\end{aligned}\tag{4.1}$$

Proof. Since (x^*,λ^*) is a solution of (2.2) and $\tilde{x}^k \in \mathcal{X}$, there exists $\mathbf{F}_i(x_i^*) \in \partial\theta_i(x_i^*)$ such that

$$(\tilde{x}_i^k - x_i^*)^T(\mathbf{F}_i(x_i^*) - A_i^T\lambda^*) \ge 0, \quad i = 1,\cdots,m.$$

On the other hand, since $\tilde{x}_i^k$ is a solution of (3.8) and $x_i^* \in \mathcal{X}_i$, there exists $\mathbf{G}_i(\tilde{x}_i^k) \in \partial\theta_i(\tilde{x}_i^k)$ such that

$$(x_i^* - \tilde{x}_i^k)^T\left\{\mathbf{G}_i(\tilde{x}_i^k) - A_i^T[\lambda^k - Hp(\tilde{x}_i^k, x_{-i}^k)]\right\} \ge 0, \quad i = 1,\cdots,m.$$

Adding the above two inequalities, we obtain that for any $i = 1, \cdots, m$,

$$(\tilde{x}_i^k - x_i^*)^T \left\{ \mathbf{F}_i(x_i^*) - \mathbf{G}_i(\tilde{x}_i^k) - A_i^T(\lambda^* - \lambda^k) - A_i^T Hp(\tilde{x}_i^k, x_{-i}^k) \right\} \geq 0. \tag{4.2}$$

Recall that the subdifferential mapping $\partial\theta_i$ is maximal monotone. Since $\mathbf{F}_i(x_i^*) \in \partial\theta_i(x_i^*)$ and $\mathbf{G}_i(\tilde{x}_i^k) \in \partial\theta_i(\tilde{x}_i^k)$, we have

$$(\tilde{x}_i^k - x_i^*)^T(\mathbf{G}_i(\tilde{x}_i^k) - \mathbf{F}_i(x_i^*)) \geq 0.$$

By rearranging the terms of (4.2) and using (3.2), we derive that

$$\begin{aligned}(\lambda^k - \lambda^*)^T A_i(\tilde{x}_i^k - x_i^*) &\geq (\tilde{x}_i^k - x_i^*)^T A_i^T Hp_i(\tilde{x}_i^k, x_{-i}^k) \\ &= (A_i\tilde{x}_i^k - A_i x_i^*)^T H \Big[\sum_{j=1}^m \left(A_j x_j^k - A_j x_j^*\right) + A_i\left(\tilde{x}_i^k - x_i^k\right) \Big] \\ &= (A_i\tilde{x}_i^k - A_i x_i^*)^T H \Big[\Big(\sum_{j=1}^m A_j x_j^k - b \Big) + A_i\left(\tilde{x}_i^k - x_i^k\right) \Big], \qquad (4.3)\end{aligned}$$

where the first equality follows from the identity

$$p_i(\tilde{x}_i^k, x_{-i}^k) = \sum_{j=1}^m A_j x_j^k + A_i\left(\tilde{x}_i^k - x_i^k\right) - b,$$

and the last one is due to

$$\sum_{j=1}^m A_j x_j^* = b. \tag{4.4}$$

Summing both sides of (4.3) for all i and using (4.4), we get that

$$\begin{aligned}(\lambda^k - \lambda^*)^T \Big(\sum_{i=1}^m A_i\tilde{x}_i^k - b \Big) &\geq \Big(\sum_{i=1}^m A_i\tilde{x}_i^k - b \Big)^T H \Big(\sum_{i=1}^m A_i x_i^k - b \Big) \\ &\quad + \sum_{i=1}^m (A_i\tilde{x}_i^k - A_i x_i^*)^T H(A_i\tilde{x}_i^k - A_i x_i^k).\end{aligned}$$

This completes the proof. □

Hereafter, we define an auxiliary block-diagonal matrix M

$$M = \begin{pmatrix} A_1^T H A_1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & A_m^T H A_m & 0 \\ 0 & \cdots & 0 & H^{-1} \end{pmatrix}. \tag{4.5}$$

Under the assumption that all A_i's are full column-rank, the matrix M defined in (4.5) is positive definite, and thus we have

$$\|\mathbf{u}\|_M^2 := \mathbf{u}^T M \mathbf{u} = \|A_1 x_1\|_H^2 + \|A_2 x_2\|_H^2 + \cdots + \|A_m x_m\|_H^2 + \|\lambda\|_{H^{-1}}^2.$$

Our next step towards the convergence is to find a lower bound of the quantity $\|u^k - u^*\|_M^2 - \|u^{k+1} - u^*\|_M^2$, which measures the progress made by the new iterate u^{k+1} on the proximity to the solution set of (2.2). With the result of Lemma 4.1, we achieve this goal in the following lemma.

Lemma 4.2. *Let* (x^*, λ^*) *be an arbitrary solution of* (2.2)*. For the iterates generated by Algorithm* 3.1*, we have*

$$\begin{aligned}(4.6)\quad &\|\mathbf{u}^k - \mathbf{u}^*\|_M^2 - \|\mathbf{u}^{k+1} - \mathbf{u}^*\|_M^2 \\ &\geq 2\gamma\alpha_k \sum_{i=1}^m \|A_i(\tilde{x}_i^k - x_i^k)\|_H^2 - \gamma^2\alpha_k^2 \sum_{i=1}^m \|A_i(\tilde{x}_i^k - x_i^k)\|_H^2 \\ &\quad + 2\gamma\alpha_k \Big\| \sum_{i=1}^m A_i x_i^k - b \Big\|_H^2 \\ &\quad - \gamma^2\alpha_k^2 \Big\| \sum_{i=1}^m A_i \tilde{x}_i^k - b \Big\|_H^2 - 2\gamma\alpha_k \Big(\sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) \Big)^T H \Big(\sum_{i=1}^m A_i x_i^k - b \Big).\end{aligned}$$

Proof. It follows from (3.4) and (4.1) that

(4.7)

$$\begin{aligned}\|\lambda^{k+1} - \lambda^*\|_{H^{-1}}^2 &= \|\lambda^k - \lambda^* - \gamma\alpha_k(\lambda^k - \tilde{\lambda}^k)\|_{H^{-1}}^2 \\ &= \|\lambda^k - \lambda^*\|_{H^{-1}}^2 - 2\gamma\alpha_k(\lambda^k - \lambda^*)^T H^{-1}(\lambda^k - \tilde{\lambda}^k) + \gamma^2\alpha_k^2\|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2 \\ &= \|\lambda^k - \lambda^*\|_{H^{-1}}^2 - 2\gamma\alpha_k(\lambda^k - \lambda^*)^T \Big(\sum_{i=1}^m A_i\tilde{x}_i^k - b \Big) + \gamma^2\alpha_k^2\|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2 \\ &\leq \|\lambda^k - \lambda^*\|_{H^{-1}}^2 - 2\gamma\alpha_k \Big\| \sum_{i=1}^m A_i\tilde{x}_i^k - b \Big\|_H^2 \\ &\quad - 2\gamma\alpha_k \Big(\sum_{i=1}^m (A_i x_i^k - A_i\tilde{x}_i^k) \Big)^T H \Big(\sum_{i=1}^m A_i\tilde{x}_i^k - b \Big) \\ &\quad - 2\gamma\alpha_k \sum_{i=1}^m (A_i\tilde{x}_i^k - A_i x_i^*)^T H (A_i\tilde{x}_i^k - A_i x_i^k) + \gamma^2\alpha_k^2 \Big\| \sum_{i=1}^m A_i\tilde{x}_i^k - b \Big\|_H^2.\end{aligned}$$

On the other hand, for any i, it follows from (3.4) that

$$\begin{aligned}\|A_i(x_i^{k+1} - x_i^*)\|_H^2 &= \|A_i[x_i^k - x_i^* - \gamma\alpha_k(x_i^k - \tilde{x}_i^k)]\|_H^2 \\ &= \|A_i(x_i^k - x_i^*)\|_H^2 - 2\gamma\alpha_k(A_i x_i^k - A_i x_i^*)^T H (A_i x_i^k - A_i\tilde{x}_i^k) \\ &\quad + \gamma^2\alpha_k^2\|A_i(x_i^k - \tilde{x}_i^k)\|_H^2.\end{aligned}$$

Summing up for all i, we obtain

$$\begin{aligned}&\sum_{i=1}^m \|A_i(x_i^{k+1} - x_i^*)\|_H^2 \\ &\quad = \sum_{i=1}^m \|A_i(x_i^k - x_i^*)\|_H^2 - 2\gamma\alpha_k \sum_{i=1}^m (A_i x_i^k - A_i x_i^*)^T H (A_i x_i^k - A_i\tilde{x}_i^k) \\ &\qquad + \gamma^2\alpha_k^2 \sum_{i=1}^m \|A_i(x_i^k - \tilde{x}_i^k)\|_H^2 \\ (4.8)\quad &\quad = \sum_{i=1}^m \|A_i(x_i^k - x_i^*)\|_H^2 - 2\gamma\alpha_k \sum_{i=1}^m (A_i\tilde{x}_i^k - A_i x_i^*)^T H (A_i x_i^k - A_i\tilde{x}_i^k) \\ &\qquad - 2\gamma\alpha_k \sum_{i=1}^m \|A_i(\tilde{x}_i^k - x_i^k)\|_H^2 + \gamma^2\alpha_k^2 \sum_{i=1}^m \|A_i(x_i^k - \tilde{x}_i^k)\|_H^2.\end{aligned}$$

Combining (4.7) and (4.8), we have

$$\begin{aligned}
(4.9)\quad &\|\lambda^{k+1}-\lambda^*\|_{H^{-1}}^2+\sum_{i=1}^m\|A_i(x_i^{k+1}-x_i^*)\|_H^2\\
&\le\|\lambda^k-\lambda^*\|_{H^{-1}}^2+\sum_{i=1}^m\|A_i(x_i^k-x_i^*)\|_H^2-2\gamma\alpha_k\sum_{i=1}^m\|A_i(\tilde{x}_i^k-x_i^k)\|_H^2\\
&\quad+\gamma^2\alpha_k^2\sum_{i=1}^m\|A_i(x_i^k-\tilde{x}_i^k)\|_H^2-2\gamma\alpha_k\Big\|\sum_{i=1}^m A_i\tilde{x}_i^k-b\Big\|_H^2\\
&\quad+\gamma^2\alpha_k^2\Big\|\sum_{i=1}^m A_i\tilde{x}_i^k-b\Big\|_H^2-2\gamma\alpha_k\Big(\sum_{i=1}^m(A_ix_i^k-A_i\tilde{x}_i^k)\Big)^TH\Big(\sum_{i=1}^m A_i\tilde{x}_i^k-b\Big).
\end{aligned}$$

Inserting the equality

$$\begin{aligned}
&\Big\|\sum_{i=1}^m A_i\tilde{x}_i^k-b\Big\|_H^2+\Big(\sum_{i=1}^m(A_ix_j^k-A_i\tilde{x}_i^k)\Big)^TH\Big(\sum_{i=1}^m A_i\tilde{x}_i^k-b\Big)\\
&=\Big(\sum_{i=1}^m A_ix_i^k-b\Big)^TH\Big(\sum_{i=1}^m A_i\tilde{x}_i^k-b\Big)\\
&=\Big\|\sum_{i=1}^m A_ix_i^k-b\Big\|_H^2+\Big(\sum_{i=1}^m A_ix_i^k-b\Big)^TH\Big(\sum_{i=1}^m(A_i\tilde{x}_i^k-A_ix_i^k\Big),
\end{aligned}$$

into (4.9), we obtain (4.6) immediately. This completes the proof. □

Then, with Lemma 4.2, we can prove that the monotonicity and boundedness of the sequence $\big\{\|\mathbf{u}^k-\mathbf{u}^*\|_M^2+\|\sum_{i=1}^m A_ix_i^k-b\|_H^2\big\}$, from which we can easily establish the convergence of the proposed Algorithm 3.1.

Lemma 4.3. *Let* (x^*,λ^*) *be an arbitrary solution of* (2.2). *For the iterates generated by Algorithm* 3.1, *we have*

$$\begin{aligned}
(4.10)\quad \|\mathbf{u}^{k+1}-\mathbf{u}^*\|_M^2+\Big\|\sum_{i=1}^m A_ix_i^{k+1}-b\Big\|_H^2\le\|\mathbf{u}^k-\mathbf{u}^*\|_M^2+\Big\|\sum_{i=1}^m A_ix_i^k-b\Big\|_H^2\\
-\gamma(2-\gamma)\alpha_k\varphi(x^k,\tilde{x}^k).
\end{aligned}$$

Proof. It follows from (3.4) that

$$\begin{aligned}
\Big\|\sum_{i=1}^m A_ix_i^{k+1}-b\Big\|_H^2&=\Big\|\Big(\sum_{i=1}^m A_ix_i^k-b\Big)-\gamma\alpha_k\sum_{i=1}^m\Big(A_ix_i^k-A_i\tilde{x}_i^k\Big)\Big\|_H^2\\
&=\Big\|\sum_{i=1}^m A_ix_i^k-b\Big\|_H^2+\gamma^2\alpha_k^2\Big\|\sum_{i=1}^m(A_ix_i^k-A_i\tilde{x}_i^k)\Big\|_H^2\\
&\quad-2\gamma\alpha_k\Big(\sum_{i=1}^m A_ix_i^k-b\Big)^TH\Big(\sum_{i=1}^m(A_ix_i^k-A_i\tilde{x}_i^k)\Big).
\end{aligned}$$

Then, adding the above inequality and (4.6) and using (3.12), we have

$$\Big\|\sum_{i=1}^m(A_ix_i^k-A_i\tilde{x}_i^k)\Big\|_H^2\le m\Big(\sum_{i=1}^m\|A_ix_i^k-A_i\tilde{x}_i^k\|_H^2\Big),$$

and therefore that

$$\begin{aligned}
&\|\mathbf{u}^{k+1}-\mathbf{u}^*\|_M^2+\Big\|\sum_{i=1}^m A_i x_i^{k+1}-b\Big\|_H^2\\
&\le \|\mathbf{u}^{k}-\mathbf{u}^*\|_M^2+\Big\|\sum_{i=1}^m A_i x_i^{k}-b\Big\|_H^2-2\gamma\alpha_k\sum_{i=1}^m\|A_i(\tilde{x}_i^k-x_i^k)\|_H^2\\
&\quad+\gamma^2\alpha_k^2\sum_{i=1}^m\|A_i(\tilde{x}_i^k-x_i^k)\|_H^2-2\gamma\alpha_k\Big\|\sum_{i=1}^m A_i x_i^{k}-b\Big\|_H^2+\gamma^2\alpha_k^2\Big\|\sum_{i=1}^m A_i \tilde{x}_i^{k}-b\Big\|_H^2\\
&\quad+m\gamma^2\alpha_k^2\sum_{i=1}^m\|A_i(\tilde{x}_i^k-x_i^k)\|_H^2.
\end{aligned}$$

Invoking the definition of α_k in (3.5), we prove the assertion immediately. □

With Lemmas 4.1–4.3, we are now ready to establish the convergence for the proposed Algorithm 3.1, as shown in the next theorem.

Theorem 4.1. *The sequence* $\{\mathbf{u}^k\}$ *generated by Algorithm* 3.1 *converges to a solution of* (2.2).

Proof. It follows from Lemma 3.2 and (4.10) that

$$\|\mathbf{u}^{k+1}-\mathbf{u}^*\|_M^2+\Big\|\sum_{i=1}^m A_i x_i^{k+1}-b\Big\|_H^2\le \|\mathbf{u}^{k}-\mathbf{u}^*\|_M^2+\Big\|\sum_{i=1}^m A_i x_i^{k}-b\Big\|_H^2 -\gamma(2-\gamma)\alpha_{\min}\varphi(x^k,\tilde{x}^k). \tag{4.11}$$

Since $\varphi(x^k,\tilde{x}^k)\ge 0$, $\gamma\in(0,2)$ and $\alpha_{\min}>0$, we have

$$\begin{aligned}
\|\mathbf{u}^{k+1}-\mathbf{u}^*\|_M^2+\Big\|\sum_{i=1}^m A_i x_i^{k+1}-b\Big\|_H^2&\le \|\mathbf{u}^{k}-\mathbf{u}^*\|_M^2+\Big\|\sum_{i=1}^m A_i x_i^{k}-b\Big\|_H^2\\
&\le\cdots\\
&\le \|\mathbf{u}^{0}-\mathbf{u}^*\|_M^2+\Big\|\sum_{i=1}^m A_i x_i^{0}-b\Big\|_H^2,
\end{aligned}\tag{4.12}$$

which implies the boundedness of the sequence $\{\|\mathbf{u}^{k+1}-\mathbf{u}^*\|_M^2\}$. Recall the assumption that A_i's are all full column-rank and the definition of M in (4.5). Thus, $\{\mathbf{u}^k\}$ is also bounded.

Furthermore, it follows from (4.11) that

$$\begin{aligned}
&\gamma(2-\gamma)\alpha_{\min}\sum_{k=0}^\infty\varphi(x^k,\tilde{x}^k)\\
&\le\sum_{k=0}^\infty\Big(\|\mathbf{u}^{k}-\mathbf{u}^*\|_M^2+\Big\|\sum_{i=1}^m A_i x_i^{k}-b\Big\|_H^2-\|\mathbf{u}^{k+1}-\mathbf{u}^*\|_M^2-\Big\|\sum_{i=1}^m A_i x_i^{k+1}-b\Big\|_H^2\Big)\\
&<+\infty,
\end{aligned}$$

which means that $\lim_{k\to\infty}\varphi(x^k,\tilde{x}^k)=0$, or equivalently,

$$\lim_{k\to\infty}\Big\|\sum_{i=1}^m A_i x_i^{k}-b\Big\|_H=0 \quad\text{and}\quad \lim_{k\to\infty}\|A_ix_i^k-A_i\tilde{x}_i^k\|_H=0,\quad i=1,\cdots,m. \tag{4.13}$$

Since $\{\mathbf{u}^k\}$ is bounded, it has at least one cluster point. Let $\bar{\mathbf{u}}$ be a cluster point of $\{\mathbf{u}^k\}$ and let $\{\mathbf{u}^{k_j}\}$ be the corresponding subsequence converging to $\bar{\mathbf{u}}$. Note that (3.3) can be written as

$$\tilde{\lambda}^k = \lambda^k - Hp_i(\tilde{x}_i^k, \tilde{x}_{-i}^k). \tag{4.14}$$

Then, by taking the limit along this subsequence in (3.10) and (4.14), and using (4.13), it is easy to derive that there exists $\mathbf{G}_i(\bar{x}_i^k) \in \partial\theta_i(\bar{x}_i^k)$ such that

$$(x_i' - \bar{x}_i)^T \left\{\mathbf{G}_i(\bar{x}_i^k) - A_i^T \bar{\lambda}^k\right\} \geq 0, \ \forall x_i' \in \mathcal{U}_i \quad \text{and} \quad \sum_{i=1}^m A_i \bar{x}_i^k - b = 0.$$

Hence, $\bar{\mathbf{u}}$ is a solution of (2.2).

Since (4.12) holds for any solution of (2.2), we can set $\mathbf{u}^* = \bar{\mathbf{u}}$ in (4.12) and obtain

$$\|\mathbf{u}^{k+1} - \bar{\mathbf{u}}\|_M^2 + \left\| \sum_{i=1}^m A_i x_i^{k+1} - b \right\|_H^2 \leq \|\mathbf{u}^k - \bar{\mathbf{u}}\|_M^2 + \left\| \sum_{i=1}^m A_i x_i^k - b \right\|_H^2 \ \ \forall\, k.$$

Therefore, we have

$$\|\mathbf{u}^{k+1} - \bar{\mathbf{u}}\|_M^2 \leq \|\mathbf{u}^k - \bar{\mathbf{u}}\|_M^2 + \left\| \sum_{i=1}^m A_i x_i^k - b \right\|_H^2 \ \ \forall\, k.$$

Recall (4.13). Thus, when k is sufficiently large, we have

$$\|\mathbf{u}^{k+1} - \bar{\mathbf{u}}\|_M^2 \leq \|\mathbf{u}^k - \bar{\mathbf{u}}\|_M^2,$$

which means that $\{\mathbf{u}^k\}$ has a unique cluster point. Hence, the sequence $\{\mathbf{u}^k\}$ converges to $\bar{\mathbf{u}}$, a solution of (2.2). □

5. Application to image processing

In this section, we show that the abstract model (1.1) captures some important concrete applications in image processing; and applying the proposed method to these applications is efficient because the resulting subproblems are all easy enough to have closed-form solutions or can be easily solved up to high precisions. Numerical comparisons with some other splitting methods and some benchmark methods in the literature are reported.

Throughout we choose the matrix H required by the proposed method as a diagonal matrix where the penalty parameter with respect to the variable x_i is β_i. That is,

$$H = \begin{pmatrix} \beta_1 I_{n_1\times n_1} & 0 & \cdots & 0 \\ 0 & \beta_2 I_{n_2\times n_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \beta_m I_{n_m\times n_m} \end{pmatrix}, \tag{5.1}$$

where $\beta_i > 0$ and $I_{n_i\times n_i}$ is the identity matrix in $\mathcal{R}^{n_i\times n_i}$ for $i = 1, \cdots, m$.

Since our numerical experiments are conducted in an ordinary personal computer without parallel processors, the resulting subproblems (3.1) can only be solved in the sequential order. For this reason, as in [25], we count only the most time-demanding subproblem among all the subproblems in (3.1) when we count the computing time for the proposed method. All the codes for implementing the proposed method were written by MATLAB 7.1 and all the numerical experiments were performed on a personal Lenovo laptop computer with Intel(R) Core (TM) 2.30GHZ and 8G memory.

5.1. **Background extraction from surveillance video with missing and noisy data.** We first test the problem of extracting background from surveillance video with missing and noisy data, whose mathematical model turns out to be an application of the model (1.1) but with matrix variables. More specifically, this application can be captured by the model of robust principle component analysis (RPCA) with incomplete and noisy observations proposed in [53]:

$$\min \ \{\|X\|_* + \tau\|Y\|_1 \ \mid \ \|P_\Omega(D - X - Y)\|_F \leq \sigma\}, \tag{5.2}$$

where $D \in \mathcal{R}^{l\times n}$ is a given matrix (data); $\|\cdot\|_*$ is the nuclear norm which is defined as the sum of all singular values and it is used to induce the low-rank component X of D; $\|\cdot\|_1$ denotes the sum of absolute values of all entries (an extension of the l^1-norm for vectors) and it is used to induce the sparse component Y of D; $\tau > 0$ is a constant balancing of the low-rank and sparsity; Ω is a subset of the index set of entries $\{1, 2, \cdots, l\} \times \{1, 2, \cdots, n\}$ which denotes the observable entries $\{D_{ij}, (i, j) \in \Omega\}$; the operator $P_\Omega : \mathcal{R}^{l\times n} \to \mathcal{R}^{l\times n}$ summarizes the incomplete observation information and it is the orthogonal projection onto the span of matrices vanishing outside of Ω so that the ijth entry of $P_\Omega(X)$ is X_{ij} if $(i, j) \in \Omega$ and zero otherwise, i.e.,

$$[P_\Omega(X)]_{ij} = \left\{ \begin{array}{ll} X_{ij}, & \text{if } (i,j) \in \Omega, \\ 0, & \text{if } (i,j) \notin \Omega, \end{array} \right. \quad 1 \leq i \leq l, \ 1 \leq j \leq n;$$

$\sigma > 0$ is the magnitude of Gaussian noise corrupting the observed data (thus the observed data may be corrupted by Gaussian noise); and $\|\cdot\|_F$ is the standard Frobenius norm. We refer to [53] for more details of the model (5.2), and [13, 16] for the original RPCA model where $\sigma = 0$ and $\Omega = \{1, 2, \cdots, l\} \times \{1, 2, \cdots, n\}$.

Let $M := P_\Omega(D)$. Then (5.2) can be equivalently expressed as

$$\min \ \{\|X\|_* + \tau\|Y\|_1 \mid \|P_\Omega(M - X - Y)\|_F \leq \sigma\}, \tag{5.3}$$

or its unconstrained version

$$\min \ \|X\|_* + \tau\|Y\|_1 + \tfrac{1}{2\mu} \|P_\Omega(M - X - Y)\|_F^2, \tag{5.4}$$

where $\mu > 0$ is a penalty parameter. The equivalence between (5.3) and (5.4) can be found in [49]. Further, introducing an auxiliary variable $Z \in \mathcal{R}^{l\times n}$, we see that (5.4) can be reformulated as

$$\min \{\|X\|_* + \tau\|Y\|_1 + \tfrac{1}{2\mu}\|P_\Omega(Z)\|_F^2 \mid X + Y + Z = M\}. \tag{5.5}$$

Therefore, (5.5) is a special case of (1.1) with matrix variables and the following specifications:

$$x = (x_1, x_2, x_3) := (X, Y, Z), \ \ A_i := I \ (i = 1, 2, 3), \ \ b = M$$

and

$$\theta_1(X) := \|X\|_*, \quad \theta_2(Y) := \tau\|Y\|_1, \quad \theta_3(Z) := \tfrac{1}{2\mu}\|P_\Omega(Z)\|_F^2.$$

To see how the model (5.2) captures the application of extracting background from surveillance video with missing and noisy data, we refer to [13]. Here we provide some preliminary background for completeness. More specifically, the video consists of a sequence of frames, and mathematically it is a natural candidate for low-rank modeling due to the low correlation between frames. Each frame consists of foreground and background. Since the background of the video needs to be flexible enough to accommodate changes in the scene, it is natural to model it as approximately low rank. Foreground objects, such as cars or pedestrians, occupy a relatively small fraction of the image pixels and hence can be treated as sparse errors. One basic imaging task in video surveillance is to separate the background from the foreground. However, in real application, the video may include missing and noisy pixels. Thus only a fraction of noise entries can be obtained. A natural question is: can we extract the background, i.e., the low-rank part, from the foreground even with missing and noisy observations? To relate this application to (5.2), D is the matrix representation of a sequence of video frames where each column represents a frame; the index set Ω (assumed known) indicates the locations of observed pixels, i.e., pixels outside Ω are missing; X represents the background while Y denotes the foreground; and σ denotes the magnitude of Gaussian noise of corrupted pixels. We assume that a video consists of n frames and each frame is expressed by an $l_1 \times l_2$ matrix. Stacking each frame as an l-dimensional vector (where $l = l_1 \cdot l_2$), we realign a sequence of frames as a matrix in $\mathcal{R}^{l\times n}$.

Now, we analyze the resulting subproblems (3.1) when the proposed method is applied to solve (5.5). Analogous to the analysis in [53], all the resulting subproblems in (3.1) are actually simple enough to have closed-form solutions. Let us review two operators for the purpose of presenting their closed-form solutions. Let $c > 0$ be a constant and $T \in \mathcal{R}^{l\times n}$ a matrix. Then, the operator $\mathcal{S} : \mathcal{R}^{l\times n} \to \mathcal{R}^{l\times n}$ defined by

$$\left(\mathcal{S}_c(T)\right)_{ij} := \operatorname{sign}(T_{ij}) \cdot \max\left\{|T_{ij}| - c, 0\right\}, \quad 1 \le i \le l,\ 1 \le j \le n, \quad \forall\, T \in \mathcal{R}^{l\times n},$$

is called the shrinkage operator (see [10]); and the operator $\mathcal{D} : \mathcal{R}^{l\times n} \to \mathcal{R}^{l\times n}$ is defined by

$$\mathcal{D}_c(T) := U \operatorname{diag}\left(\mathcal{S}_c(\Sigma)\right) V^T, \quad \forall\, T \in \mathcal{R}^{l\times n},$$

where $U\Sigma V^T$ is the singular value decomposition (SVD) of T. Now, it is easy to verify that the resulting subproblems in (3.1) for the particular application (5.5) have the following solutions. $(\tilde{X}^k, \tilde{Y}^k, \tilde{Z}^k)$ are given by

$$\begin{aligned}
\tilde{X}^k &= \mathcal{D}_{\frac{1}{\beta}}\left(\Lambda^k/\beta + M - Z^k - Y^k\right),\\
\tilde{Y}^k &= \mathcal{S}_{\frac{\tau}{\beta}}\left(\Lambda^k/\beta + M - \tilde{X}^k - Z^k\right),\\
\tilde{Z}^k_{ij} &= \begin{cases} N^k_{ij}, & \text{if } (i,j) \notin \Omega,\\ \mu\beta N^k_{ij}/(1+\mu\beta), & \text{if } (i,j) \in \Omega, \end{cases}
\end{aligned}$$

where Λ^k is the Lagrange multiplier at the kth iteration and $N^k = \Lambda^k/\beta + M - \tilde{X}^k - \tilde{Y}^k$.

We compare the proposed method (denoted by ALBPS) with: 1) the straightforward extension of ADM (1.4) without proved convergence (EADM for short); 2) the ADM-based prediction-correction method in [31] (ADBC) with proved convergence; 3) the parallel splitting augmented Lagrangian method in [29] (PSALM).

We test a video taken in the hall of an airport[1], which consists of 200 grayscale frames in the size of 144×176. Hence, the video can be realigned as a matrix $D \in \mathcal{R}^{25,344\times 200}$ and each column of D represents a frame of the video. The index set Ω representing the missing information in D is determined randomly with a certain sample ratio which is measured by $\texttt{sr}{=}\frac{100|\Omega|}{ln}\%$ ($|\Omega|$ is the cardinality of Ω).

[1] Available at http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html.

Table 5.1. Numerical results for background extraction on surveillance video.

	Iter	CPU	rank($\hat{X}$)	$\|\hat{Y}\|_0$
EADM	14	21.53	8	268321
PSALM	36	40.86	11	405537
ADBC	33	98.86	10	346701
ALBPS	24	20.03	9	271264

Here, we choose `sr`=80%. That is, there is 20% missing information in the tested video. We add additive Gaussian noise with a 10^{-3} variance and zero-mean value to each frame of the video. In Figure 5.1, we display some original and corrupted frames of this video.

Recall that theoretically the rank of the low-rank component in (5.2) should be 1. Hence, we apply certain strategies of partial SVD for the X-subproblem with the purpose of avoiding the full SVD which is computationally very demanding. In our experiments, we apply the PROPACK in [40] to implement partial SVD. As in [10, 24, 53], the strategy for determining the number of singular values to compute is as follows: setting $sv_0 = 100$ and update sv_k via

$$sv_{k+1} = \begin{cases} svp_k + 1, & \text{if } svp_k < sv_k, \\ \min\{svp_k + \mathrm{round}(0.04d), d\}, & \text{if } svp_k = sv_k, \end{cases} \tag{5.6}$$

where $d = \min\{l, n\}$ and svp_k is the number of singular values that are larger than the given threshold $1/\beta$.

The parameters in model (5.5) are fixed as $\tau = 1/\sqrt{l}$ and $\mu = 10^{-2}$. For the parameters of the methods to be compared, we take $\beta = 0.1|\Omega|/\|M\|_1$ for ADBC; $\beta = 0.005|\Omega|/\|M\|_1$ for EADM, PSALM and ALBPS. The relaxed factor $\gamma = 0.75$ for ADBC and $\gamma = 1.0$ for PSALM and ALBPS. All these values are tuned via some numerical experiments and it seems that these values are good enough to lead to their own best numerical performance for the methods to be compared. All the methods start their iterations with zero, and all take the stopping criterion as

$$\max\left\{\frac{\|X^k - \tilde{X}^k\|_F}{1+\|X^k\|_F}, \frac{\|Y^k - \tilde{Y}^k\|_F}{1+\|Y^k\|_F}\right\} < 3 \times 10^{-2}.$$

In Table 5.1, we report the iterations (‘Iter’), computing time in seconds (‘CPU’), rank of recovered low-rank component (‘rank($\hat{X}$)’), sparsity of the sparse component (‘$\|\hat{Y}\|_0$’) for the tested methods. Data in this table shows that the proposed ALBPS with proved convergence performs very competitively with the extended ADM scheme (1.4) whose convergence is still unclear, and they both outperform significantly ADBC and PSALM.

5.2. **Image restoration for mixed noise removal.** In this subsection, we show that the model (1.1) also captures the application of restoring an image corrupted by mixed noise.

Let $w \in \mathcal{R}^n$ denote a two-dimensional image and let $n = n_1 \cdot n_2$ be the total number of pixels, where n_1 and n_2 denote the numbers of pixels in the horizontal and vertical directions, respectively. Let $\partial_1 : \mathcal{R}^n \to \mathcal{R}^n$ and $\partial_2 : \mathcal{R}^n \to \mathcal{R}^n$ be the finite-difference operators in the horizontal and vertical directions, respectively,

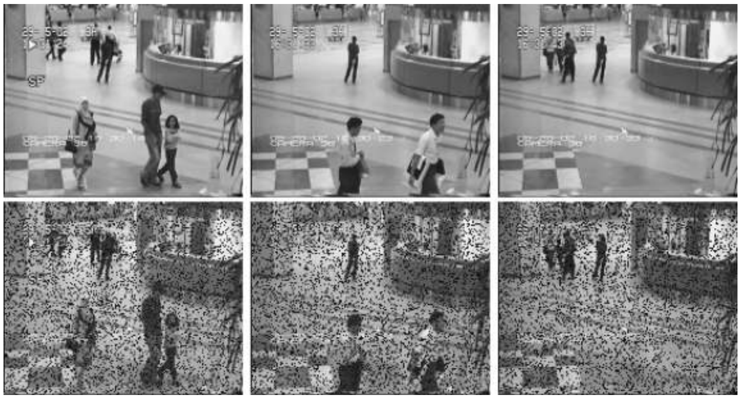

FIGURE 5.1. The 10th, 100th and 200th frames of the original video (top row) and the corrupted video (bottom row).

and let $\nabla := (\partial_1, \partial_2)$ denote the gradient operator.

$$|||y|||_p := \|(|y|)\|_p, \quad \forall y = (y_1, y_2) \in \mathcal{R}^n \times \mathcal{R}^n,$$

where $|y| := \sqrt{y_1^2 + y_2^2} \in \mathcal{R}^n$ is understood in the componentwise sense: $(|y|)_i := \sqrt{(y_1)_i^2 + (y_2)_i^2}$; see e.g., [47, Chapter 1].

To remove the mixture of impulsive and Gaussian noise, some two-phase methods were proposed in the literature, e.g., [10, 37]. We concentrate on the model in [37]:

$$\min_{w,y} \tau|||\nabla w|||_1 + \tfrac{\rho}{2}\|w - y\|^2 + \|P_{\mathcal{A}}(Gy - w^0)\|_p^p, \quad p = 1, 2, \tag{5.7}$$

where τ and ρ are positive parameters; $\mathcal{A}$ represents the set of pixels which are corrupted by the impulsive noise (all the pixels outside $\mathcal{A}$ are corrupted by the Gaussian noise); $P_{\mathcal{A}}$ is the characteristic function of the set $\mathcal{A}$, i.e., $P_{\mathcal{A}}(w)$ has the value 1 for any pixel within $\mathcal{A}$ and 0 for any pixel outside $\mathcal{A}$; G is a linear transform (e.g., a convolution by a blurring kernel); w^0 is the corrupted image by the mixed noise. As proposed in [37], by implementing the adaptive median filter (AMF), the set $\mathcal{A}$ can be identified and most of the impulsive noise within $\mathcal{A}$ can be removed. The variable y in (5.7) thus can be understood as the restored image without the impulsive noise after the implementation of AMF.

We show that the model (5.7) can be reformulated as a special case of (1.1) with $m = 3$. For succinctness, we focus only on the case of $p = 1$. By introducing the auxiliary variables u, v and z, we can reformulate (5.7) as

$$\min \left\{\tau|||u|||_1 + \tfrac{\rho}{2}\|v\|^2 + \|P_{\mathcal{A}}(z)\|_1 \mid u = \nabla w,\ v = w - y,\ z = Gy - w^0\right\}. \tag{5.8}$$

Therefore, the model (5.8) is a special case of (1.1) where $x = (x_1, x_2, x_3) := (w, y, (u, v, z))$,

$$A_1 := \begin{pmatrix} \nabla \\ I \\ 0 \end{pmatrix}, \ A_2 := \begin{pmatrix} 0 \\ -I \\ G \end{pmatrix}, \ A_3 := \begin{pmatrix} -I & 0 & 0 \\ 0 & -I & 0 \\ 0 & 0 & -I \end{pmatrix}, \ b := \begin{pmatrix} 0 \\ 0 \\ w^0 \end{pmatrix}$$

and

$$\theta_1(w) := 0, \quad \theta_2(y) := 0, \quad \theta_3(u, v, z) := \tau|||u|||_1 + \tfrac{\rho}{2}\|v\|^2 + \|P_{\mathcal{A}}(z)\|_1.$$

Recall that we take H as (5.1). Below we show that all the resulting subproblems have closed-form solutions when the proposed method is applied to solve (5.8).

- The first subproblem in (3.1), i.e., the w-subproblem for (5.8), can be formulated as
$$\begin{aligned}\tilde{w}^k = \arg\min_{w} & \left\{\left\|\nabla w - u^k - \frac{\lambda_1^k}{\beta_1}\right\|^2 + \left\|w - y^k - v^k - \frac{\lambda_2^k}{\beta_2}\right\|^2\right\}\\ & \Leftrightarrow (\beta_1 \nabla^T \nabla + \beta_2 I)\tilde{w}^k = \lambda_2^k + \beta_2(y^k + v^k) + \nabla^T(\lambda_1^k + \beta_1 u^k).\end{aligned}$$
In particular, the operator ∇ can be diagonalized by the Fast Fourier transform (FFT) when the periodic boundary condition is used, i.e., $\nabla = \mathcal{F}^{-1}D\mathcal{F}$, where D is diagonal matrix with positive diagonal entries and $\mathcal{F}$ is the FFT operator [28]. Thus, the above system of equations reduces to
$$(\beta_1 D^T D + \beta_2 I)\hat{\tilde{w}}^k = \hat{\lambda}_2^k + \beta_2(\hat{y}^k + \hat{v}^k) + D^T(\hat{\lambda}_1^k + \beta_1 \hat{u}^k),$$
where $\hat{u}^k := \mathcal{F}u^k$ for any vector u. Thus, we can get $\hat{\tilde{w}}^k$ easily by a FFT and then obtain $\tilde{w}^k$ by an inverse FFT.
- The second subproblem in (3.1), i.e., the y-subproblem for (5.8), can be written as
$$\begin{aligned}\tilde{y}^k = \arg\min_{y} & \left\{\left\|w^k - y - v^k - \frac{\lambda_2^k}{\beta_2}\right\|^2 + \left\|Gy - z^k - w^0 - \frac{\lambda_3^k}{\beta_3}\right\|^2\right\}\\ & \Leftrightarrow (\beta_2 I + \beta_3 G^T G)\tilde{y}^k = G^T[\lambda_3^k + \beta_3(z^k + w^0)] - \lambda_2^k + \beta_2(w^k - v^k).\end{aligned}$$
When G is a spatially-invariant convolution operator and the periodic boundary conditions are used, it can also be diagonalized by the FFT, i.e., $G = \mathcal{F}^{-1}D_G\mathcal{F}$, where D_G is diagonal matrix with positive diagonal entries. Thus, its solution can also be obtained easily by utilizing FFT.
- The third subproblem in (3.1), i.e., the (u, v, z)-subproblem for (5.8), is
$$\begin{aligned}(\tilde{u}^k, \tilde{v}^k, \tilde{z}^k) = \arg\min_{u,v,z} & \left\{\tau \||u|\|_1 + \frac{\rho}{2}\|v\|^2 + \|P_{\mathcal{A}}(z)\|_1 + \frac{\beta_1}{2}\left\|\nabla w^k - u - \frac{\lambda_1^k}{\beta_1}\right\|^2\right.\\ & \left. + \frac{\beta_2}{2}\left\|w^k - y^k - v - \frac{\lambda_2^k}{\beta_2}\right\|^2 + \frac{\beta_3}{2}\left\|Gy^k - z - w^0 - \frac{\lambda_3^k}{\beta_3}\right\|^2\right\},\end{aligned}$$
which is a separable problem. Thus, the solution $(\tilde{u}^k, \tilde{v}^k, \tilde{z}^k)$ can be solved separably as follows:
  - $\tilde{u}^k = \text{shrink}_{\frac{\tau}{\beta_1}}\left(\nabla w^k - \lambda_1^k/\beta_1\right)$, where $\text{shrink}_c(v)$ denotes the well known shrinkage operator (see [18, 56]). Specifically,
$$\text{shrink}_c(v) := v - \min\{c, |v|\}\frac{v}{|v|}, \quad \forall c > 0, \;\; v \in \mathcal{R}^n \times \mathcal{R}^n, \tag{5.9}$$
and $(\frac{v}{|v|})_i$ should be taken as 0 if $|v|_i = 0$.
  - $\tilde{v}^k = \left[\beta_2(w^k - y^k) - \lambda_2^k\right]/(\rho + \beta_2)$.
  - $\tilde{z}^k$ is given by
$$\tilde{z}^k = \arg\min_{z}\left\{\|P_{\mathcal{A}}(z)\|_1 + \frac{\beta_3}{2}\left\|Gy^k - z - w^0 - \frac{\lambda_3^k}{\beta_3}\right\|^2\right\},$$
which implies that
$$(\tilde{z}^k)_i = \begin{cases} \left[\text{shrink}_{\frac{1}{\beta_3}}\left(Gy^k - w^0 - \lambda_3^k/\beta_3\right)\right]_i, & \text{if } i \in \mathcal{A},\\ \left[Gy^k - w^0 - \lambda_3^k/\beta_3\right]_i, & \text{if } i \notin \mathcal{A}.\end{cases}$$

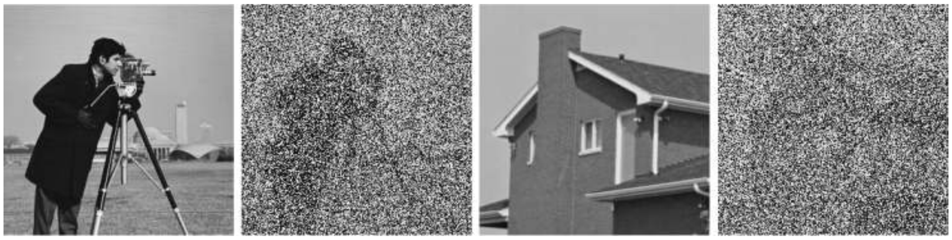

FIGURE 5.2. Original and degraded images for mixed noise removal. From left to right: original Cameraman (256 × 256), degraded Cameraman, original House (256 × 256) and degraded House.

For this application, we test the images Cameraman.png (256 × 256) and House.png (256 × 256). Both of these images are corrupted by out-of-focus blur with radius 3. The blurred images are further corrupted by both impulsive noise with intensity 0.7 and zero-mean Gaussian noise with variance 0.01. The original and degraded images are shown in Figure 5.2. As mentioned, we first apply AMF [38] to identify the set $\mathcal{A}$ and remove the impulsive noise within that set. The window size for AMF is taken as 19, as suggested in [10, 37].

For numerical comparison, besides ADBC and PSALM, we also compare ALBPS with the method in [37] (denoted by AMA). For our numerical experiments, we take $\tau = 0.02$ and $\rho = 1$ in (5.7). For the involved parameters, $\beta_1 = \beta_2 = 0.1$, $\beta_3 = 1$ for ADBC, PSALM and ALBPS; $\gamma = 0.6$ for ADBC, $\gamma = 1.6$ for PSALM and ALBPS. Recall that AMA applies the method in [14] to solve the w-subproblem and the preconditioned conjugate gradient method (PCG) for the y-subproblem. We use a maximum of 20 iterations for the w-subproblem and a tolerance of 10^{-7} for the y-subproblem. All the tested methods start their iterations from the degraded images. In the literature, the signal-to-noise ratio (SNR) in the unit of dB is usually used to measure the quality of restored images, and is defined by

$$\mathrm{SNR} = 10 \log_{10} \frac{\|w\|^2}{\|\bar{w} - w\|^2},$$

where $\bar{w}$ is the restored image and w is the original one (see e.g. [47, Appendix 3]). Thus, the initial SNR value is 0.99dB for the Cameraman image and 2.13dB for the House image.

For the tested methods, in Figure 5.3 we plot the evolutions of SNR value with respect to the computing time in seconds for the tested images. According to these curves, ALBPS is capable of restoring images with stable SNR values different than those restored by AMA within a shorter time. Moreover, since we have observed through our experiments that all of the tested methods turn out to be static in improving SNR values further after at most 80 iterations, we show the filtered image by AMF and the restored images by ALBPS after 100 iterations in Figure 5.4.

5.3. **TV-l^p image restoration problem.** In this subsection, we concentrate on the TV-l^p image restoration problem and we discuss the cases of $p = 1, 2, \infty$.

Our purpose is to show that even for a particular problem which can be reformulated as (1.1) with either $m \geq 3$ or $m = 2$, applying the proposed method to the

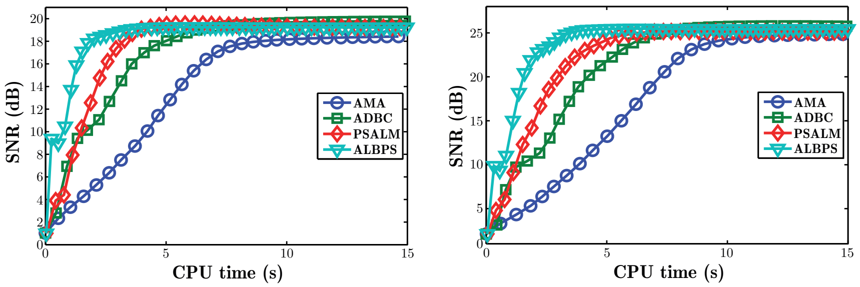

FIGURE 5.3. Evolutions of SNR w.r.t. computing time for mixed noise removal. Left: Cameraman; Right: House.

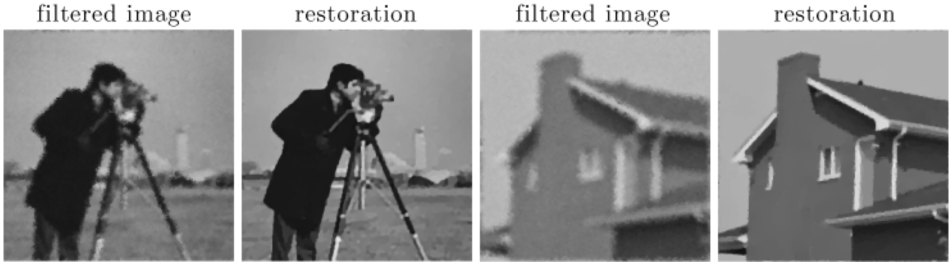

FIGURE 5.4. Filtered images by AMF and restored images via mixed noise removal by ALBPS.

reformulation with $m \geq 3$ is still very competitive with, or even more efficient than, the application of ADM (1.3) to the reformulation with $m = 2$. Therefore, together with the tested experiments for irreducible cases of (1.1) with $m \geq 3$ (where the ADM (1.3) is not applicable), efficiency of the proposed method is further illustrated.

We briefly review the background of the digital image restoration problem, and we refer to [28, 47] for more details. The digital image restoration problem plays a fundamental role in areas as varied as medical and astronomical imaging, film restoration, image and video coding. The image restoration problem is used to restore the original image $w \in \mathcal{R}^n$ from its degraded image, denoted by $w^0 \in \mathcal{R}^n$ and the mathematical model is

$$w^0 = Gw + \mathbf{n}, \tag{5.10}$$

where $\mathbf{n} \in \mathcal{R}^n$ is an additive noise corrupting the original image w and G is a convolution by a blurring kernel. Since the model (5.10) is usually ill-posed, certain regularization techniques are required. One of the most popular techniques is total variation (TV) regularization proposed in the seminal work [50], mainly because of its capability of preserving the edges of images. The TV-l^p image restoration problem is

$$\min \left\{ \||\nabla w|\|_1 \mid \|Gw - w^0\|_p \leq \sigma \right\}, \tag{5.11}$$

where $\sigma > 0$ is a constant measuring the noise level. Typically, the scalar σ is chosen as $\sigma = \|\mathbf{n}\|_p$. Different values of p reflect different types of distributions of the noise: impulsive or Laplacian noise when $p = 1$ (e.g., [15, 43]), Gaussian noise when $p = 2$ (e.g., [2, 5, 48, 55]) and uniform noise when $p = \infty$ (e.g., [57]).

Now, we show that (5.11) is in the form of (1.1) with $m = 3$. As [56], by introducing the auxiliary variables y and z, the model (5.11) can be reformulated as

$$\min \left\{ \| |y| \|_1 \mid y = \nabla w,\ Gw = z,\ z \in \mathcal{Z} \right\}, \tag{5.12}$$

where $\mathcal{Z} := \left\{ z \in \mathcal{R}^n \mid \|z - w^0\|_p \le \sigma \right\}$. Obviously, (5.12) is a special case of (1.1) with $x = (x_1, x_2, x_3) := (w, y, z)$, $\theta_1(w) := 0$, $\theta_2(y) := \| |y| \|_1$, $\theta_3(z) := \chi_{\mathcal{Z}}(z)$ (i.e., the indicator function of the set $\mathcal{Z}$), $b := 0$, and the A_i $(i = 1, 2, 3)$ are given by

$$A_1 := \begin{pmatrix} \nabla \\ G \end{pmatrix},\ A_2 := \begin{pmatrix} -I \\ 0 \end{pmatrix},\ A_3 := \begin{pmatrix} 0 \\ -I \end{pmatrix}.$$

Taking H as (5.1), we now illustrate how to solve the resulting subproblems when the proposed method is applied to solve (5.12).

- The first subproblem in (3.1), i.e., the w-subproblem for (5.12), is

  $$\begin{aligned} \tilde{w}^k &= \arg\min_w \left\{ \beta_1 \left\| \nabla w - y^k + \frac{\lambda_1^k}{\beta_1} \right\|^2 + \beta_2 \left\| Gw - z^k + \frac{\lambda_2^k}{\beta_2} \right\|^2 \right\} \\ &\Leftrightarrow (\beta_1 \nabla^T \nabla + \beta_2 G^T G) \tilde{w}^k = \nabla^T (\beta_1 y^k - \lambda_1^k) + G^T (\beta_2 z^k - \lambda_2^k). \end{aligned}$$

  As elucidated in [56], the solution of the above system of equations can be easily obtained since both ∇ and G can be diagonalized by FFT.
- The second subproblem in (3.1), i.e., the y-subproblem for (5.12), amounts to

  $$\tilde{y}^k = \arg\min_y \left\{ \| |y| \|_1 + \frac{\beta_1}{2} \left\| y - \nabla w^k - \frac{\lambda_1^k}{\beta_1} \right\|^2 \right\} = \text{shrink}_{\frac{1}{\beta_1}} \Big(\nabla w^k + \frac{\lambda_1^k}{\beta_1} \Big).$$

  where the shrinkage operator is given in (5.9).
- The third subproblem in (3.1), i.e., the z-subproblem for (5.12), is equivalent to the problem

  $$\tilde{z}^k = \arg\min_z \left\{ \chi_{\mathcal{Z}}(z) + \frac{\beta_2}{2} \left\| z - Gw^k - \frac{\lambda_2^k}{\beta_2} \right\|^2 \right\} = P_{\mathcal{Z}} \Big[Gw^k + \frac{\lambda_2^k}{\beta_2} \Big].$$

  Here, the projection onto $\mathcal{Z}$ can be computed as follows:
  - If $p = \infty$, we have

    $$(P_{\mathcal{Z}}[z])_i = w_i^0 + \min \Big\{ 1, \frac{\sigma}{|z_i - w_i^0|} \Big\} (z_i - w_i^0), \quad i = 1, 2, \cdots, n.$$
  - If $p = 2$, we have

    $$P_{\mathcal{Z}}[z] = w^0 + \min \Big\{ 1, \frac{\sigma}{\|z - w^0\|} \Big\} (z - w^0). \tag{5.13}$$
  - If $p = 1$, $P_{\mathcal{Z}}(z)$ can be efficiently solved by many existing subroutines, e.g., [19, 55].

Therefore, applying the proposed method to the reformulation (5.12), all the resulting subproblems are easy enough to have closed-form solutions or can be easily solved up to high precisions. Moreover, different values of p only result in a slight difference in computing the projection onto the set $\mathcal{Z}$. For the purpose of succinctness, below we only report the numerical results for the model (5.11) with $p = 2$ and ignore the cases $p = 1, \infty$.

Now, we show that the TV-l^2 model (i.e., (5.11) with $p = 2$) can be also treated by (1.1) but with $m = 2$ via different treatments. The first one is the approach in [42] (see Section 4.1 therein), which reformulates the TV-l^2 model as

$$\min \left\{ \||y|\|_1 \mid y = \nabla w, \|Gw - w^0\| \le \sigma \right\}, \tag{5.14}$$

where y is an auxiliary variable. For notational convenience, we denote by $\mathcal{K}$ the set $\{w \in \mathcal{R}^n \mid \|Gw - w^0\| \le \sigma\}$ in (5.14). By setting $x = (x_1, x_2) := (y, w)$, $\theta_1(y) := \||y|\|_1$, $\theta_2(w) := \chi_{\mathcal{K}}(w)$ (the indictor function defined in (2.1)), $A_1 := -I$, $A_2 := \nabla$ and $b := 0$, then (5.14) is a special case of (1.1) with $m = 2$ and thus the ADM scheme (1.3) is applicable. In fact, the resulting subproblems by applying the ADM scheme (1.3) to (5.14) can be elucidated as follows:

- The y-subproblem is

$$\begin{aligned} y^{k+1} &= \arg\min_y \left\{ \||y|\|_1 - (\lambda^k)^T(y - \nabla w^k) + \frac{\beta}{2}\|y - \nabla w^k\|^2 \right\} \\ &= \operatorname{shrink}_{\frac{1}{\beta}}\Big(\nabla w^k + \frac{\lambda^k}{\beta}\Big), \end{aligned}$$

  where the shrinkage operator is given in (5.9).
- The w-subproblem is

$$w^{k+1} = \arg\min_w \left\{ \chi_{\mathcal{K}}(w) - (\lambda^k)^T(y^{k+1} - \nabla w) + \frac{\beta}{2}\|y^{k+1} - \nabla w\|^2 \right\},$$

  which amounts to finding the projection of a vector onto the ellipse $\mathcal{K}$ (see [42, Appendix 7.1]).

We denote by C-ADMM the application of ADM (1.3) to the reformulation (5.14) in [42].

The second one is the approach proposed in [2], which reformulates the TV-l^2 model as

$$\min \left\{ \||\nabla y|\|_1 \mid w = y, Gw = z, z \in \mathcal{Z} \right\}, \tag{5.15}$$

where $\mathcal{Z} := \left\{ z \in \mathcal{R}^n \mid \|z - w^0\| \le \sigma \right\}$. Then, (5.15) is a special case of (1.1) with $m = 2$, where $x = (x_1, x_2) := (w, (y, z))$, $\theta_1(w) := 0$, $\theta_2(y, z) := \||\nabla y|\|_1 + \chi_{\mathcal{Z}}(z)$, $b := 0$ and

$$A_1 := \begin{pmatrix} I \\ G \end{pmatrix}, \quad A_2 = \begin{pmatrix} -I & 0 \\ 0 & -I \end{pmatrix}. \tag{5.16}$$

Thus, the ADM scheme (1.3) is applicable to the reformulation (5.15). Accordingly, the resulting subproblems can be solved as follows:

- The w-subproblem is

$$\begin{aligned} w^{k+1} &= \arg\min_w \left\{ \big\|w - y^k - \frac{\lambda_1^k}{\beta}\big\|^2 + \big\|Gw - z^k - \frac{\lambda_2^k}{\beta}\big\|^2 \right\} \\ &\Leftrightarrow (G^TG + I)w = y^k + \frac{\lambda_1^k}{\beta} + G^T\big(z^k + \frac{\lambda_2^k}{\beta}\big), \end{aligned}$$

  to which fast solvers such as FFT could be applicable.

- The (y,z)-subproblem is

$$(y^{k+1},z^{k+1}) = \arg\min_{y,z}\Big\{|||\nabla y|||_1 + \chi_{\mathcal{K}}(z) + \frac{\beta}{2}\big\|w^{k+1} - y - \frac{\lambda_1^k}{\beta}\big\|^2 + \frac{\beta}{2}\big\|Gw^{k+1} - z - \frac{\lambda_2^k}{\beta}\big\|^2\Big\},$$

which actually can be decomposed into two independent problems regarding y^{k+1} and z^{k+1}, respectively. More concretely, the y-subproblem is

$$y^{k+1} = \arg\min_{y}\Big\{|||\nabla y|||_1 + \frac{\beta}{2}\big\|w^{k+1} - y - \frac{\lambda_1^k}{\beta}\big\|^2\Big\},$$

to which faster solvers in the literature such as [14] are available. The z-subproblem is

$$z^{k+1} = \arg\min_{z}\Big\{\chi_{\mathcal{K}}(z) + \frac{\beta}{2}\big\|Gw^{k+1} - z - \frac{\lambda_2^k}{\beta}\big\|^2\Big\} = P_{\mathcal{Z}}\Big[Gw^{k+1} - \frac{\lambda_2^k}{\beta}\Big],$$

where $P_{\mathcal{Z}}$ is defined in (5.13).

We denote by C-SALSA the application of ADM (1.3) to the reformulation (5.15) in [2].

Moreover, there are two more influential methods in the literature for the TV-l^2 problem, i.e., the FTVd in [56] and SALSA in [1], both of which solve the unconstrained alternative of the TV-l^2 model:

$$\min_{w}\ \tau|||\nabla w|||_1 + \frac{1}{2}\|Gw - w^0\|^2, \tag{5.17}$$

where $\tau > 0$ is a constant balancing the data-fitting and regularization. More specifically, by introducing the auxiliary variable y, FTVd reformulates (5.17) as

$$\min\ \left\{\tau|||y|||_1 + \tfrac{1}{2}\|Gw - w^0\|^2\ \mid\ y = \nabla w\right\}.$$

By setting $x = (x_1, x_2) := (w, y)$, $\theta_1(w) := \frac{1}{2}\|Gw - w^0\|^2$, $\theta_2(y) := \tau|||y|||_1$, $A_1 := \nabla$, $A_2 := -I$ and $b := 0$, the ADM scheme (1.3) is thus applicable to (5.17), and the resulting subproblems can be solved as follows:

- The w-subproblem is calculated via

$$w^{k+1} = \arg\min_{w}\Big\{\|Gw - w^0\|^2 + \beta\big\|\nabla w - y^k - \frac{\lambda^k}{\beta}\big\|^2\Big\}$$
$$\Leftrightarrow (\beta\nabla^T\nabla + G^TG)w = \nabla^T(\beta y^k + \lambda^k) + G^Tw^0,$$

to which faster solvers such as FFT could be applicable (see again [56]).

- The y-subproblem is updated by

$$y^{k+1} = \arg\min_{y}\Big\{\tau|||y|||_1 + \frac{\beta}{2}\big\|\nabla w^{k+1} - y - \frac{\lambda^k}{\beta}\big\|^2\Big\} = \mathrm{shrink}_{\frac{\tau}{\beta}}\Big(\nabla w^{k+1} - \frac{\lambda^k}{\beta}\Big),$$

where the shrinkage operator is given in (5.9).

On the other hand, SALSA treats the reformulation of (5.17) as

$$\min\ \left\{\tau|||\nabla y|||_1 + \tfrac{1}{2}\|Gw - w^0\|^2\ \mid\ w = y\right\}. \tag{5.18}$$

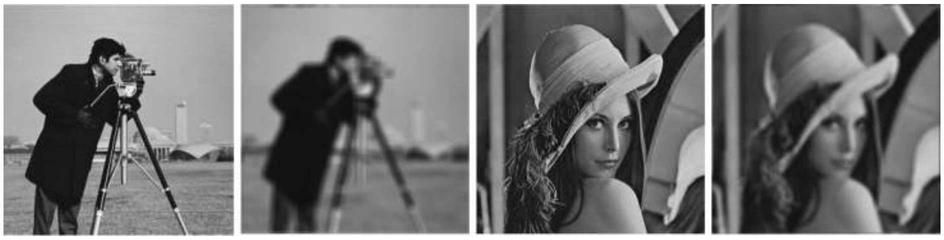

FIGURE 5.5. Original and degraded images for TV-l^2. From left to right: original Cameraman (256×256), blurred and noised Cameraman, original Lena (512×512), blurred and noised Lena.

By setting $x = (x_1, x_2) := (w, y)$, $\theta_1(w) := \frac{1}{2}\|Gw - w^0\|^2$, $\theta_2(y) := \tau\||\nabla y|\|_1$, $A_1 := I$, $A_2 := -I$ and $b := 0$, the ADM scheme (1.3) is also applicable to (5.18) and the resulting subproblems can be solved as follows:

- The w-subproblem is
$$\begin{aligned} w^{k+1} =& \arg\min_{w} \Big\{ \|Gw - w^0\|^2 + \beta \big\| w - y^k - \frac{\lambda^k}{\beta} \big\|^2 \Big\} \\ \Leftrightarrow& (\beta I + G^T G) w = \beta y^k + \lambda^k + G^T w^0, \end{aligned}$$
which could be solved by faster solvers such as FFT.
- The y-subproblem is
$$y^{k+1} = \arg\min_{y} \Big\{ \tau\||\nabla y|\|_1 + \frac{\beta}{2} \big\| y - w^{k+1} + \frac{\lambda^k}{\beta} \big\|^2 \Big\},$$
whose closed-form solution is not available. Efficient algorithms in the literature such as that in [14] can be applied to solve this subproblem iteratively.

As we have shown, we can reformulate the TV-l^p model (5.11) in a special case of (1.1) with either $m = 2$ or $m = 3$. The goodness of considering a reformulation with a smaller m (i.e., $m = 2$) is that the benchmark ADM scheme (1.3) becomes applicable. The curse of this approach, however, is that usually there is at least one resulting subproblem that is too hard to have a closed-form solution and thus it requires internal iterations to pursue an approximated solution (except for FTVd for TV-l^2). Recall the analysis of the resulting subproblems for the methods of C-ADMM, C-SALSA and SALSA. Meanwhile, the advantage of considering a reformulation with a larger m (i.e., $m = 3$) is that the resulting subproblems can be much alleviated such that they are all simple enough to have closed-form solutions.

The tested images are Camerman.png (256×256) and Lena.jpg (512×512). We degrade the Cameraman image by blurring it with the 10×10 uniform kernel and adding zero-mean white Gaussian noise with the standard deviation 0.003; we degrade the Lena image by blurring it with the 13×13 uniform kernel and adding zero-mean white Gaussian noise with the standard deviation 0.0025. See Figure 5.5 for the original images and degraded images.

Recall that there are some subproblems without closed-form solutions for C-ADMM, C-SALSA and SALSA, and we need to apply certain iterative schemes to solve these subproblems. For C-ADMM, the authors recommend applying a Newton method in [55] to compute the involved projection onto an ellipse, and for

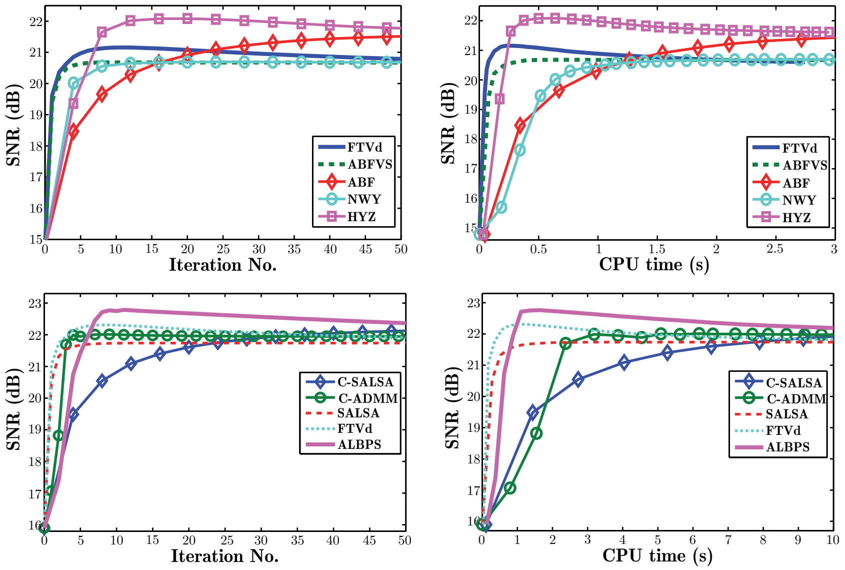

FIGURE 5.6. Evolutions of SNR w.r.t iterations and computing time for TV-l^2. Upper row: Cameraman. Lower row: Lena.

SALSA and C-SALSA, the authors suggest applying the projection method in [14]. We use the same methods for these subproblems, and as suggested by [55], SALSA, C-SALSA and C-ADMM all perform 5 iterations for their internal iterations.

Finally, each of the mentioned methods has some parameters involved and the numerical efficiency of these methods depends on the choices of their respective parameters. In the following, for each method, we label its involved parameters by the same letters as its original paper where the method is proposed. For example, for FTVd there are one penalty parameter β and one trade-off parameter τ; for SALSA there are one penalty parameter μ and one trade-off parameter τ; for C-SALSA there are two penalty parameters μ_1 and μ_2; for C-ADMM there is a penalty β; for ALBPS, there are penalty parameters β_1 and β_2 (recall (5.1)) and a relaxation parameter γ at the correction. So, we have to specify the choices of their respective parameters in order to compare them numerically. In fact, for all the mentioned methods except for FTVd, we test the Cameraman image with various values of the involved parameters and we empirically determine appropriate values for these parameters. Our tuned results are: $\tau = 5\times 10^{-4}$ and $\mu = 10^{-3}$ for SALSA; $\mu_1 = 10^3$ and $\mu_2 = 10^2$ for C-SALSA; $\beta = 10$ for C-ADMM; $\beta_1 = 10$, $\beta_2 = 10^4$ and $\gamma = 1.6$ for ALBPS. For FTVd, we run the original code with tuned parameters which was downloaded from the authors' webpage. Hence, the parameters are chosen as originally: $\tau = 5 \times 10^4$ and $\beta = 10$. For our numerical experiments, we start the iterations with the degraded images. Therefore, the initial value of SNR is 14.78dB for the Cameraman image and 15.9dB for the Lena image.

In Figure 5.6, we plot the evolutions of SNR values with respect to iterations (within 50 iterations) and computing time in seconds for these methods. The curves in Figure 5.6 show that for each tested method, its restored SNR value turns to be stable after about 30 iterations, and the improvement over this stable value is very

Table 5.2. Numerical results for TV-l^2 image restoration.

	Cameraman			Lena		
	Iter	CPU	SNR	Iter	CPU	SNR
C-SALSA	53	4.33	21.67	49	18.09	22.13
C-ADMM	35	4.86	20.54	26	17.93	21.92
SALSA	22	1.39	20.31	22	6.70	20.94
FTVd	21	0.83	20.47	21	3.87	21.72
ALBPS	13	0.63	22.16	13	1.69	22.74

little even if the iteration keeps running. Thus, we can regard the restored stable SNR value as the asymptotically optimal value, or equivalently, the effectiveness, of a method. Recall that the capability of achieving a higher SNR value reflects a better quality of the restored image for one method. Therefore, the higher the asymptotically optimal SNR value is, the better the method is. In practice, as popularly used in the literature of image processing, we can adopt the stopping criterion

$$\text{Tol} := \frac{\|w^{k+1} - w^k\|^2}{\|w^k\|^2} < 10^{-7}, \tag{5.19}$$

which measures the relative change of restored images. Empirically, it is reliable to regard that the asymptotically optimal SNR value of a method is approached when (5.19) is satisfied. In Table 5.2, for each tested method, we report the number of iterations ('Iter') and the computing time in seconds ('CPU') when the criterion (5.19) and the required SNR value are satisfied. In other words, the 'Iter' and 'CPU' in Table 5.2 reflect how fast a method can achieve its own asymptotically optimal SNR value when the stopping criterion (5.19) is used.

According to Figure 5.6 and Table 5.2, we find that ALBPS is attractive and promising in the sense that it is able to achieve the asymptotically optimal SNR value with the fastest speed. In terms of the restored SNR value, C-SALSA is almost as effective as ALBPS, while it requires much more time to achieve its asymptotically optimal SNR values.

We have also implemented these methods to solve the cases of (5.11) with $p = 1$ and $p = \infty$, and obtained numerical conclusions similar as those of the case $p = 2$. Thus, for succinctness, we do not report the numerical results for TV-l^1 and TV-l^∞.

6. Conclusions

We propose a splitting method for the convex minimization problem with linear constraints and a separable objective function which is the sum of many individual functions without coupled variables. Our idea is to decompose the subproblems generated by the classical augmented Lagrangian method into some smaller and easier subproblems. With this decomposition, the resulting subproblems can exploit individually properties of the functions in the objective, and they could be easy enough to have a closed-from solutions for many applications. In addition, all the decomposed subproblems can be solved simultaneously. We apply the proposed method to solve some applications arising in the area of image processing and compare it numerically with some existing methods. The numerical results verify the efficiency and easy implementation of the proposed method.

Acknowledgments

We are grateful to the anonymous referees for their valuable suggestions which have helped to improve the presentation of this paper substantially.

References

[1] Manya V. Afonso, José M. Bioucas-Dias, and Mário A. T. Figueiredo, *Fast image recovery using variable splitting and constrained optimization*, IEEE Trans. Image Process. **19** (2010), no. 9, 2345–2356, DOI 10.1109/TIP.2010.2047910. MR2798930 (2011k:94012)

[2] Manya V. Afonso, José M. Bioucas-Dias, and Mário A. T. Figueiredo, *An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems*, IEEE Trans. Image Process. **20** (2011), no. 3, 681–695, DOI 10.1109/TIP.2010.2076294. MR2799178

[3] Hédy Attouch, Luis M. Briceño-Arias, and Patrick L. Combettes, *A parallel splitting method for coupled monotone inclusions*, SIAM J. Control Optim. **48** (2009/10), no. 5, 3246–3270, DOI 10.1137/090754297. MR2599918 (2011c:90121)

[4] Johnathan M. Bardsley, Sarah Knepper, and James Nagy, *Structured linear algebra problems in adaptive optics imaging*, Adv. Comput. Math. **35** (2011), no. 2-4, 103–117, DOI 10.1007/s10444-011-9172-9. MR2827082

[5] Stephen Becker, Jérôme Bobin, and Emmanuel J. Candès, *NESTA: a fast and accurate first-order method for sparse recovery*, SIAM J. Imaging Sci. **4** (2011), no. 1, 1–39, DOI 10.1137/090756855. MR2765668 (2011m:90087)

[6] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: Numerical methods*, Prentice-Hall, Englewood Cliffs, New Jersey, NJ, 1989.

[7] E. Blum and W. Oettli, *Mathematische Optimierung*, Springer-Verlag, Berlin, 1975. Grundlagen und Verfahren; Mit einem Anhang "Bibliographie zur Nichtlinearer Programmierung"; Ökonometrie und Unternehmensforschung, No. XX. MR0418921 (54 #6956)

[8] N. Bose and K. Boo, *High-resolution image reconstruction with multisensors*, Int. J. Imag. Syst. Tech., **9** (1998), pp. 294–304.

[9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn., **3** (2010), pp. 1–122.

[10] Jian-Feng Cai, Raymond H. Chan, and Mila Nikolova, *Two-phase approach for deblurring images corrupted by impulse plus Gaussian noise*, Inverse Probl. Imaging **2** (2008), no. 2, 187–204, DOI 10.3934/ipi.2008.2.187. MR2395140 (2009c:94002)

[11] Jian-Feng Cai, Stanley Osher, and Zuowei Shen, *Linearized Bregman iterations for compressed sensing*, Math. Comp. **78** (2009), no. 267, 1515–1536, DOI 10.1090/S0025-5718-08-02189-3. MR2501061 (2010e:65086)

[12] Jian-Feng Cai, Stanley Osher, and Zuowei Shen, *Linearized Bregman iterations for frame-based image deblurring*, SIAM J. Imaging Sci. **2** (2009), no. 1, 226–252, DOI 10.1137/080733371. MR2486529 (2010h:94041)

[13] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright, *Robust principal component analysis?*, J. ACM **58** (2011), no. 3, Art. 11, 37, DOI 10.1145/1970392.1970395. MR2811000 (2012f:65064)

[14] Antonin Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision **20** (2004), no. 1-2, 89–97, DOI 10.1023/B:JMIV.0000011320.81911.38. Special issue on mathematics and image analysis. MR2049783 (2005m:49058)

[15] Tony F. Chan and Selim Esedoḡlu, *Aspects of total variation regularized L^1 function approximation*, SIAM J. Appl. Math. **65** (2005), no. 5, 1817–1837, DOI 10.1137/040604297. MR2177726 (2006k:94003)

[16] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky, *Rank-sparsity incoherence for matrix decomposition*, SIAM J. Optim. **21** (2011), no. 2, 572–596, DOI 10.1137/090761793. MR2817479 (2012m:90128)

[17] Gong Chen and Marc Teboulle, *A proximal-based decomposition method for convex minimization problems*, Math. Programming **64** (1994), no. 1, Ser. A, 81–101, DOI 10.1007/BF01582566. MR1274173 (95e:90069)

[18] Patrick L. Combettes and Valérie R. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Model. Simul. **4** (2005), no. 4, 1168–1200 (electronic), DOI 10.1137/050626090. MR2203849 (2007g:94016)

[19] J. Duchi, S. Gould, and D. Koller, *Projected subgradient methods for learning sparse Gaussian*, Conference on Uncertainty in Artificial Intelligence (UAI 2008), 2008.

[20] Jonathan Eckstein and B. F. Svaiter, *General projective splitting methods for sums of maximal monotone operators*, SIAM J. Control Optim. **48** (2009), no. 2, 787–811, DOI 10.1137/070698816. MR2486094 (2009k:47143)

[21] F. Facchinei and J. S. Pang, *Finite-dimensional variational inequalities and complementarity problems*, Volumes I and II, Berlin, Springer Verlag, 2003.

[22] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximations*, Comput. Math. Appl., **2** (1976), pp. 17–40.

[23] R. Glowinski and A. Marrocco, *Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité, d'une classe de problèmes de Dirichlet non linéaires* (French, with Loose English summary), Rev. Française Automat. Informat. Recherche Opérationnelle RAIRO Analyse Numérique **9** (1975), no. R-2, 41–76. MR0388811 (52 #9645)

[24] D. Goldfarb and S. Q. Ma, and K. Scheinberg, *Fast alternating linearization methods for minimizing the sum of two convex functions*, Math. Program., (2012). DOI 10.1007/s10107-012-0530-2.xx

[25] Donald Goldfarb and Shiqian Ma, *Fast multiple-splitting algorithms for convex optimization*, SIAM J. Optim. **22** (2012), no. 2, 533–556, DOI 10.1137/090780705. MR2968865

[26] Tom Goldstein and Stanley Osher, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sci. **2** (2009), no. 2, 323–343, DOI 10.1137/080725891. MR2496060 (2010e:65087)

[27] Deren Han and Xiaoming Yuan, *A note on the alternating direction method of multipliers*, J. Optim. Theory Appl. **155** (2012), no. 1, 227–238, DOI 10.1007/s10957-012-0003-z. MR2983116

[28] Per Christian Hansen, James G. Nagy, and Dianne P. O'Leary, *Deblurring images*, Fundamentals of Algorithms, vol. 3, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006. Matrices, spectra, and filtering. MR2271138 (2008d:94007)

[29] Bing-Sheng He, *Parallel splitting augmented Lagrangian methods for monotone structured variational inequalities*, Comput. Optim. Appl. **42** (2009), no. 2, 195–212, DOI 10.1007/s10589-007-9109-x. MR2471396 (2010b:90152)

[30] B. S. He, X. L. Fu, and Z. K. Jiang, *Proximal-point algorithm using a linear proximal term*, J. Optim. Theory Appl. **141** (2009), no. 2, 299–319, DOI 10.1007/s10957-008-9493-0. MR2500836 (2010i:90128)

[31] B. S. He, M. Tao, M. H. Xu, and X. M. Yuan, *Alternating directions based contraction method for linearly constrained separable convex programming problems*, Optimization, **62** (2013), no. 4, 573–596. MR3056148

[32] B. S. He, M. Tao, and X. M. Yuan, *A splitting method for separable convex programming*, IMA J. Num. Anal., to appear.

[33] Bingsheng He, Min Tao, and Xiaoming Yuan, *Alternating direction method with Gaussian back substitution for separable convex programming*, SIAM J. Optim. **22** (2012), no. 2, 313–340, DOI 10.1137/110822347. MR2968856

[34] Magnus R. Hestenes, *Multiplier and gradient methods*, J. Optimization Theory Appl. **4** (1969), 303–320. MR0271809 (42 #6690)

[35] M. Y. Hong and Z. Q. Luo, *On the linear convergence of the alternating direction method of multipliers*, manuscript, 2012.

[36] B. Huang, S. Q. Ma and D. Goldfarb, *Accelerated linearized Bregman method*, Under review in SIAM J. Imaging Sci., 2011.

[37] Y. M. Huang, M.K. Ng, and Y. W. Wen, *Fast image restoration methods for impulse and Gaussian noise removal*, IEEE Signal Process. Lett., **16** (2009), pp. 457–460.

[38] H. Hwang and A. Haddad, *Adaptive median filters: new algorithms and results*, IEEE Trans. Image Process., **4** (1995), pp. 499–502.

[39] Z. K. Jiang and X. M. Yuan, *New parallel descent-like method for solving a class of variational inequalities*, J. Optim. Theory Appl. **145** (2010), no. 2, 311–323, DOI 10.1007/s10957-009-9619-z. MR2630080 (2011j:49011)

[40] R. M. Larsen, *PROPACK-Software for large and sparse SVD calculations*, Available from http://sun.stanford.edu/srmunk/PROPACK/.

[41] B. Martinet, *Régularisation d'inéquations variationnelles par approximations successives* (French), Rev. Française Informat. Recherche Opérationnelle **4** (1970), no. Ser. R-3, 154–158. MR0298899 (45 #7948)

[42] Michael K. Ng, Pierre Weiss, and Xiaoming Yuan, *Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods*, SIAM J. Sci. Comput. **32** (2010), no. 5, 2710–2736, DOI 10.1137/090774823. MR2684734 (2011i:65065)

[43] Mila Nikolova, *A variational approach to remove outliers and impulse noise*, J. Math. Imaging Vision **20** (2004), no. 1-2, 99–120, DOI 10.1023/B:JMIV.0000011920.58935.9c. Special issue on mathematics and image analysis. MR2049784 (2005b:94006)

[44] Stanley Osher, Yu Mao, Bin Dong, and Wotao Yin, *Fast linearized Bregman iteration for compressive sensing and sparse denoising*, Commun. Math. Sci. **8** (2010), no. 1, 93–111. MR2655902 (2011c:90098)

[45] Y. G. Peng, A. Ganesh, J. Wright, W. L. Xu, and Y. Ma, *Robust alignment by sparse and low-rank decomposition for linearly correlated images*, IEEE Trans. Pattern Anal. Mach. Intel. **34** (2012), 2233–2246.

[46] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, Optimization (Sympos., Univ. Keele, Keele, 1968), Academic Press, London, 1969, pp. 283–298. MR0272403 (42 #7284)

[47] W. K. Pratt, *Digital Image Processing: PIKS Inside*, 3rd Edition, John Wiley & Sons, Inc, 2001.

[48] Z. W. Qin, D. Goldfarb and S. Q. Ma, *An alternating direction method for total variation denoising*, manuscript, 2011.

[49] R. Tyrrell Rockafellar, *Convex analysis*, Princeton Mathematical Series, No. 28, Princeton University Press, Princeton, N.J., 1970. MR0274683 (43 #445)

[50] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D., **60** (1992), pp. 259–268.

[51] S. Setzer, G. Steidl, and T. Tebuber, *Deblurring Poissonian images by split Bregman techniques*, J. Vis. Commun. Image R., **21** (2010), pp. 193–199.

[52] Jonathan E. Spingarn, *Partial inverse of a monotone operator*, Appl. Math. Optim. **10** (1983), no. 3, 247–265, DOI 10.1007/BF01448388. MR722489 (85b:47059)

[53] Min Tao and Xiaoming Yuan, *Recovering low-rank and sparse components of matrices from incomplete and noisy observations*, SIAM J. Optim. **21** (2011), no. 1, 57–81, DOI 10.1137/100781894. MR2765489 (2011m:90089)

[54] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight, *Sparsity and smoothness via the fused lasso*, J. R. Stat. Soc. Ser. B Stat. Methodol. **67** (2005), no. 1, 91–108, DOI 10.1111/j.1467-9868.2005.00490.x. MR2136641

[55] Ewout van den Berg and Michael P. Friedlander, *Probing the Pareto frontier for basis pursuit solutions*, SIAM J. Sci. Comput. **31** (2008/09), no. 2, 890–912, DOI 10.1137/080714488. MR2466141 (2010b:90086)

[56] Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang, *A new alternating minimization algorithm for total variation image reconstruction*, SIAM J. Imaging Sci. **1** (2008), no. 3, 248–272, DOI 10.1137/080724265. MR2486032 (2010e:68198)

[57] Pierre Weiss, Laure Blanc-Féraud, and Gilles Aubert, *Efficient schemes for total variation minimization under constraints in image processing*, SIAM J. Sci. Comput. **31** (2009), no. 3, 2047–2080, DOI 10.1137/070696143. MR2516143 (2010f:65117)

[58] Wotao Yin, *Analysis and generalizations of the linearized Bregman model*, SIAM J. Imaging Sci. **3** (2010), no. 4, 856–877, DOI 10.1137/090760350. MR2735964 (2011j:68172)

[59] Xiaoqun Zhang, Martin Burger, Xavier Bresson, and Stanley Osher, *Bregmanized nonlocal regularization for deconvolution and sparse reconstruction*, SIAM J. Imaging Sci. **3** (2010), no. 3, 253–276, DOI 10.1137/090746379. MR2679428 (2011m:65135)

[60] Xiaoqun Zhang, Martin Burger, and Stanley Osher, *A unified primal-dual algorithm framework based on Bregman iteration*, J. Sci. Comput. **46** (2011), no. 1, 20–46, DOI 10.1007/s10915-010-9408-8. MR2753250 (2012f:90120)

School of Mathematical Science, Nanjing Normal University, Nanjing 210023, People's Republic of China
E-mail address: handeren@njnu.edu.cn

Corresponding author. Department of Mathematics, Hong Kong Baptist University, Kowloon, Hong Kong, People's Republic of China
E-mail address: xmyuan@hkbu.edu.hk

School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, People's Republic of China
E-mail address: wenxing84@gmail.com