J. Nonlinear Var. Anal. 5 (2021), No. 3, pp. 403-420
Available online at http://jnva.biemdas.com
https://doi.org/10.23952/jnva.5.2021.3.05

LOW PATCH-RANK IMAGE DECOMPOSITION USING ALTERNATING MINIMIZATION ALGORITHMS

LULU ZHAO, CHENG CHEN, TINGXIA LU, ZHIYUAN ZHANG, HONGJIN HE*

Department of Mathematics, Hangzhou Dianzi University, Hangzhou, 310018, China

Abstract. Cartoon-texture image decomposition, which refers to the problem of decomposing an image into a cartoon part and a texture component, is one of the most fundamental problems in image processing. In this paper, we are concerned with the low patch-rank enhanced image decomposition model, which is a convex but nonsmooth optimization problem that could not be solved directly by traditional gradient-based optimization algorithms. Accordingly, we introduce two unconstrained reformulations to the underlying low patch-rank optimization model. Furthermore, by exploiting favorable structures of the resulting reformations, we propose two easily implementable alternating minimization algorithms, whose subproblems have closed-form solutions. Compared to the state-of-the-art multi-block alternating direction method of multipliers and its variants, our proposed algorithms enjoy simpler iterative schemes and lower memory requirements for saving computing time. A series of numerical experiments further support the promising performance of the proposed approaches.

Keywords. Alternating minimization algorithm; Image decomposition; Low rank; Alternating direction method of multipliers.

1. INTRODUCTION

Image decomposition plays a fundamental role in image processing and computer vision, such as image inpainting [1], pattern recognition [2, 3], biomedical engineering [4, 5, 6], and many others. Generally, image decomposition refers to the task of decomposing a natural image into a so-called *cartoon* part corresponding to the geometrical part or sketchy approximation of the image, and a *texture* component characterizing the oscillating part or small scale special patterns of the image. Therefore, a given image $f \in \mathbb{R}^{m\times n}$ can be represented as a composition of a cartoon part $u \in \mathbb{R}^{m\times n}$ and a texture component $v \in \mathbb{R}^{m\times n}$, i.e.,

$$f = u + v. \tag{1.1}$$

Accordingly, image decomposition can be described as the process of finding u and v such that both components satisfy (1.1). Actually, when f, u, v are stacked into $(m \times n := l)$-dimensional vectors, image decomposition model (1.1) is an under-determined linear system, which has infinitely many solutions so that it is incredibly difficult to find a perfect decomposition of an

*Corresponding author.

E-mail addresses: zhllu01@163.com (L. Zhao), chencheng@hdu.edu.cn (C. Chen), lutx2020@163.com (T. Lu), zhang510zg@hdu.edu.cn (Z. Zhang), hehjmath@hdu.edu.cn (H. He).

Received February 11, 2021; Accepted March 10, 2021.

image. Moreover, when a given image is corrupted by some degraded operators (denoted by K, which is assumed to be a linear operator in this paper) or noises, i.e.,

$$f = K(u+v) + \text{noise}, \tag{1.2}$$

it becomes more challenging than (1.1) to find an ideal pair of u and v due to the ill-posedness of (1.2). Mathematically, the cartoon part can be described as a piecewise smooth (or a piecewise constant) function, while the texture component is commonly oscillating. Such two mathematical properties encourage researchers to develop approaches to find an ideal decomposition of an image.

In the literature, it has been well documented that optimization approaches for image decomposition have received much considerable attention in the communities of image processing and optimization, e.g., see [7, 8, 9, 10, 11, 12, 13] and references therein. As we know, one of the most popular optimization models is the so-named low patch-rank enhanced minimization model proposed by Schaeffer and Osher in [12], which takes the form

$$\min_{u,v} \ \tau_1 \| |\nabla u| \|_1 + \tau_2 \|Pv\|_* + \frac{\tau_3}{2} \|K(u+v) - f\|^2, \tag{1.3}$$

where τ_i $(i = 1,2,3)$ are positive tuning parameters; $\nabla := \begin{pmatrix} \nabla_1 \\ \nabla_2 \end{pmatrix}$ denotes the first-order derivative operator (throughout this paper, we will employ the widely used isotropic total variation (TV) defined by $(|\nabla u|)_i = \sqrt{(\nabla_1 u)_i^2 + (\nabla_2 u)_i^2}$, e.g., see [14]); $\|\cdot\|_1$ and $\|\cdot\|_*$ represent the ℓ_1-norm and nuclear norm (i.e., the sum of all singular values of a matrix), respectively; in particular, $\|\cdot\|$ corresponds to the standard Frobenius norm of a matrix; and P is the patch map realigning v into a low-rank matrix, and we refer the reader to [12] for some detailed promising properties of P.

Clearly, the low patch-rank minimization model (1.3) is a convex but nonsmooth optimization problem. From the computational perspective, it is usually not an easy task to design a fast and easily implementable algorithm for nonsmooth optimization problems. However, it is fortunate that the nonsmooth model (1.3) has some promising properties that could be exploited in algorithmic design, since the proximity functions associated with the ℓ_1-norm and nuclear norm have closed-form solutions. Therefore, Schaeffer and Osher [12] judiciously employed the so-called *split Bregman method* [15], which is indeed the state-of-the-art two-block alternating direction method of multipliers (ADMM) [16], to find a solution of (1.3). It is notable that the application of the split Bregman method to (1.3) efficiently exploits the favorable structure of the objective. However, their first subproblem contains u and v simultaneously, that is, both u and v are treated as a bigger block variable, thereby resulting in an augmented large-scale linear system. To alleviate the computational burden caused by the coupled variables u and v (i.e., the smooth fidelity term in (1.3)), Han et al. [8] skillfully introduced one more auxiliary variable to decouple the objective function and obtained a three-block separable optimization model, which can be efficiently solved by the three-block ADMM. However, there is no theoretical guarantee to the global convergence of the direct application ADMM under a general convex assumption. Here, we refer the reader to [17] for a counter example showing that the multi-block ADMM is not necessarily convergent for general convex models. Therefore, Han et al. [8] introduced an ADMM-based splitting method to find a solution of the resulting three-block model. Certainly, some recently developed ADMM-based variants equipped with one more correction step or a (partially) parallel decomposition for multi-block separable models are also applicable to their model, e.g., see [8, 18, 19, 20, 21, 22] and references therein. Taking a close look at these splitting methods, it can be easily observed that all of them are designed based upon the augmented Lagrangian method. When applying these splitting methods to problem (1.3), there are accordingly several (at least one) Lagrangian multipliers associated to the underlying linear constraints (see [8, 9]), which are produced artificially for the purpose of separating the objective function. As a consequence, it often requires some more memory storage to implement these splitting methods due to the extra Lagrangian multipliers. As we know, it has been verified empirically that augmented Lagrangian-based methods work better than penalty methods for linearly constrained (two-block) convex optimization problems to achieve higher accuracy. However, we observe that moderate accuracy solutions are sufficient for image processing in many cases, which motivates us to consider whether the augmented Lagrangian-based splitting methods are the best choices for image decomposition. Moreover, can we design an easily implementable algorithm which still exploits the favorable structure of (1.3)?

In this paper, we are interested in designing algorithms for solving the low patch-rank minimization problem (1.3). Specifically, we first introduce some auxiliary variables to separate the objective function. Then, we follow the philosophy behind the penalty method to reformulate the resulting linearly constrained problems as unconstrained forms. Due to the weakly coupled objective, we can employ the alternating spirit to update variables in a sequential order. Correspondingly, we propose two alternating minimization algorithms for the resulting weakly coupled convex optimization models. It is remarkable that our algorithms are globally convergent under general convex assumptions. Moreover, both proposed algorithms are easily implementable in the sense that they have no extra correction steps, and their subproblems have closed-form solutions, thereby saving memory storage so that they run fast. Finally, a series of computational results on pure image decomposition, image inpainting and deblurring demonstrate that our algorithms perform better than the state-of-the-art augmented Lagrangian-based splitting methods in terms of taking less computing time, which also supports the idea of this paper.

This paper is divided into five parts. In Section 2, we introduce three auxiliary variables to fully separate the objective of (1.3), and then reformulate it as an unconstrained optimization problem with three block variables. Accordingly, we employ the alternating minimization algorithm to the resulting problem, in addition to proving its global convergence. To further reduce the memory storage, in Section 3, we introduce two auxiliary variables to simplify the nonsmooth terms and also reformulate it as an unconstrained problem. Then, we propose a partially linearized alternating minimization algorithm and establish its global convergence. In Section 4, we conduct the numerical performance of our algorithms on pure cartoon-texture image decomposition, image inpainting and deblurring. Finally, the conclusion is given in Section 5 to complete this paper.

2. Three-Block Model and Algorithm

In this section, we first introduce three auxiliary variables to fully separate the objective of (1.3), and then reformulate it as an unconstrained three-block optimization problem. Then, we describe the iterative schemes of alternating minimization algorithm for the resulting three-block model. Finally, we prove that the algorithm is globally convergent.

2.1. **Model and alternating minimization algorithm.** Taking a look at (1.3), we can observe that the main difficulty of algorithmic implementation is caused by the nonsmooth terms due to the appearances of ∇ and P. So, as used in [12], we can introduce x and y to simplify the ℓ_1 and nuclear norm, respectively. However, such an approach will result in an augmented large-scale linear system for u and v. In this subsection, we follow the way used in [8] to introduce one more auxiliary variable z to extract $(u+v)$ from the smooth term, i.e.,

$$\begin{aligned} \min \quad & \tau_1 \||x|\|_1 + \tau_2 \|y\|_* + \frac{\tau_3}{2}\|Kz - f\|^2 \\ s.t. \quad & x = \nabla u, \\ & y = Pv, \\ & z = u + v. \end{aligned} \tag{2.1}$$

Clearly, model (2.1) is a linearly constrained optimization problem, and the benchmark solver is augmented Lagrangian method. However, as we mentioned, there will be three Lagrangian multipliers associated to the three equality constraints. Therefore, unlike the approach in [8], we directly reformulate (2.1) as an unconstrained form

$$\min_{u,v,x,y,z} H(u,v,x,y,z), \tag{2.2}$$

where

$$\begin{aligned} H(u,v,x,y,z) = \tau_1 \||x|\|_1 \ + \tau_2 \|y\|_* + \frac{\tau_3}{2}\|Kz - f\|^2 + \frac{\beta_1}{2}\|\nabla u - x\|^2 \\ + \frac{\beta_2}{2}\|Pv - y\|^2 + \frac{\beta_3}{2}\|u + v - z\|^2 \end{aligned} \tag{2.3}$$

with β_i $(i = 1,2,3)$ being positive penalty parameters. Apparently, model (2.2) is a weakly coupled convex optimization problem, which can be solved in an alternating order via $u \to v \to x \to y \to z$, and the algorithm is also called *alternating minimization algorithm* (AMA, see [23, Ch. 14]). Specifically, for given (v^k, x^k, y^k, z^k), the concrete iterative scheme of AMA reads as

$$\left\{ \begin{array}{ll} u^{k+1} \in \arg\min\limits_u \ H(u, v^k, x^k, y^k, z^k), & \text{(2.4a)} \\ v^{k+1} \in \arg\min\limits_v \ H(u^{k+1}, v, x^k, y^k, z^k), & \text{(2.4b)} \\ x^{k+1} \in \arg\min\limits_x \ H(u^{k+1}, v^{k+1}, x, y^k, z^k), & \text{(2.4c)} \\ y^{k+1} \in \arg\min\limits_y \ H(u^{k+1}, v^{k+1}, x^{k+1}, y, z^k), & \text{(2.4d)} \\ z^{k+1} \in \arg\min\limits_z \ H(u^{k+1}, v^{k+1}, x^{k+1}, y^{k+1}, z). & \text{(2.4e)} \end{array} \right.$$

Actually, it is noteworthy that we can solve the v- and x-subproblems (also y- and z-subproblems) simultaneously, since both subproblems are fully separable. Hence, the five variables u, v, x, y, z are not fully coupled, that is why we call (2.2) a weakly coupled optimization model.

Below, we elaborate the specific iterative schemes for all subproblems.

- The u-subproblem. We see that (2.4a) can be simplified as follows:

$$u^{k+1} = \arg\min_u \left\{ \frac{\beta_1}{2}\|\nabla u - x^k\|^2 + \frac{\beta_3}{2}\|z^k - u - v^k\|^2 \right\},$$

which amounts to finding u^{k+1} satisfying the following linear system

$$(\beta_1\nabla^\top\nabla+\beta_3 I)u=\beta_1\nabla^\top x^k+\beta_3(z^k-v^k). \tag{2.5}$$

Generally, (2.5) can be easily handled by the *fast Fourier transform* (FFT) if periodic boundary condition is exploited to the first-order derivative operator ∇, or by the *discrete cosine transform* (DCT) if reflective boundary condition is exploited. Throughout this paper, we use I to denote an identity matrix.

- The v-subproblem. Dropping the constant terms of (2.4b) yields the following optimization problem:

$$v^{k+1}=\arg\min_v\left\{\frac{\beta_2}{2}\|Pv-y^k\|_2^2+\frac{\beta_3}{2}\|v+u^{k+1}-z^k\|^2\right\}. \tag{2.6}$$

In accordance with the isometry property of the patch mapping P (see [12]), the simplified v-subproblem (2.6) has the closed-form solution:

$$v^{k+1}=\frac{1}{\beta_2+\beta_3}\left[\beta_2P^\top y^k+\beta_3(z^k-u^{k+1})\right]. \tag{2.7}$$

- The x-subproblem. It is clear that (2.4c) reduces to

$$x^{k+1}=\arg\min_x\left\{\tau_1\||x|\|_1+\frac{\beta_1}{2}\|x-\nabla u^{k+1}\|^2\right\},$$

which also has a closed-form solution given by the well-known shrinkage operator, i.e.,

$$x^{k+1}=\mathrm{shrink}\left(\nabla u^{k+1},\frac{\tau_1}{\beta_1}\right), \tag{2.8}$$

where the shrinkage operator $\mathrm{shrink}(\cdot,\cdot)$ is defined as

$$\mathrm{shrink}(a,t):=\frac{a}{|a|}\odot\max\{0,\mathrm{abs}(a)-t\},\quad a\in\mathbb{R}^l\times\mathbb{R}^l,\ t>0,$$

with 'max' and the absolute value function '$\mathrm{abs}(a)$' being component-wise, and '$\odot$' representing the component-wise product. Here, $\frac{a_i}{(|a|)_i}=0$ when $(|a|)_i=0$.

- The y-subproblem. We can simplify (2.4d) as a nuclear norm minimization problem

$$y^{k+1}=\arg\min_y\left\{\tau_2\|y\|_*+\frac{\beta_2}{2}\|y-Pv^{k+1}\|^2\right\}. \tag{2.9}$$

Recall the *singular value thresholding* (SVT) operator, i.e.,

$$\mathrm{SVT}(M,t):=U\mathrm{shrink}(\Sigma,t)V^\top,\quad t>0,$$

where $U\Sigma V^\top$ is the singular value decomposition of matrix M. As a consequence, the explicit solution of (2.9) is given by

$$y^{k+1}=\mathrm{SVT}\left(Pv^{k+1},\frac{\tau_2}{\beta_2}\right). \tag{2.10}$$

- The z-subproblem. The last subproblem (2.4e) is a least squares problem:

$$z^{k+1}=\arg\min_z\left\{\frac{\tau_3}{2}\|Kz-f\|^2+\frac{\beta_3}{2}\|u^{k+1}+v^{k+1}-z\|^2\right\},$$

which amounts to finding a z^{k+1} being a solution of the following linear system:

$$(\tau_3 K^\top K + \beta_3 I)z = \tau_3 K^\top f + \beta_3 (u^{k+1} + v^{k+1}). \tag{2.11}$$

When K is the identity matrix, diagonal matrix or down-sampling matrix, the coefficient matrix on the left of (2.11) is diagonal matrix, which can be solved directly. When K is a blurring matrix, it can be solved effectively by FFT or DCT.

With the above derivations for all subproblems, it is easily seen that v^{k+1} and x^{k+1} can be updated simultaneously, since both updating iterative schemes have no coupled information. Similarly, we can also compute y^{k+1} and z^{k+1} in a parallel way. Formally, we summarize the details of our proposed method in Algorithm 1.

Algorithm 1 Alternating minimization algorithm for solving (2.2).

```
Initialization Given starting points $(v^0, x^0, y^0, z^0)$ and penalty parameters $\beta_1, \beta_2, \beta_3 > 0$.
Repeat
Update $u^{k+1}$ via solving (2.5).
Update $v^{k+1}$ and $x^{k+1}$ by (2.7) and (2.8), respectively.
Update $y^{k+1}$ and $z^{k+1}$ via (2.10) and (2.11), respectively.
$k \leftarrow k+1$
Until stopping criterion is satisfied.
```

2.2. **Convergence theorem.** In this subsection, we show that Algorithm 1 is globally convergent. It can be easily verified that our model (2.2) is a convex optimization model, we accordingly follow the analysis of [23] (see also [24]) to prove the global convergence and $O(1/k)$ convergence rate.

As we mentioned in Section 2.1, both v and x (also y and z) are fully separable, we can gainfully group them into one block. Letting $\boldsymbol{w}_1 = u$, $\boldsymbol{w}_2 = (x, v)$, $\boldsymbol{w}_3 = (y, z)$, we can easily reformulate (2.2) as a three-block coupled optimization model, i.e.,

$$\min_{u,v,x,y,z} H(u,v,x,y,z) = \min_{\boldsymbol{w}_1,\boldsymbol{w}_2,\boldsymbol{w}_3} \left\{ \sum_{i=1}^{3} \theta_i(\boldsymbol{w}_i) + F(\boldsymbol{w}_1,\boldsymbol{w}_2,\boldsymbol{w}_3) \right\}, \tag{2.12}$$

where

$$\begin{cases} \theta_1(\boldsymbol{w}_1) = 0, & \text{(2.13a)} \\ \theta_2(\boldsymbol{w}_2) = \tau_1 \||x|\|_1, & \text{(2.13b)} \\ \theta_3(\boldsymbol{w}_3) = \tau_2 \|y\|_* + \dfrac{\tau_3}{2}\|Kz - f\|^2, & \text{(2.13c)} \\ F(\boldsymbol{w}_1,\boldsymbol{w}_2,\boldsymbol{w}_3) = \dfrac{1}{2}\|A_1\boldsymbol{w}_1 + A_2\boldsymbol{w}_2 + A_3\boldsymbol{w}_3\|^2 & \text{(2.13d)} \end{cases}$$

with

$$A_1 = \begin{pmatrix} \sqrt{\beta_3} I \\ \mathbf{0} \\ \sqrt{\beta_1}\nabla \end{pmatrix}, \quad A_2 = \begin{pmatrix} \mathbf{0} & \sqrt{\beta_3} I \\ \mathbf{0} & \sqrt{\beta_2} P \\ -\sqrt{\beta_1} I & \mathbf{0} \end{pmatrix}, \quad A_3 = \begin{pmatrix} \mathbf{0} & -\sqrt{\beta_3} I \\ -\sqrt{\beta_2} I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}. \tag{2.14}$$

It is clear that (2.12) is a convex optimization problem and A_i's are full column rank matrices. Moreover, it is clear that the coupled function F defined by (2.13d) is continuously differentiable. In what follows, we denote $\boldsymbol{w} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3\}$ and $F(\boldsymbol{w}) = F(\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3)$ for notational simplicity. Obviously, the gradient of $F(\cdot)$ is block coordinatewise Lipschitz continuous, i.e.,

$$\begin{cases} \|\nabla_1 F(\boldsymbol{w}_1,\boldsymbol{w}_2,\boldsymbol{w}_3) - \nabla_1 F(\boldsymbol{w}_1',\boldsymbol{w}_2,\boldsymbol{w}_3)\| \le \lambda_{\max}(A_1^\top A_1)\|\boldsymbol{w}_1 - \boldsymbol{w}_1'\| = \sigma_1\|\boldsymbol{w}_1 - \boldsymbol{w}_1'\|, \\ \|\nabla_2 F(\boldsymbol{w}_1,\boldsymbol{w}_2,\boldsymbol{w}_3) - \nabla_2 F(\boldsymbol{w}_1,\boldsymbol{w}_2',\boldsymbol{w}_3)\| \le \lambda_{\max}(A_2^\top A_2)\|\boldsymbol{w}_2 - \boldsymbol{w}_2'\| = \sigma_2\|\boldsymbol{w}_2 - \boldsymbol{w}_2'\|, \\ \|\nabla_3 F(\boldsymbol{w}_1,\boldsymbol{w}_2,\boldsymbol{w}_3) - \nabla_3 F(\boldsymbol{w}_1,\boldsymbol{w}_2,\boldsymbol{w}_3')\| \le \lambda_{\max}(A_3^\top A_3)\|\boldsymbol{w}_3 - \boldsymbol{w}_3'\| = \sigma_3\|\boldsymbol{w}_3 - \boldsymbol{w}_3'\|. \end{cases}$$

where $\lambda_{\max}(\cdot)$ represents the largest eigenvalue of a give matrix and

$$\begin{cases} \sigma_1 := \lambda_{\max}(A_1^\top A_1) = \lambda_{\max}(\beta_1 \nabla^\top \nabla + \beta_3 I), \\ \sigma_2 := \lambda_{\max}(A_2^\top A_2) = \max\left\{\lambda_{\max}(\beta_2 P^\top P + \beta_3 I), \beta_1\right\}, \\ \sigma_3 := \lambda_{\max}(A_3^\top A_3) = \max\{\beta_2, \beta_3\}. \end{cases}$$

Consequently, the gradient of F, denoted by $\nabla F(\boldsymbol{w}) = (\nabla_1 F(\boldsymbol{w}), \nabla_2 F(\boldsymbol{w}), \nabla_3 F(\boldsymbol{w}))$, is then Lipschitz continuous with a constant σ such that

$$\|\nabla F(\boldsymbol{w}) - \nabla F(\boldsymbol{w}')\| \le \sigma\|\boldsymbol{w} - \boldsymbol{w}'\|.$$

where $\sigma \le \sigma_1 + \sigma_2 + \sigma_3$. Accordingly, it follows from [23, Chapter 14] that our Algorithm 1 has the following convergence theorem. Here, we omit its proof for the conciseness of this paper, and refer the reader to [23, Theorems 14.9 and 14.11] for details.

Theorem 2.1. *The sequence $\{\boldsymbol{w}^k\}$ generated by Algorithm 1 is bounded, and any limit point of the sequence is an optimal solution to (2.12). Moreover, for all $k \ge 2$, we have*

$$H(\boldsymbol{w}^k) - H(\boldsymbol{w}^*) \le \max\left\{\left(\frac{1}{2}\right)^{\frac{(k-1)}{2}}\left(H(\boldsymbol{w}^0) - H(\boldsymbol{w}^*)\right), \frac{72\sigma\kappa^2}{k-1}\right\},$$

where $\boldsymbol{w}^* := (\boldsymbol{w}_1^*, \boldsymbol{w}_2^*, \boldsymbol{w}_3^*)$ is an optimal solution of (2.12), $H(\boldsymbol{w}) := F(\boldsymbol{w}) + \sum_{i=1}^3 \theta_i(\boldsymbol{w}_i)$, and κ is a constant given by

$$\kappa = \max_{\boldsymbol{w} \in \mathbb{R}^{5l}} \max_{\boldsymbol{w}^* \in \mathscr{W}^*} \left\{\|\boldsymbol{w} - \boldsymbol{w}^*\| : H(\boldsymbol{w}) \le H(\boldsymbol{w}^0)\right\}$$

with $\mathscr{W}^*$ being the optimal solution set of (2.12). In addition, an ε-optimal solution is obtained after at most

$$\max\left\{\frac{2}{\ln(2)}\left(\ln\left(H(\boldsymbol{w}^0) - H(\boldsymbol{w}^*)\right) + \ln(1/\varepsilon)\right), \frac{72\sigma\kappa^2}{\varepsilon}\right\} + 2,$$

which means Algorithm 1 has $O(1/k)$ convergence rate.

3. Two-Block Model and Algorithm

In the last section, we can see that one more auxiliary variable z can efficiently decouple u and v such that two related subproblems have closed-form solutions. However, one more auxiliary variable requires more memory storage, which motivates us to further reduce the memory requirement. In this section, we only employ two auxiliary variables and reformulate it as an unconstrained problem. Then, we propose a partially linearized AMA.

3.1. **Model and partially linearized AMA.** Based on the linearly constrained model (2.1), we drop the last equality constraint and arrive at a simplified convex minimization problem, i.e.,

$$\begin{aligned} \min_{u,y,v,x} \ & \tau_1 \| |x| \|_1 + \tau_2 \|y\|_* + \frac{\tau_3}{2} \|K(u+v) - f\|^2 \\ \text{s.t.} \ & x = \nabla u, \\ & y = Pv. \end{aligned} \tag{3.1}$$

Like (2.2), we also reformulate (3.1) as an unconstrained optimization problem as follows

$$\min_{u,y,v,x} \ \Phi(u,y,v,x), \tag{3.2}$$

where

$$\Phi(u,y,v,x) = \tau_1 \| |x| \|_1 + \tau_2 \|y\|_* + \frac{\tau_3}{2} \|K(u+v) - f\|^2 + \frac{\beta_1}{2} \|\nabla u - x\|^2 + \frac{\beta_2}{2} \|Pv - y\|^2 \tag{3.3}$$

with β_1 and β_2 being positive penalty parameters. For given (y^k, v^k, x^k), applying the AMA to (3.2) via the order $u \to y \to v \to x$ immediately yields

$$\begin{cases} u^{k+1} \in \arg\min\limits_u \ \Phi(u, y^k, v^k, x^k), & \text{(3.4a)} \\ y^{k+1} \in \arg\min\limits_y \ \Phi(u^{k+1}, y, v^k, x^k), & \text{(3.4b)} \\ v^{k+1} \in \arg\min\limits_v \ \Phi(u^{k+1}, y^{k+1}, v, x^k), & \text{(3.4c)} \\ x^{k+1} \in \arg\min\limits_x \ \Phi(u^{k+1}, y^{k+1}, v^{k+1}, x). & \text{(3.4d)} \end{cases}$$

Hereafter, we derive the concrete iterative schemes for all subproblems.

- The u-subproblem (3.4a) corresponds to the following optimization problem:

$$u^{k+1} \in \arg\min_u \left\{ \frac{\tau_3}{2} \|K(u+v^k) - f\|^2 + \frac{\beta_1}{2} \|\nabla u - x^k\|^2 \right\}, \tag{3.5}$$

  which is equivalent to finding a solution u^{k+1} to the following linear system:

$$(\tau_3 K^\top K + \beta_1 \nabla^\top \nabla) u = \tau_3 K^\top (f - K v^k) + \beta_1 \nabla^\top x^k. \tag{3.6}$$

  We can easily derive the explicit solution of (3.6) when K or ∇ are full column rank. Hence, we can update u^{k+1} by solving (3.6) directly when K is an identity matrix. However, when both K and ∇ are not identity matrices, solving (3.6) directly is possibly not an ideal way even though both K and ∇ perhaps have special structures. To maximally exploit the structures of K or ∇, we accordingly linearize the first smooth term of (3.5) at u^k and arrive at

$$(\tau_3 \gamma_1 I + \beta_1 \nabla^\top \nabla) u = \tau_3 \gamma_1 u^k - \tau_3 (K^\top K u^k + K^\top K v^k - K^\top f) + \beta_1 \nabla^\top x^k, \tag{3.7}$$

  where $\gamma_1 > 0$ is a linearization proximal parameter. Consequently, we update u^{k+1} via solving (3.7) for the cases where K and ∇ are not identity matrices. In this situation, we can obtain u^{k+1} fast by exploiting structure of ∇.

- The y-subproblem (3.4b) can be solved explicitly by SVT operator used in (2.10), i.e.,

$$y^{k+1} = \arg\min_y \left\{ \tau_2 \|y\|_* + \frac{\beta_2}{2} \|y - P v^k\|^2 \right\} = \text{SVT}\left(P v^k, \frac{\tau_2}{\beta_2} \right). \tag{3.8}$$

- The v-subproblem (3.4c) can be simplified as

$$v^{k+1} = \arg\min_v \left\{ \frac{\tau_3}{2}\|K(u^{k+1}+v)-f\|^2 + \frac{\beta_2}{2}\|Pv - y^{k+1}\|^2 \right\},$$

  which is equivalent to finding v^{k+1} such that

$$(\tau_3 K^\top K + \beta_2 I)v^{k+1} = \beta_2 P^\top y^{k+1} + \tau_3(K^\top f - K^\top K u^{k+1}). \tag{3.9}$$

- The x-subproblem (3.4d) can also be solved explicitly by the shrinkage operator, i.e.,

$$x^{k+1} = \arg\min_x \left\{ \tau_1 \| |x| \|_1 + \frac{\beta_1}{2}\|x - \nabla u^{k+1}\|^2 \right\} = \text{shrink}\left(\nabla u^{k+1}, \frac{\tau_1}{\beta_1}\right). \tag{3.10}$$

From the above derivations, we can see that the updates of u and y can be implemented in a parallel way. Similarly, we can compute v and x simultaneously. Therefore, we can group u, y, v, x into two bigger blocks, which will facilitate our convergence analysis in the next subsection. Moreover, we just require two and three starting points, i.e., (v^k, x^k) and (u^k, v^k, x^k) for the cases where K is an identity or blurring matrix and K is a down-sampling matrix, respectively. Since we only linearize the first u-subproblem, we call our method a partially linearized AMA. Below, we summarize the details of our partially linearized AMA in Algorithm 2.

Algorithm 2 Partially linearized AMA for (3.2).

```
1. Initialization Given starting points $(v^0, x^0)$ or $(u^0, v^0, x^0)$, and parameters $\beta_1, \beta_2 > 0$.
2. Repeat
3. Update $u^{k+1}$ by (3.6) when $K$ is the identity or blurring matrix, or by (3.7) when $K$ is a down-sampling matrix;
4. Update $y^{k+1}$ by (3.8);
5. Update $v^{k+1}$ and $x^{k+1}$ by (3.9) and (3.10), respectively;
6. $k \leftarrow k+1$
7. Until stopping criterion is satisfied.
```

3.2. **Convergence theorem.** Like the reformulation (2.12), by letting $\hat{\boldsymbol{w}}_1 = (u, y)$ and $\hat{\boldsymbol{w}}_2 = (v, x)$, we can reformulate (3.2) as a two-block coupled optimization problem as follows

$$\min_{u,y,v,x} \Phi(u,y,v,x) = \min_{\hat{\boldsymbol{w}}_1, \hat{\boldsymbol{w}}_2} \left\{ \hat{\theta}_1(\hat{\boldsymbol{w}}_1) + \hat{\theta}_2(\hat{\boldsymbol{w}}_2) + \hat{F}(\hat{\boldsymbol{w}}_1, \hat{\boldsymbol{w}}_2) \right\}, \tag{3.11}$$

where

$$\begin{cases} \hat{\theta}_1(\hat{\boldsymbol{w}}_1) = \tau_2 \|y\|_*, & (3.12a) \\ \hat{\theta}_2(\hat{\boldsymbol{w}}_2) = \tau_1 \| |x| \|_1, & (3.12b) \\ \hat{F}(\hat{\boldsymbol{w}}_1, \hat{\boldsymbol{w}}_2) = \dfrac{1}{2}\|\hat{A}_1 \hat{\boldsymbol{w}}_1 + \hat{A}_2 \hat{\boldsymbol{w}}_2 - \boldsymbol{b}\|^2, & (3.12c) \end{cases}$$

with

$$\hat{A}_1 = \begin{pmatrix} \sqrt{\tau_3} K & \mathbf{0} \\ \sqrt{\beta_1}\nabla & \mathbf{0} \\ \mathbf{0} & -\sqrt{\beta_2} I \end{pmatrix}, \quad \hat{A}_2 = \begin{pmatrix} \sqrt{\tau_3} K & \mathbf{0} \\ \mathbf{0} & -\sqrt{\beta_1} I \\ \sqrt{\beta_2} P & \mathbf{0} \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} \sqrt{\tau_3} f \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}. \tag{3.13}$$

With the above reformulation, it can be easily seen that the function $\hat{F}$ defined by (3.12c) is also continuously differentiable. Moreover, its gradient $\nabla\hat{F}$ is block coordinatewise Lipschitz continuous, i.e.,

$$\begin{cases} \|\nabla_1 F(\hat{\boldsymbol{w}}_1,\hat{\boldsymbol{w}}_2)-\nabla_1 F(\hat{\boldsymbol{w}}'_1,\hat{\boldsymbol{w}}_2)\| \le \lambda_{\max}(\hat{A}_1^\top\hat{A}_1)\|\hat{\boldsymbol{w}}_1-\hat{\boldsymbol{w}}'_1\| = \hat{\sigma}_1\|\hat{\boldsymbol{w}}_1-\hat{\boldsymbol{w}}'_1\|, \\ \|\nabla_2 F(\hat{\boldsymbol{w}}_1,\hat{\boldsymbol{w}}_2)-\nabla_2 F(\hat{\boldsymbol{w}}_1,\hat{\boldsymbol{w}}'_2)\| \le \lambda_{\max}(\hat{A}_2^\top\hat{A}_2)\|\hat{\boldsymbol{w}}_2-\hat{\boldsymbol{w}}'_2\| = \hat{\sigma}_2\|\hat{\boldsymbol{w}}_2-\hat{\boldsymbol{w}}'_2\|, \end{cases}$$

where

$$\begin{cases} \hat{\sigma}_1 := \lambda_{\max}(\hat{A}_1^\top\hat{A}_1) = \max\left\{\lambda_{\max}(\tau_3 K^\top K),\ \beta_2\right\}, \\ \hat{\sigma}_2 := \lambda_{\max}(\hat{A}_2^\top\hat{A}_2) = \max\left\{\lambda_{\max}(\tau_3 K^\top K+\beta_2 P^\top P),\ \beta_1\right\}. \end{cases}$$

Consequently, letting $\hat{\sigma}_{\min} = \min\{\hat{\sigma}_1,\hat{\sigma}_2\}$, it follows from [23, Theorem 14.15] that Algorithm 2 is globally convergent with $O(1/k)$ convergence rate. Here, we also skip the detailed proof for conciseness of the paper.

Theorem 3.1. *The sequence $\{\hat{\boldsymbol{w}}^k\}$ generated by Algorithm 2 is bounded and any limit point of the sequence is an optimal solution to (3.11). Moreover, for $k \ge 2$, we have*

$$\Phi(\hat{\boldsymbol{w}}^k)-\Phi(\hat{\boldsymbol{w}}^*) \le \max\left\{\left(\frac{1}{2}\right)^{\frac{(k-1)}{2}}\left(\Phi(\hat{\boldsymbol{w}}^0)-\Phi(\hat{\boldsymbol{w}}^*)\right),\ \frac{8\hat{\sigma}_{\min}\hat{\kappa}^2}{k-1}\right\},$$

where $\hat{\boldsymbol{w}}^* := (\hat{\boldsymbol{w}}_1^*,\hat{\boldsymbol{w}}_2^*)$ be an optimal solution of (3.11), $\Phi(\hat{\boldsymbol{w}}) := \hat{F}(\hat{\boldsymbol{w}})+\hat{\theta}_1(\hat{\boldsymbol{w}}_1)+\hat{\theta}_2(\hat{\boldsymbol{w}}_2)$, and the constant $\hat{\kappa}$ is given by

$$\hat{\kappa} := \max_{\hat{\boldsymbol{w}}\in\mathbb{R}^{4l}}\ \max_{\hat{\boldsymbol{w}}^*\in\hat{\mathscr{W}}^*}\{\|\hat{\boldsymbol{w}}-\hat{\boldsymbol{w}}^*\| : H(\hat{\boldsymbol{w}}) \le H(\hat{\boldsymbol{w}}^0)\}$$

with $\hat{\mathscr{W}}^*$ being the optimal solution set of (3.11). In addition, an ε-optimal solution is obtained after at most

$$\max\left\{\frac{2}{\ln(2)}\left(\ln(\Phi(\hat{\boldsymbol{w}}^0)-\Phi(\hat{\boldsymbol{w}}^*))+\ln(1/\varepsilon)\right),\ \frac{8\hat{\sigma}_{\min}\hat{\kappa}^2}{\varepsilon}\right\}+2,$$

which means Algorithm 2 has $O(1/k)$ convergence rate.

4. Numerical Experiments

In this section, we will investigate the numerical performance of Algorithms 1 and 2 on pure cartoon-texture image decomposition, image inpainting and deblurring. To support the main idea of this paper, we also compare our proposed algorithms with state-of-the-art three-block ADMM (‘EADMM’ in short) and the splitting method tailored for image decomposition in [8] (denoted by ‘HKZ’). For simplicity, Algorithms 1 and 2 are denoted by ‘AMA’ and ‘LAMA’, respectively. All numerical experiments are implemented in MATLAB 2014b and were conducted on a Lenovo laptop with Intel(R) Core(TM) i5-6200U CPU @2.30 GHz and 4.00G memory running Microsoft Windows 10. All test images in our experiments are summarized in Figure 1.

Due to the appearance of the linear degraded operator K in (1.3), we conduct three scenarios on K which correspond to three types of applications of cartoon-texture image decomposition, respectively. (i) $K = I$ corresponds to the pure cartoon-texture decomposition of a clean image;

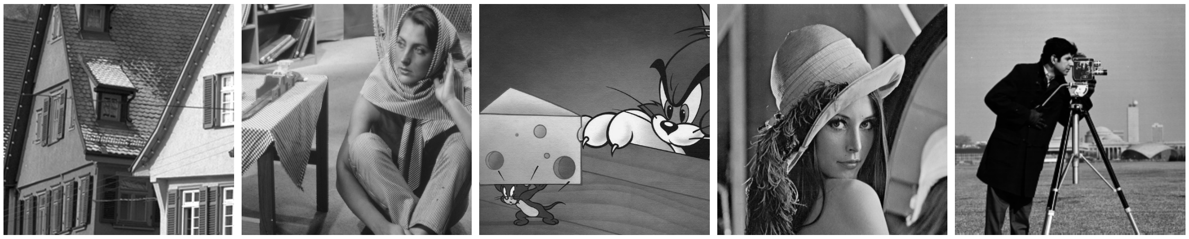

FIGURE 1. Images under test. (a) House 256×256; (b) Barbara 512×512; (c) TomAndJerry 512×512; (d) Lena 512×512; (e) Cameraman 256×256.

(ii) $K=S$ corresponds to the application of cartoon-texture decomposition to image inpainting, where S is a down-sampling operator; (iii) $K=B$ is an application of image decomposition to image deblurring, where B is a blurring matrix. In our experiments, all initial points are taken as zero matrices (or vectors). As used in the literature, the stopping criterion is taken as follows for all experiments:

$$\mathrm{Tol}=\max\left\{\frac{\|u^{k+1}-u^k\|}{\|u^k\|+1},\frac{\|v^{k+1}-v^k\|}{\|v^k\|+1}\right\}\le\varepsilon, \tag{4.1}$$

where ε is positive stopping tolerance controlling the precision of algorithms.

4.1. **Case: $K=I$.** We first consider the pure cartoon-texture decomposition of a clean image. Different parameter values are selected according to different test images. It is documented in [25] that the cartoon and texture components of images are statistically uncorrelated, i.e., the correlation between the cartoon u and the texture v defined by

$$\mathrm{Corr}(u,v):=\mathrm{cov}(u,v)/\sqrt{\mathrm{var}(u)\cdot\mathrm{var}(v)}$$

should be close to zero for an ideal decomposition, where $\mathrm{var}(\cdot)$ and $\mathrm{cov}(\cdot)$ are the variance and covariance of given variables. Therefore, we shall employ $\mathrm{Corr}(u,v)$ to measure the quality of image decomposition for case $K=I$. As suggested in [8], we set the tuning parameters (τ_1,τ_2,τ_3) and penalty parameters $(\beta_1,\beta_2,\beta_3)$ as $(0.01,0.001,2)$ and $(1,1,1)$ respectively for both EADMM and HKZ. The parameters of our AMA are specified as $(\tau_1,\tau_2,\tau_3)=(0.005,0.05,0.2)$ and $(\beta_1,\beta_2,\beta_3)=(0.8,0.5,0.1)$. For parameters of the proposed LAMA, we take $(\tau_1,\tau_2,\tau_3)=(0.005,0.05,0.16)$ and $(\beta_1,\beta_2)=(1,0.3)$. In addition, we set $r=11$ for the patch mapping P and $\varepsilon=10^{-2}$ in (4.1) throughout the case $K=I$. The maximum number of iterations is set 150.

As displayed in Figure 2, the four algorithms can decompose these natural images ideally. A quick glance at these cartoon and texture images illustrates that our proposed algorithms, i.e., both AMA and LAMA, are competitive with the state-of-the-art EADMM and HKZ. Taking a close look at the two images TomAndJerry and Lena, we can observe that both AMA and LAMA achieve a slightly better decomposition than HKZ, since our methods obtain smoother cartoons and clearer textures. To show more numerical details, we further plot the evolution of $\mathrm{Corr}(u,v)$ with respect to iterations and computing time in seconds in Figure 3, respectively.

It can be easily seen from Figure 3 that the values $\mathrm{Corr}(u,v)$ of EADMM was the lowest, which means that its decomposition is better than others. However, our AMA and LAMA can achieve a relatively low correlation by taking fewer iterations and less computing time. Moreover, we can see that LAMA takes less computing time than AMA, since LAMA has less

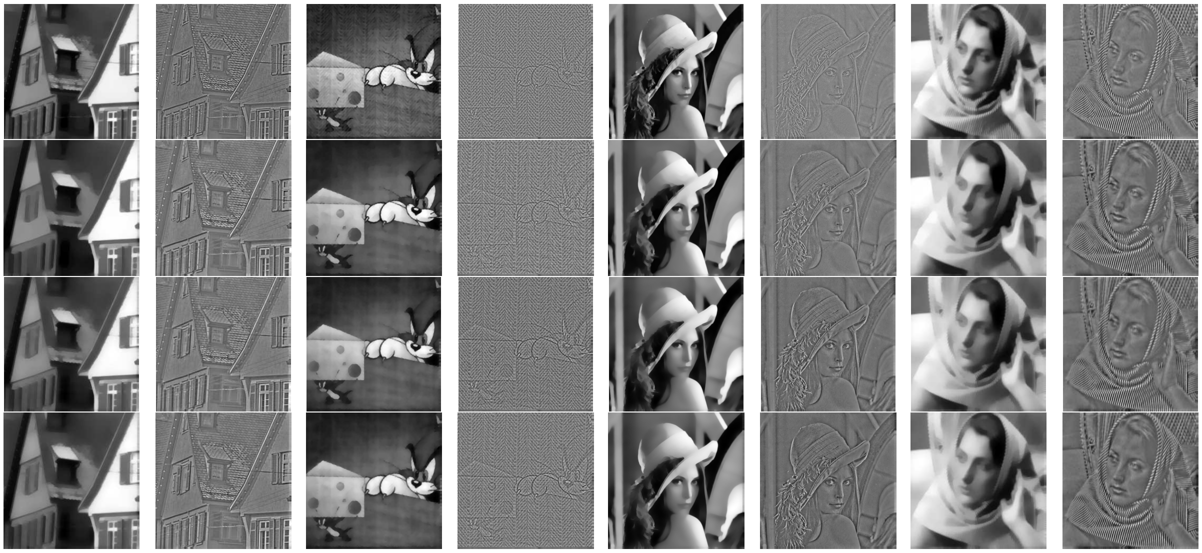

FIGURE 2. Pure cartoon-texture image decomposition on clean images. From to bottom: HKZ, EADMM, AMA, LAMA, respectively.

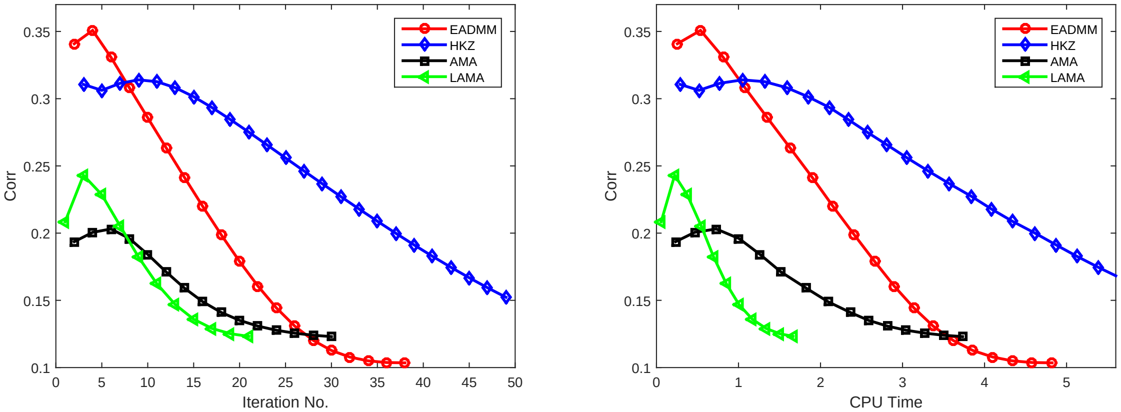

FIGURE 3. Evolutions of correlation with respect to iterations and computing time for the case of K = I

variables than AMA to save memory storage. The preliminary numerical results support our idea of this paper.

4.2. **Case:** $K = S$. Now, we consider an application of image decomposition to image inpainting, which refers to the problem of inpainting an image with missing pixels. In this case, the linear operator S is a down-sampling matrix. We conduct the numerical performance of the four algorithms on three images, i.e., Barbara, Lena, and Cameraman. To measure the quality of a recovered image, we employ the commonly used *signal-to-noise ratio* (SNR), which is given by

$$\mathrm{SNR} = 20\log_{10}\frac{\|f^*\|_2}{\|\hat{f} - f^*\|_2},$$

where $\hat{f}$ is an approximation to the ground truth image f^*. In our experiments, we take $(\tau_1,\tau_2,\tau_3) = (0.05,0.01,2)$, $(\beta_1,\beta_2,\beta_3) = (10,10,0.1)$ for both EADMM and HKZ, (τ_1,τ_2,τ_3) = $(0.006,0.025,1.2)$, $(\beta_1,\beta_2,\beta_3) = (1,1,0.1)$ for AMA, and $(\tau_1,\tau_2,\tau_3) = (0.015,0.01,0.6)$, $(\beta_1,\beta_2) = (20,10)$ for LAMA. Due to the linearization of LAMA, the linearization parameter γ is set to be $\gamma = 1.001$. Besides, we set the size of patch mapping P as $r = 11$ and consider three types of down-sampling matrices. In particular, we generate a 512×512 mask with 0.3 of missing pixels to degrade Lena. All recovered images are summarized in Figure 4.

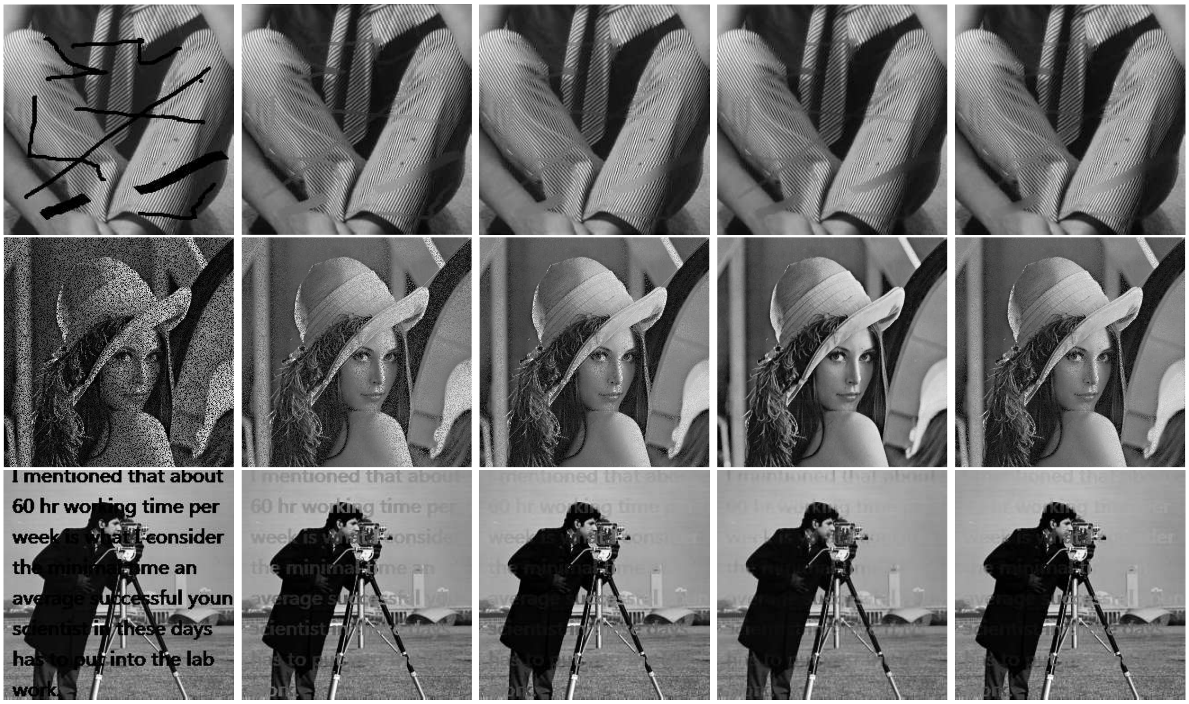

FIGURE 4. Case $K = S$: From left to right: Degraded images and images recovered by EADMM, HKZ, AMA and LAMA, respectively.

It can be seen from Figure 4 that both AMA and LAMA have competitive numerical performance with EADMM and HKZ. Moreover, it seems that AMA and LAMA have slightly better recovering quality on Cameraman image, since EADMM and HKZ have slightly stronger text shadows than our algorithms. To see the iterative behaviors of the algorithm, we further show the evolution of SNR values with respect to iterations and computing time for Barbara in Figure 5, where we set $\varepsilon = 5 \times 10^{-3}$ in (4.1). Clearly, we can see that both AMA and LAMA achieve higher SNR values (i.e., better quality of a recovered image) than HKZ by taking less computing time and fewer iterations. When we further improve the precision of algorithms, i.e., setting $\varepsilon = 5 \times 10^{-4}$ in (4.1), we can see from Figure 6 that the decomposition of our proposed AMA and LAMA to Barbara image can recover more details of the image than both EADMM and HKZ.

The results in Figures 5 and 6 demonstrate that both AMA and LAMA seem to be superior to both EADMM and HKZ for pursuing higher accuracy solutions. Therefore, we report some more numerical details (i.e., number of iterations (iter), computing time in seconds (time), and

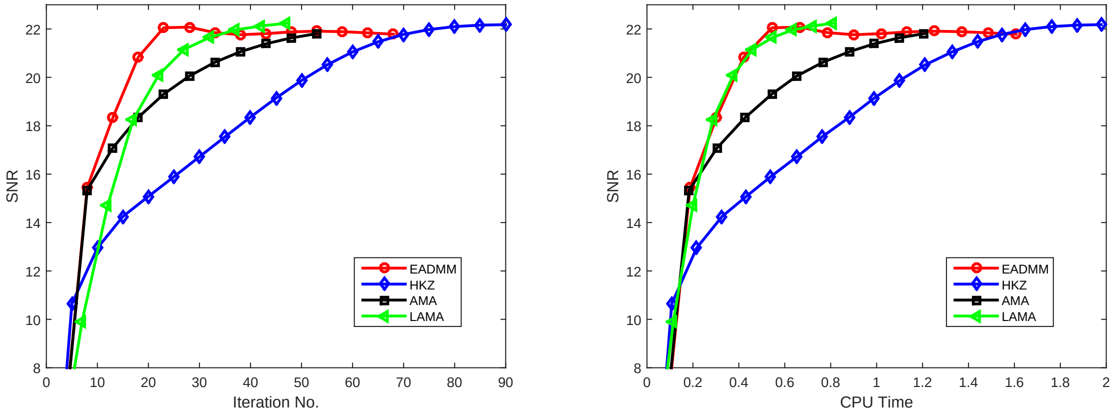

FIGURE 5. Evolution of SNR values with respect to iterations and computing time for the case $K = S$.

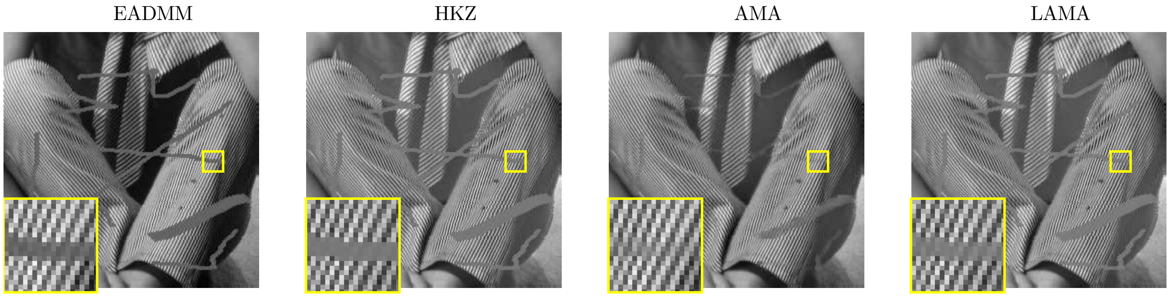

FIGURE 6. Details of the recovered images by EADMM, HKZ, AMA and LAMA.

SNR values) in Table 1, where we investigate three different ε's, i.e., $\varepsilon = \{10^{-2}, 5\times 10^{-3}, 10^{-3}\}$ on Barbara and Lena.

TABLE 1. Numerical results for image inpainting with different stopping tolerances.

Image	Method	$\text{Tol} = 10^{-2}$	$\text{Tol} = 5\times 10^{-3}$	$\text{Tol} = 10^{-3}$
		iter / time / SNR	iter / time / SNR	iter / time / SNR
Barbara	EADMM	33 / 1.12 / 21.85	68 / 1.54 / 21.78	540 / 12.55 / 18.91
	HKZ	70 / 1.61 / 21.77	90 / 1.95 / 22.18	424 / 9.13 / 21.29
	AMA	31 / 0.75 / 20.41	53 / 1.13 / 21.81	404 / 8.71 / 22.58
	LAMA	37 / 0.64 / 21.96	47 / 0.77 / 22.23	395 / 6.96 / 21.75
Lena	EADMM	52 / 6.14 / 21.32	112 / 13.16 / 19.46	574 / 65.98 / 14.66
	HKZ	65 / 7.22 / 23.27	98 / 10.89 / 22.66	585 / 63.84 / 18.41
	AMA	49 / 5.41 / 21.57	115 / 12.46 / 23.01	512 / 54.02 / 27.00
	LAMA	31 / 2.41 / 23.38	59 / 4.38 / 23.12	549 / 40.30 / 19.29

It can be seen that our AMA or LAMA achieve higher SNR values than EADMM and HKZ. More promisingly, AMA and LAMA run faster than EADMM and HKZ to achieve the same SNR values in many cases, which further support the efficiency of our algorithms.

4.3. **Case: $K = B$.** We now turn our attention to the application of image decomposition to image deblurring, which corresponds to the case where $K = B$ is a blurring matrix. Since a blurred image often eliminates the subtle difference of the texture component, it is a challenging task to recover an ideal image from a blurred image. In this subsection, we consider three images, i.e., House, Barbara, and Lena, which were degraded by a convolution with blurring kernel radius 9. For parameters of these methods, we take $(\tau_1,\tau_2,\tau_3) = (0.0005, 0.001, 2)$, $(\beta_1,\beta_2,\beta_3) = (0.1, 0.1, 0.1)$ for EADMM, HKZ, and AMA, and $(\tau_1,\tau_2,\tau_3) = (0.005, 0.001, 4)$, $(\beta_1,\beta_2) = (18, 0.2)$ for LAMA. The size of the patch mapping P is still specified as $r = 11$ and the tolerance is taken as $\varepsilon = 5\times 10^{-3}$. All recovered images are listed in Figure 7, which show that AMA and LAMA can recover better details, for example, the window of House, the tie and scarf of Barbara.

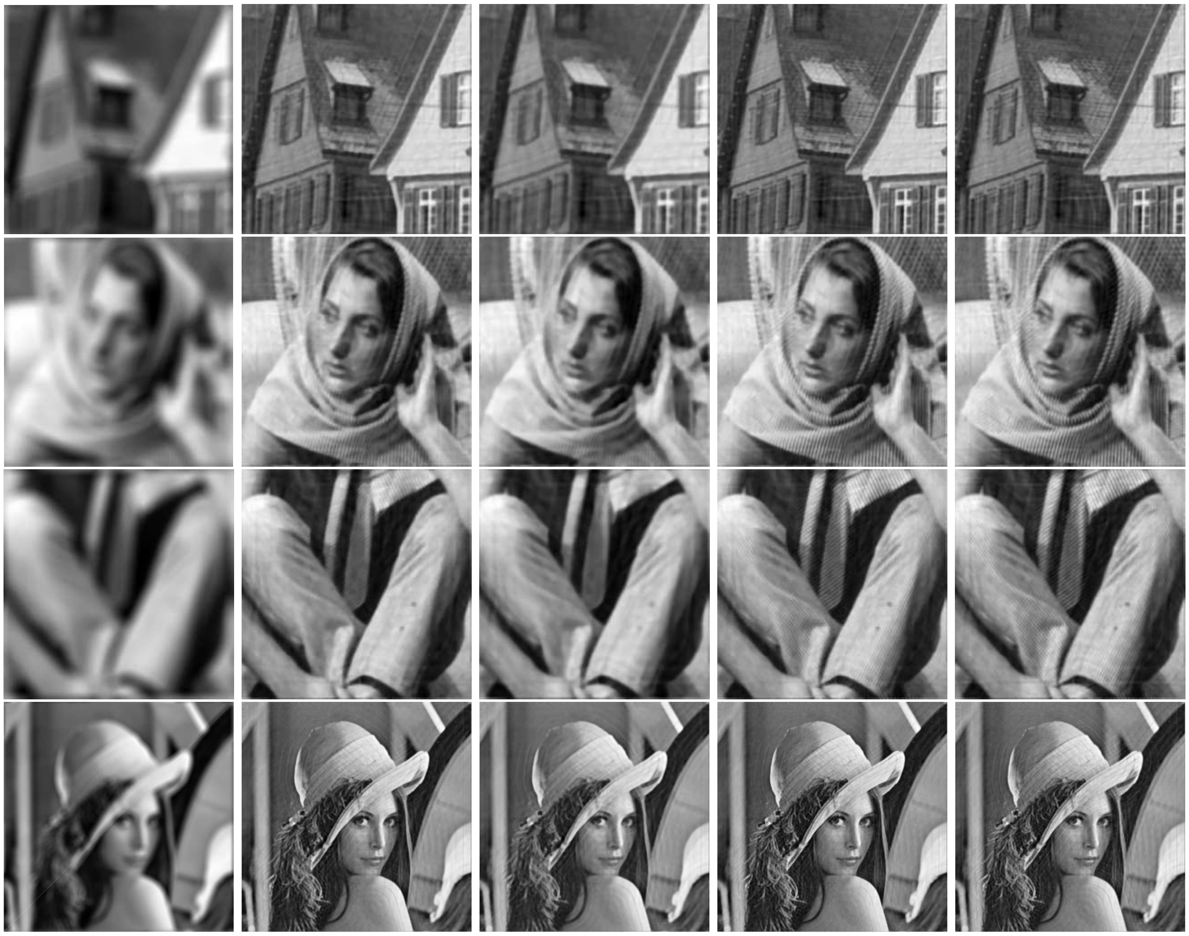

FIGURE 7. Case $K = B$. From left to right: Degraded images and images recovered by EADMM, HKZ, AMA and LAMA, respectively.

As shown in Figure 5, we also plot the evolution of SNR values with respect to computing time in seconds and iterations in Figure 8, which shows that both AMA and LAMA outperform

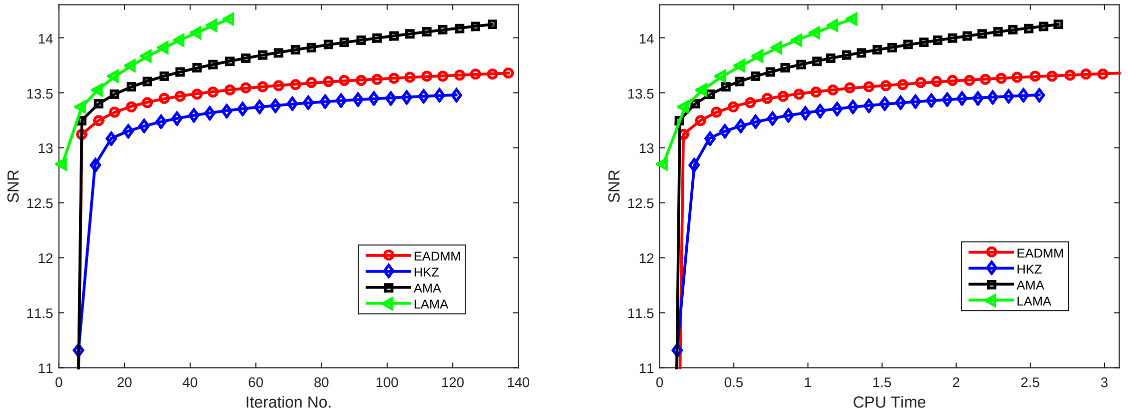

FIGURE 8. Evolution of SNR values with respect to iterations and computing time for the case $K = B$.

both EADMM and HKZ in terms of taking fewer iterations and less computing time to recover better images.

Finally, we also conduct three different stopping tolerances, i.e., $\varepsilon = \{10^{-2}, 5\times 10^{-3}, 10^{-3}\}$ to investigate the performance of these algorithms on image deblurring. All computational results are summarized in Table 2, which also demonstrate that both AMA and LAMA are more efficient than both EADMM and HKZ for image deblurring.

TABLE 2. Numerical results for image deblurring with different stopping tolerances.

Image	Method	Tol $= 10^{-2}$	Tol $= 5\times 10^{-3}$	Tol $= 10^{-3}$
		iter / time / SNR	iter / time / SNR	iter / time / SNR
Barbara	EADMM	63 / 1.61 / 13.56	137 / 3.10 / 13.68	656 / 15.46 / 13.92
	HKZ	57 / 1.31 / 13.36	121 / 2.56 / 13.48	711 / 16.34 / 13.73
	AMA	62 / 1.37 / 13.84	132 / 2.69 / 14.12	602 / 12.68 / 14.82
	LAMA	22 / 0.54 / 13.74	52 / 1.30 / 14.17	419 / 10.43 / 15.90
Lena	EADMM	66 / 7.56 / 21.31	128 / 14.75 / 22.03	654 / 74.84 / 23.07
	HKZ	67 / 7.19 / 19.95	130 / 13.92 / 20.71	646 / 69.85 / 22.31
	AMA	63 / 6.59 / 22.29	121 / 12.75 / 23.03	602 / 63.02 / 24.07
	LAMA	23 / 3.65 / 21.32	43 / 6.44 / 22.26	352 / 52.74 / 24.77
House	EADMM	65 / 1.56 / 16.18	117 / 2.69 / 16.61	576 / 13.64 / 17.58
	HKZ	65 / 1.46 / 15.33	127 / 2.64 / 15.77	558 / 12.68 / 16.81
	AMA	64 / 1.37 / 16.97	114 / 2.37 / 17.51	513 / 11.01 / 18.63
	LAMA	25 / 0.62 / 16.59	47 / 1.18 / 17.34	288 / 7.11 / 19.45

5. CONCLUSION

In this paper, we considered a low patch-rank minimization model for image decomposition. Since there are two nonsmooth terms in the objective function, it is not an easy task to find a solution to the model by straightforward applications of gradient-based methods. To maximally exploit the special structure of the low patch-rank model, we proposed two unconstrained reformulations such that the objective function possesses a weakly coupled property. Correspondingly, we introduced two easily implementable algorithms in the sense that their subproblems have closed-form solutions. A series of numerical experiments illustrated that our approaches have competitive numerical performance with some state-of-the-art ADMM-like algorithms.

Acknowledgments

H. He was supported in part by Natural Science Foundation of Zhejiang Province (No. LY20A010018) and National Natural Science Foundation of China (No. 11771113).

REFERENCES

[1] M. Bertalmio, L. Vese, G. Sapiro, S. Osher, Simultaneous structure and texture image inpainting, IEEE Trans. Image Proces. 12 (2003), 882-889.

[2] V. Singh, A.H. Tewfik, D.B. Ress, Under-sampled functional MRI using low-rank plus sparse matrix decomposition, In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, April 19-24, 2015, pp. 897-901, South Brisbane, Queensland, Australia, 2015.

[3] X. T. Yuan, S.C. Yan, Visual classification with multi-task joint sparse representation, In: The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, 2010, pp. 3493-3500, IEEE Computer Society, 2010.

[4] M. A. Balafar, A. R. Ramli, M. I. Saripan, S. Mashohor, Review of brain MRI image segmentation methods, Artificial Intelligence Rev. 33 (2010), 261-274.

[5] S. X. Tu, S. Zhang, Y. Z. Chen, M. Freedman, B. Wang, J. Xuan, Y. Wang, Automatic tracing and segmentation of rat mammary fat pads in MRI image sequences based on cartoon-texture model, Trans. Tianjin Univ. 15 (2009), 229-235.

[6] H. W. Zhang, X. Q. Ren, Y. Liu, Q.X. Zhou, The application of compressed sensing reconstruction algorithms for MRI of glioblastoma, In: 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI 2017, pp. 1-6, Shanghai, 2017.

[7] Y.R. Fan, T.Z. Huang, T.H. Ma, X.L. Zhao, Cartoon-texture image decomposition via non-convex low-rank texture regularization, J. Franklin Institute 354 (2017), 3170-3187.

[8] D.R. Han, W.W. Kong, W.X. Zhang, A partial splitting augmented lagrangian method for low patch-rank image decomposition, J. Math. Imaging Vision 51 (2015), 145-160.

[9] M. K. Ng, X. Yuan, W. Zhang, Coupled variational image decomposition and restoration model for blurred cartoon-plus-texture images with missing pixels, IEEE Trans. Image Process. 22 (2013), 2233-2246.

[10] S. Ono, T. Miyata, I. Yamada, Cartoon-texture image decomposition using blockwise low-rank texture characterization, IEEE Trans. Image Process. 23 (2014), 1128-1142.

[11] J. L. Starck, M. Elad, D. Donoho, Image decomposition via the combination of sparse representations and a variational approach, IEEE Trans. Image Process. 14 (2005), 1570-82.

[12] H. Schaeffer, S. Osher, A low patch-rank interpretation of texture, SIAM J. Imaging Sci. 6 (2013), 226-262.

[13] Z.Y. Zhang, H.J. He, A customized low-rank prior model for stuctured cartoon-texture image decomposition, Signal Process. Image Commun. Under revision.

[14] L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D, 60 (1992), 227-238.

[15] T. Goldstein, S. Osher, The split Bregman method for ℓ_1 regularized problems, SIAM J. Imaging Sci. 2 (2009), 323-343.

[16] E. Esser, Applications of Lagrangian-based alternating direction methods and connections to split Bregman, CAM Report 09-31, UCLA, 2009.

[17] C.H. Chen, B.S. He, Y.Y. Ye, X.M. Yuan, The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent, Math. Program. Series A 155 (2016), 57-79.

[18] B.S. He, M. Tao, X.M. Yuan, Alternating direction method with Gaussian-back substitution for separable convex programming, SIAM J. Optim. 22 (2012), 313-340.

[19] B.S. He, M. Tao, X.M. Yuan, A splitting method for separate convex programming, IMA J. Numer. Anal. 35 (2014), 394-426.

[20] H. J. He, L.S. Hou, H.K. Xu, A partially isochronous splitting algorithm for three-block separable convex minimization problems, Adv. Comput. Math. 44 (2018), 1091-1115.

[21] H.J. He, D.R. Han, A distributed Douglas-Rachford splitting method for multi-block convex minimization problems, Adv. Comput. Math. 42 (2016), 27-53.

[22] L.S. Hou, H.J. He, J.F. Yang, A partially parallel splitting method for multiple-block separable convex programming with applications to robust PCA, Comput. Optim. Appl. 63 (2016), 273-303.

[23] A. Beck, First-Order Methods in Optimization, SIAM, Philadelphia, 2018.

[24] R. Shefi, M. Teboulle, On the rate of convergence of the proximal alternating linearized minimization algorithm for convex problems, EURO J. Comput. Optim. 4 (2016), 27-46.

[25] J. Aujol, G. Gilboa, T. Chan, S. Osher, Structure-texture image decomposition-modeling, algorithms, and parameter selection, Int. J. Comput. Vision 67 (2006), 111–136.