Noname manuscript No.
(will be inserted by the editor)

Cartoon-Texture Image Decomposition using Least Squares and Low-Rank Regularization

Kexin Li · You-wei Wen · Raymond H. Chan

Received: November 6, 2024

Abstract In this paper, we propose a novel model for the decomposition of cartoon-texture images, which integrates the edge-aware weighted least squares (WLS) with low-rank regularization. Unlike conventional methodologies that depend on total variation-based penalty functions, our model represents cartoon images using an edge-preserving WLS penalty. This approach effectively enhances edges and suppresses texture through iterative updates of an edge-preserving weight matrix. For the texture component, we introduce a low-rank penalty function to capture the structured regularity of texture patterns. By leveraging the repetitive nature of texture, our low-rank models can accurately represent these components. We employ a prediction-correction approach based on a three-block separable alternating direction multiplier method to solve the minimization problem, providing closed-form solutions for all subproblems. We also provide a convergence proof for the proposed algorithm. Numerical experiments validate the efficacy of our proposed method in successfully separating cartoon and texture components while preserving edges.

Keywords Image decomposition · edge-preserving · weighted least squares · low-rank regularization · ADMM.

Mathematics Subject Classification (2020) 68U10 · 94A08 · 90C99 · 65K99

K. Li
School of Statistics and Mathematics, Yunnan University of Finance and Economics, Kunming, Yunnan, P. R. China
E-mail: likx1213@163.com

Y. Wen (Corresponding author)
School of Mathematics and Statistics, Hunan Normal University, Changsha, Hunan 410081, P. R. China
E-mail: wenyouwei@gmail.com

R. Chan
Department of Operations and Risk Management and School of Data Science, Lingnan University, Hong Kong SAR.
E-mail: raymond.chan@ln.edu.hk

1 Introduction

Image decomposition plays an important role in image processing, enabling a detailed and comprehensive analysis of an image's components. By decomposing an image into its distinct components, we can extract valuable information such as edges, contours, and textures. This information is crucial for efficiently performing various image processing tasks in many applications, including image denoising [13,41,49], image fusion [28,52,56], image segmentation [19,50], and more.

In this paper, we specifically focus on decomposing an image into two fundamental components: the cartoon and the texture components. The cartoon component captures the overall structure and global information of the image. It exhibits clear boundaries and smoothly connected components. The texture component represents the high-frequency oscillations within the image. It typically models repetitive structural information. Mathematically, given an observed image $\mathbf{b}$ with size $m \times n$, the decomposition can be expressed as follows:

$$\mathbf{b} = K(\mathbf{u} + \mathbf{v}) + \mathbf{n}, \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^{mn}$ is the cartoon component, $\mathbf{v} \in \mathbb{R}^{mn}$ is the texture component, $\mathbf{n}$ is the additive Gaussian white noise, and $K : \mathbb{R}^{mn} \to \mathbb{R}^{mn}$ is a linear degradation operator. The linear operator has various forms depending on the specific applications. For instance, it can be a binary matrix, effectively representing images with missing pixel values or as a blurring matrix associated with a spatially-invariant point spread function, which accounts for various blurring effects.

Image decomposition aims to extract the values of $\mathbf{u}$ and $\mathbf{v}$ from $\mathbf{b}$. The task of cartoon-texture decomposition can be formulated as a minimization problem:

$$\min_{\mathbf{u},\mathbf{v}} \mathcal{J}(\mathbf{u}, \mathbf{v}) = \frac{1}{2} \|K(\mathbf{u} + \mathbf{v}) - \mathbf{b}\|_2^2 + \lambda \, \phi(\mathbf{u}) + \mu \, \psi(\mathbf{v}), \tag{2}$$

where $\phi(\mathbf{u})$ and $\psi(\mathbf{v})$ are penalty functions representing prior knowledge of the cartoon and texture components, respectively. The regularization parameters λ and μ play a crucial role in governing the balance between the contributions of the cartoon and texture components. The key to successfully decomposing an image into cartoon and texture components is to select appropriate penalty functions $\phi(\mathbf{u})$ and $\psi(\mathbf{v})$ and develop numerical methods to solve the minimization problem. Consequently, many researchers have investigated different regularization functions to optimize these choices; see [33,38,43,44]

1.1 Regularization Term for Cartoon Images

Cartoon images are typically characterized by piecewise constant or smoothing regions accompanied by distinct corners and edges. A cartoon-oriented prior function, denoted as $\phi(\mathbf{u})$, is essential to capture these characteristics effectively. This function aims to preserve the sharpness of edges while preventing any blurring effects. A prevalent choice for the prior function for cartoon images $\mathbf{u}$ is the total variation (TV) norm, defined as $\|\nabla \mathbf{u}\|_1$ [37]. The TV norm has obtained significant popularity in various image processing tasks due to its ability to preserve edges while suppressing unwanted noise [7,8,23]. It is well known that the conventional TV norm introduces staircase artifacts, resulting in unnatural and visually unpleasing transitions between image regions. To overcome this limitation, an alternative approach is to utilize the non-convex L_p norm

($0 < p < 1$), denoted as $\|\nabla \mathbf{u}\|_p^p$ [22,26,34,35]. This non-convex regularization term better preserves edges in the image while suppressing noise, thereby reducing the occurrence of staircase artifacts and leading to more visually appealing results. However, using non-convex functions introduces numerical challenges in finding the global minimum, as they may have multiple local minima.

Another trend in image processing is the utilization of quadratic penalty functions. These functions offer the advantage of straightforward computation of gradients and Hessian matrices, facilitating the numerical optimization process. Being convex, they ensure the existence of a unique solution to the optimization problem. Consequently, the computational process is greatly simplified, and the overall efficiency is improved. The origins of quadratic penalty functions can be traced back to the Tikhonov regularization [14]. However, the Tikhonov regularization technique is susceptible to over-smoothing of images, leading to blurred edges. To mitigate these limitations, a generalized Tikhonov regularization quadratic function has been proposed to preserve edges in the image while suppressing noise and smoothing regions [2].

The quadratic penalty function can be expressed as the square of the weighted L_2 norm of the image as follows:

$$\phi(\mathbf{u}) = \|WL\mathbf{u}\|_2^2 . \tag{3}$$

Here L represents a combination of the discretized horizontal and vertical derivatives, denoted as L_x and L_y, respectively, *i.e.,* $L = (L_x^T, L_y^T)^T$, and the matrix W is a diagonal matrix whose diagonal elements are comprised of the weight vector $\mathbf{w}$. Utilizing the approach of solving a sequence of weighted least squares (WLS) problems, Gazzola *et al.* [12] proposed a novel inner-outer iterative algorithm. This algorithm constructs weights that comprehensively incorporate edge information generated throughout the iterative process.

Currently, some methods for solving TV-type regularization functions involve converting them into weighted quadratic functions, where the weights $\mathbf{w}_k$ are iteratively updated based on previous solutions. Subsequently, a series of WLS problems are solved. For instance, Vogel and Oman [42] proposed a lagged diffusivity fixed-point iteration tailored for TV-norm regularized image restoration. Non-convex L_p-TV norm penalty functions can effectively capture piecewise smooth image characteristics, particularly for $0 < p < 1$. These functions are often transformed into weighted quadratic functions using majorization-minimization or iterative reweighting to handle computational challenges in non-convex optimization [10,47,46,26]. This raises the question of whether the penalty function expressed in (3) could be initially employed as an image prior, emphasizing the selection of the weight vector $\mathbf{w}$. In this paper, we adopt the penalty function in (3) and focus on the principles of choosing an appropriate weight vector $\mathbf{w}$.

1.2 Regularization Term for Texture Images

Early studies employed the L_2 norm to represent texture, treating texture characteristics and image noise as oscillatory components. Subsequently, Meyer introduced the G-space concepts, serving as the dual of the bounded variation space and the G norm in [29]. Notably, the G norm yielded smaller texture values than the L_2 norm, suggesting its suitability for characterizing texture priors. Subsequently, image decomposition methods based on the G norm have been widely studied [39,45]. However, a key limitation of the G norm is its non-integral expression, which complicates the derivation of the corresponding Euler-Lagrange equation for the minimization problem. This poses numerical challenges in developing effective solution methods.

Texture is a ubiquitous visual pattern repeatedly appearing in images containing stripes, checks, grids, and other forms. The repetition of these patterns creates a structured regularity, which enables the utilization of low-rank models to characterize texture [53]. Indeed, the theory of low-rank matrix recovery has attracted considerable attention in image processing [16,55,51]. Zhang *et al.* introduced the low-rank prior for texture in matrix recovery and completion, revolutionizing the approach to cartoon-texture image decomposition [53]. Subsequently, Schaeffer and Osher proposed the low patch-rank (LPR) model, which employed block low-rank to explain the texture of images, introducing the innovative concept of the texture norm [38]. Ono *et al.* furthered this understanding by employing the matrix block nuclear norm (BNN) to depict texture information with local similarity [31]. Zhang *et al.* [54] adopted a simplified global nuclear norm to characterize global patterned texture structures elegantly. With the advancement of non-convex regularization, researchers have increasingly applied non-convex low-rank penalties to texture characterization. For instance, Fan *et al.* [11] characterized the low-rank property of texture by using a non-convex Log-det function, successfully integrating the self-similarity of the texture and the piecewise smoothness of the cartoon. Meanwhile, Yan *et al.* [48] utilized a generalized non-convex low-rank minimization method for image decomposition, further enhancing the separation of cartoon and texture components.

1.3 Proposed Model and Contributions

Currently, numerous image decomposition techniques often result in blurred edges when extracting structural components.To mitigate this limitation, we introduce a novel edge-preserving cartoon-texture image decomposition method. This approach leverages iterative WLS and low-rank regularization to separately model cartoon and texture components, as formulated in the following:

$$\min_{\mathbf{u},\mathbf{v}} \quad \frac{1}{2}\left\|K(\mathbf{u}+\mathbf{v})-\mathbf{b}\right\|_2^2+\frac{\lambda}{2}\left\|WL\mathbf{u}\right\|_2^2+\mu\left\|\mathbf{v}\right\|_*. \tag{4}$$

Here, W represents the edge-aware weight matrix, which serves to preserve edges and prevent the introduction of artifacts. The $\|\cdot\|_*$ norm, known as the nuclear norm, is the sum of all singular values of a matrix [38]. Given that images are represented as vectors by column concatenation, there exists a one-to-one correspondence between images and vectors. Therefore, we slightly misuse the definition and use $\|\mathbf{a}\|_*$ to represent the nuclear norm of the matrix transformed from the vector.

We remark that if $\mu = 0$, the optimization problem in (4) is consistent with the one in [12]. Although Gazzola *et al.*'s method [12] demonstrates excellent performance in image restoration, its decomposition performance for texture component separation remains unsatisfactory. We illustrate this with an example of separate synthetic images shown on the left of Fig. 1. We also show the decomposition results obtained by the method in [12] and our approach in Fig. 1. We observed that the method proposed by Gazzola *et al.* [12] produces satisfactory results for cartoon components with simple structures. However, its performance degrades when dealing with more complex structures, resulting in blurred edges and misclassifying important structural information into texture components. In contrast, our approach demonstrates robustness and

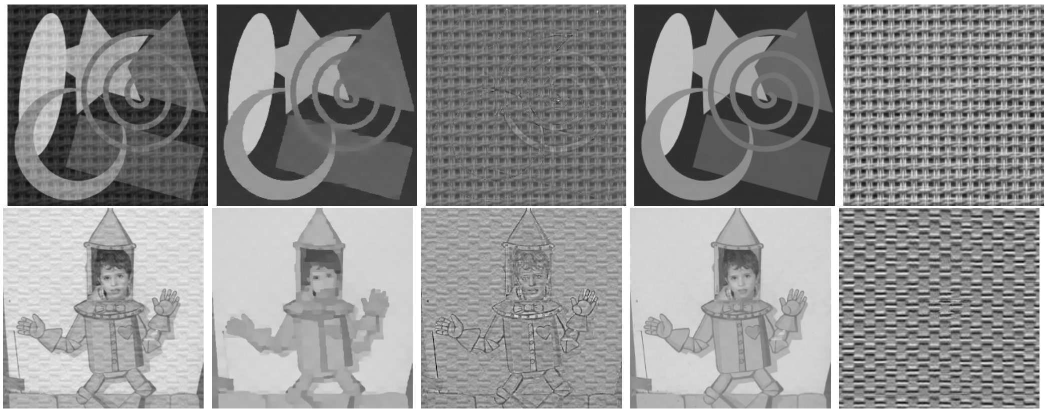


Fig. 1 Comparison of the decomposition results in two types of synthetic images: "Geometry" (having a simplified cartoon component structure) and "Boy" (having a relatively complex edge structure). From left to right: the synthetic images, the cartoon and texture image obtained by [12] (second and third columns) and our method (fourth and fifth columns).

consistency, effectively separating cartoon and texture components even in complex structures. This suggests that our method excels in image decomposition tasks, particularly in maintaining edge integrity.

The main contributions of this paper are summarized as follows:

- We propose a novel image decomposition model integrating WLS with low-rank regularization. This model employs a simplified quadratic penalty term to handle cartoon components efficiently and a low-rank penalty term to capture texture details. To solve the minimization problem, we utilize an ADMM-based approach, ensuring closed-form solutions for all sub-problems.
- We introduce a new method for obtaining an edge-aware weight matrix within the penalty function of cartoon images, distinguishing it from traditional fixed-weight methodologies. Our approach employs an iterative and adaptive approach, facilitating dynamic adjustments of the weight matrix by the guided cartoon image. This edge-aware weighting strategy enhances edges while effectively suppressing texture, improving image quality. We also show the convergence of the weight matrix.
- Numerical experiments indicate that the proposed method successfully and effectively separates cartoon and texture components, and the edges of the cartoon image are well-preserved.

The rest of this paper is organized as follows: Section 2 introduces some related works. Subsequently, Section 3 presents our edge-preserving image decomposition model along with the corresponding numerical algorithm. The convergence analysis of the proposed algorithm is discussed in Section 4. Section 5 gives numerical experiments to validate the effectiveness of our proposed method. Finally, a short conclusion is given in Section 6.

2 Related Works

2.1 Image Restoration via WLS Model

The image decomposition problem is closely related to the task of image restoration, which aims to remove noise and distortions from degraded images. Image restoration is an ill-posed problem, and regularization techniques are employed to stabilize its solution. One of the most common and well-known forms of regularization is Tikhonov regularization. The standard Tikhonov regularization often introduces artifacts, such as ringing effects near sharp intensity transitions. To address this limitation, the generalized Tikhonov regularization is utilized, which can be formulated as follows:

$$\min_{\mathbf{u}} \frac{1}{2} \left\| K\mathbf{u} - \mathbf{b} \right\|_2^2 + \frac{\lambda}{2} \left\| WL\mathbf{u} \right\|_2^2 . \tag{5}$$

Here W is the diagonal matrix with the pre-determined weight $\mathbf{w}$. The key criterion for selecting the weight coefficients $\mathbf{w}$ is as follows: in flat regions of the image, the weight coefficients should be significant to penalize high-frequency information (such as noise); at edges, the weight coefficients should be slight to preserve edge details and avoid blurring. Gazzola *et al.* [12] proposed a new inner-outer iterative algorithm for solving problem (5), which constructs weights by iteratively integrating image edge information.

2.2 Nuclear Norm Model

Texture in an image exhibits a distinct structure and pattern, typically showing regularity and repetitiveness [53, 27]. The whole texture image can be seen as composed of a few basic textures. Especially in human-constructed buildings or structures, numerous patterns with regular symmetry often display sparsity and low-rank properties. These properties suggest that the information contained within these patterns can be efficiently represented in a relatively low-dimensional space. In recent years, there has been a growing tendency to adopt low-rank features to characterize textures, and methods based on low-rank texture regularization have been proposed. The effectiveness of low-rank models in characterizing texture lies in their ability to capture redundant or repetitive information in image data. By fully exploiting the redundancy and regularity in images, low-rank models can represent and reconstruct texture images with fewer essential elements, thereby achieving more accurate characterization and in-depth analysis of textures.

Schaeffer and Osher [38] believed that if the patches of texture are represented as vectors, then the collection of patch vectors is (highly) linearly dependent, thus the matrix formed by the patch vectors of the texture image exhibits low rank. They, therefore, represented the cartoon-texture decomposition problem as the following low patch-rank (LPR) model [38]:

$$\min_{\mathbf{u},\mathbf{v}} \quad \frac{1}{2} \left\| K(\mathbf{u}+\mathbf{v}) - \mathbf{b} \right\|_2^2 + \lambda \left\| L\mathbf{u} \right\|_1 + \mu \left\| P\mathbf{v} \right\|_* . \tag{6}$$

Here, P is the patch mapping realigning texture $\mathbf{v}$ into a low-rank matrix. Recently, Zhang *et al.* [54] proposed a customized low-rank prior model (CLRP):

$$\min_{\mathbf{u},\mathbf{v}} \quad \frac{1}{2} \left\| K(\mathbf{u}+\mathbf{v}) - \mathbf{b} \right\|_2^2 + \lambda \left\| L\mathbf{u} \right\|_1 + \mu \left\| \mathbf{v} \right\|_* , \tag{7}$$

which utilizes a TV norm and a global nuclear norm to characterize the cartoon and texture components. Compared with LPR (6), the CLRP (7) is more straightforward but without losing its ability to extract cartoon and texture components from degraded images.

When solving these low-rank regularization models, the solution of subproblems usually involves the following nuclear norm regularization problem [4]:

$$\underset{X\in\mathbb{R}^{m\times n}}{\operatorname{argmin}} \quad \frac{1}{2}\|Y-X\|_F^2+\mu\|X\|_* . \tag{8}$$

Suppose the singular value decomposition (SVD) of Y is given by $Y=U\varSigma V^T\in\mathbb{R}^{m\times n}$, here $\varSigma$ is a diagonal matrix with entries $\sigma_i(Y)$. Then the optimal solution of (8) is represented as the soft-thresholding operator

$$\mathcal{S}_\mu(Y)=U\mathrm{diag}(\max\{\sigma_i(Y)-\mu,0\})V^T. \tag{9}$$

Here σ_i denotes the i-th singular value; U and V are the left singular matrix and right singular matrix of Y, respectively.

2.3 Variable Splitting Method

We now introduce variable splitting methods for solving the following convex optimization problems with three separable structures:

$$\begin{aligned} \min\ & \theta_1(\mathbf{x}_1)+\theta_2(\mathbf{x}_2)+\theta_3(\mathbf{x}_3)\\ s.t.\ & A_1\mathbf{x}_1+A_2\mathbf{x}_2+A_3\mathbf{x}_3=\mathbf{c}\\ & \mathbf{x}_1\in\mathcal{X}_1,\mathbf{x}_2\in\mathcal{X}_2,\mathbf{x}_3\in\mathcal{X}_3, \end{aligned} \tag{10}$$

where $\theta_i:\mathbb{R}^{n_i}\to(-\infty,+\infty], i=1,2,3$ are closed proper convex functions; $A_i\in\mathbb{R}^{l\times n_i}$ are given full column rank matrices; $\mathcal{X}_i\in\mathbb{R}^{m_i}$ are nonempty closed convex sets; and $\mathbf{c}\in\mathbb{R}^l$ is a vector. The augmented Lagrangian function of optimization problem (10) is defined as

$$\mathcal{L}_\beta(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{y})=\sum_{i=1}^3\theta_i(\mathbf{x}_i)-\left\langle\mathbf{y},\sum_{i=1}^3A_i\mathbf{x}_i-\mathbf{c}\right\rangle+\frac{\beta}{2}\left\|\sum_{i=1}^3A_i\mathbf{x}_i-\mathbf{c}\right\|_2^2, \tag{11}$$

where $\mathbf{y}\in\mathbb{R}^l$ is the Lagrange multiplier and $\beta>0$ is the penalty parameter. Suppose that $(\mathbf{x}_1^*,\mathbf{x}_2^*,\mathbf{x}_3^*,\mathbf{y}^*)$ is the saddle point of (11). Then, the first-order optimality condition of (10) can be easily characterized by the following variational inequality problem (VIP): finding a vector $\mathbf{z}^*\in\mathcal{U}$ such that

$$\theta(\boldsymbol{x}')-\theta(\boldsymbol{x}^*)+(\mathbf{z}'-\mathbf{z}^*)^TF(\mathbf{z}^*)\geq0,\quad\forall\mathbf{z}'\in\mathcal{U}:=\mathcal{X}_1\times\mathcal{X}_2\times\mathcal{X}_3\times\mathbb{R}^l, \tag{12}$$

where

$$\mathbf{z}:=\begin{pmatrix}\boldsymbol{x}\\\mathbf{y}\end{pmatrix}=\begin{pmatrix}\mathbf{x}_1\\\mathbf{x}_2\\\mathbf{x}_3\\\mathbf{y}\end{pmatrix}\text{ and }F(\mathbf{z}):=\begin{pmatrix}-A_1^T\mathbf{y}\\-A_2^T\mathbf{y}\\-A_3^T\mathbf{y}\\\sum_{i=1}^3A_i\mathbf{x}_i-\mathbf{c}\end{pmatrix}.$$

The ADMM method is useful in dealing with separable convex optimization problems [5,18, 21]. However, it is important to note that convergence cannot be guaranteed when the ADMM algorithm with two blocks is directly applied to a problem like (10). This was demonstrated by Chen *et al.* in [9] with a counterexample. Consequently, many variants of ADMM-based methods have been studied extensively over the past decade. Some of these variants include the variant alternating splitting augmented Lagrangian method [40], the alternating direction-based contraction method [18], and the ADMM with Gaussian back substitution [20]. In this paper, we apply the recently proposed ADMM-based prediction-correction method in [18] to solve the model (4). The process of this prediction-correction iterative scheme is given by

$$\begin{cases} \mathbf{x}_1^{(k+1)} = \mathrm{argmin}_{\mathbf{x}_1 \in \mathcal{X}_1} \mathcal{L}_\beta\left(\mathbf{x}_1, \mathbf{x}_2^{(k)}, \mathbf{x}_3^{(k)}, \mathbf{y}^{(k)}\right); \\ \mathbf{x}_2^{(k+1)} = \mathrm{argmin}_{\mathbf{x}_2 \in \mathcal{X}_2} \mathcal{L}_\beta\left(\mathbf{x}_1^{(k+1)}, \mathbf{x}_2, \mathbf{x}_3^{(k)}, \mathbf{y}^{(k)}\right); \\ \widetilde{\mathbf{x}}_3^{(k+1)} = \mathrm{argmin}_{\mathbf{x}_3 \in \mathcal{X}_3} \mathcal{L}_\beta\left(\mathbf{x}_1^{(k+1)}, \mathbf{x}_2^{(k+1)}, \mathbf{x}_3, \mathbf{y}^{(k)}\right); \\ \widetilde{\mathbf{y}}^{(k+1)} = \mathbf{y}^{(k)} - \beta\left(\sum_{i=1}^3 A_i \mathbf{x}_i^{(k+1)} - \mathbf{c}\right); \\ \mathbf{x}_3^{(k+1)} = \mathbf{x}_3^{(k)} + \alpha(\widetilde{\mathbf{x}}_3^{(k+1)} - \mathbf{x}_3^{(k)}); \\ \mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \alpha(\widetilde{\mathbf{y}}^{(k+1)} - \mathbf{y}^{(k)}). \end{cases} \tag{13}$$

The convergence of the ADMM-based prediction-correction method has been proved in [18], and here we summarize it in the following theorem:

Theorem 1 *Suppose* $\alpha \in \left(0, \frac{3-\sqrt{5}}{2}\right)$ *and* $\beta > 0$. *The sequence* $\{\mathbf{z}^{(k)}\}$ *generated by* (13) *converges to a solution of VIP* (12).

3 Image Decomposition Model with Edge-preserving Weighted Least Squares and Low-rank Regularization

In this section, we consider solving the image decomposition model (4), which decomposes an image into two components: one capturing the structure and another capturing the texture. The structure component is regularized by a weighted quadratic penalty function, and the texture component is regularized by a low-rank penalty function. For readability, we re-present the image decomposition model (4) here

$$\min_{\mathbf{u},\mathbf{v}} \quad \frac{1}{2}\left\|K(\mathbf{u}+\mathbf{v})-\mathbf{b}\right\|_2^2 + \frac{\lambda}{2}\left\|WL\mathbf{u}\right\|_2^2 + \mu\left\|\mathbf{v}\right\|_*.$$

The weighted matrix W encodes the structural information of the cartoon image, assigning larger weights to flatter regions for effective smoothing and smaller weights to edges to preserve structural features. However, the task of accurately distinguishing between flat regions and edges, and thus selecting appropriate weights for W, is a non-trivial task. To overcome this problem, we employ a self-guided update strategy similar to (21) for computing the weights.

3.1 Image Decomposition with Edge-preservation Weight Matrix

Now we consider how to compute the sequence $(\mathbf{u}, \mathbf{v})$ given the weight matrix W. To simplify the expression, we denote $D = WL$ and introduce the auxiliary variable $\mathbf{f} = \mathbf{u} + \mathbf{v}$. We solve the following constrained minimization problem to obtain $(\mathbf{u}, \mathbf{v})$

$$\begin{aligned} &\min_{\mathbf{u},\mathbf{v},\mathbf{f}} \frac{1}{2}\|K\mathbf{f}-\mathbf{b}\|_2^2 + \frac{\lambda}{2}\|D\mathbf{u}\|_2^2 + \mu\|\mathbf{v}\|_*, \\ &\text{s.t. } \mathbf{f} = \mathbf{u} + \mathbf{v}. \end{aligned} \tag{14}$$

Obviously, the problem (14) is a special case of (10) with the following specifications:

1. $\mathbf{x}_1 := \mathbf{u}, \mathbf{x}_2 := \mathbf{v}, \mathbf{x}_3 := \mathbf{f}$, and all $\mathcal{X}_i$ are full Euclidean spaces $\mathbb{R}^{mn}$;
2. $\theta_1(\mathbf{x}_1) := \frac{\lambda}{2}\|D\mathbf{u}\|_2^2, \theta_2(\mathbf{x}_2) := \mu\|\mathbf{v}\|_*$ and $\theta_3(\mathbf{x}_3) := \frac{1}{2}\|K\mathbf{f}-\mathbf{b}\|_2^2$;
3. $A_1 = A_2 := I, A_3 := -I$ and $\mathbf{c} := \mathbf{0}$.

Applying the framework of the variable splitting method, as detailed in Section 2.3, we can solve the problem (14).

The augmented Lagrangian function for (14) is formulated as follows:

$$L_\beta(\mathbf{f},\mathbf{u},\mathbf{v},\mathbf{y}) = \frac{1}{2}\|K\mathbf{f}-\mathbf{b}\|_2^2 + \frac{\lambda}{2}\|D\mathbf{u}\|_2^2 + \mu\|\mathbf{v}\|_* - \mathbf{y}^T(\mathbf{u}+\mathbf{v}-\mathbf{f}) + \frac{\beta}{2}\|\mathbf{u}+\mathbf{v}-\mathbf{f}\|_2^2,$$

where $\beta > 0$ is the penalty parameter and $\mathbf{y}$ is the Lagrangian multiplier. We solve the problem through (13). As the image decomposition algorithm consists of inner and outer iteration, we adopt the warm start strategy to initialize the algorithm to solve the inner minimization problem. Then a sequence of iterates is computed as follows:

$$\begin{cases} \mathbf{u}^{(k+1)} = \operatorname{argmin}_{\mathbf{u}} \frac{\beta}{2}\left\|\mathbf{u} - \left(\mathbf{f}^{(k)} + \frac{1}{\beta}\mathbf{y} - \mathbf{v}^{(k)}\right)\right\|_2^2 + \lambda\|D\mathbf{u}\|_2^2, \\ \mathbf{v}^{(k+1)} = \operatorname{argmin}_{\mathbf{v}} \frac{\beta}{2}\left\|\mathbf{v} - \left(\mathbf{f}^{(k)} + \frac{1}{\beta}\mathbf{y} - \mathbf{u}^{(k+1)}\right)\right\|_2^2 + \mu\|\mathbf{v}\|_*, \\ \widetilde{\mathbf{f}}^{(k+1)} = \operatorname{argmin}_{\mathbf{f}} \frac{1}{2}\|K\mathbf{f}-\mathbf{b}\|_2^2 + \frac{\beta}{2}\left\|\mathbf{u}^{(k+1)} + \mathbf{v}^{(k+1)} - \mathbf{f} - \frac{1}{\beta}\mathbf{y}^{(k)}\right\|_2^2, \\ \mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \alpha\left(\widetilde{\mathbf{f}}^{(k+1)} - \mathbf{f}^{(k)}\right), \\ \widetilde{\mathbf{y}}^{(k+1)} = \mathbf{y}^{(k)} - \beta\left(\mathbf{u}^{(k+1)} + \mathbf{v}^{(k+1)} - \mathbf{f}^{(k+1)}\right), \\ \mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \alpha\left(\widetilde{\mathbf{y}}^{(k+1)} - \mathbf{y}^{(k)}\right). \end{cases} \tag{15}$$

For the $\mathbf{u}$-subproblem, it is a Tikhonov-regularized minimization problem, and the minimizer is given as

$$\mathbf{u}^{(k+1)} = \beta\left(\beta I + \lambda(D)^T D\right)^{-1}\mathbf{u}_{\mathbf{b}}^{(k)}, \tag{16}$$

where $\mathbf{u}_{\mathbf{b}}^{(k)} = \mathbf{f}^{(k)} + \frac{1}{\beta}\mathbf{y}^{(k)} - \mathbf{v}^{(k)}$. The above linear system can be computed using an iterative least squares method, such as CGLS [15] or LSQR [32]. As the matrix D is the product of a diagonal matrix W and a difference matrix L, we can construct a pre-conditioner to decrease the iterative number and improve the computational speed effectively.

For the $\mathbf{v}$-subproblem, the optimization problem can be reformulated into the following one:

$$\mathbf{v}^{(k+1)} = \mathcal{S}_{\mu/\beta}\left(\mathbf{f}^{(k)} + \frac{1}{\beta}\mathbf{y}^{(k)} - \mathbf{u}^{(k+1)}\right). \tag{17}$$

For the $\mathbf{f}$-subproblem, it is also a Tikhonov-regularized minimization problem, and the minimizer is given by

$$\mathbf{f}^{(k+1)} = \left(K^T K + \beta I\right)^{-1} \left(K^T \mathbf{b} - \mathbf{y}^{(k)} + \beta(\mathbf{u}^{(k+1)} + \mathbf{v}^{(k+1)})\right). \tag{18}$$

When the linear operator K is an identity matrix or a down-sampling matrix, then the matrix $(K^T K + \beta I)$ is diagonal, so the linear system (18) can be solved directly. If K is a blurring operator created from a known spatially invariant point spread function, (18) can be solved quickly and efficiently by FFT or DCT [30].

3.2 Weighting Matrix

Once the weight matrix W is given, the image decomposition problem (4) can be easily solved by (15). Unfortunately, obtaining accurate weight coefficients is challenging. To compute these weights effectively, we ideally need knowledge of the regions in the original clean image. However, in practice, the observed image is often corrupted by noise, making it difficult to distinguish between true edges and noise-induced artifacts.

One approach to tackle this issue involves iteratively constructing the weight vector $\mathbf{w}$. Initially, the standard Tikhonov regularization method is employed to obtain the initial restored image $\mathbf{u}^{(0)}$ and the corresponding weight vector $\mathbf{w}^{(0)}$. Subsequently, for each iteration $k = 1, 2, \cdots$, the previously computed weight vector $\mathbf{w}^{(k-1)}$ is utilized to restore the image $\mathbf{u}^{(k)}$ through solving the minimization problem

$$\min_{\mathbf{u}} \frac{1}{2} \|K\mathbf{u} - \mathbf{b}\|_2^2 + \frac{\lambda}{2} \left\| W^{(k-1)} L\mathbf{u} \right\|_2^2 .$$

Which is equal to solving the following linear system:

$$K^T(K\mathbf{u}^{(k)} - \mathbf{b}) + \lambda (W^{(k-1)} L)^T W^{(k-1)} L\mathbf{u}^{(k)} = 0. \tag{19}$$

After obtaining $\mathbf{u}^{(k)}$, the diagonal weight matrix $W^{(k)}$ is constructed by assigning the values of $\mathbf{w}^{(k)}$ to its diagonal elements using various strategies. This iterative process continues until a satisfactory solution is attained.

These strategies can be broadly categorized into several approaches for constructing the weight $\mathbf{w}^{(k)}$. The first approach is based on the L_p-TV norm, we have $\mathbf{w}^{(k)} = \frac{1}{\sqrt{|L\mathbf{u}^{(k)}|^{2-p} + \epsilon^2}}$, with $0 < p < 2$ and $\epsilon > 0$. This iterative scheme corresponds to solving the generalized TV regularization problem using techniques such as the lagged diffusivity fixed-point iteration method [42], the majorization-minimization approach [26], or iterative reweighted least squares (IRLS)[10, 47,46].

The second approach is inspired by anisotropic diffusion [3,6,36]. The partial differential equation (PDE) can be employed to solve the image restoration problem as follows:

$$\frac{\partial \mathbf{u}}{\partial t} = -K^T(K\mathbf{u} - \mathbf{b}) + \lambda \operatorname{div}(g(|\nabla \mathbf{u}|)\nabla \mathbf{u}).$$

Here $g(|\nabla \mathbf{u}|)$ is an edge-stopping function. Then we can construct the weight according to the function $g(\|\nabla \mathbf{u}\|)$. The effectiveness of anisotropic diffusion hinges heavily on the choice of the edge-stopping function. Perona and Malik have proposed two famous edge-stopping functions defined as follows:

$$g_1(x) = e^{-\frac{x^2}{\kappa^2}}, \quad g_2(x) = \frac{1}{1+\frac{x^2}{\kappa^2}}, \tag{20}$$

where κ is a threshold parameter [36]. Additionally, there exist several other edge-stopping functions; interested readers are encouraged to explore [17,24] for further insights. The edge-stopping function should generally satisfy the following assumption [25].

Assumption 2 $g(x) \geqslant 0$ *is a smooth non-increasing function and meanwhile satisfies*

$$g(0) = 1, \lim_{x\to\infty} g(x) = 0.$$

The third approach employs an implicit weight, which differs significantly from the previous two methods. In this approach, the weight coefficients are not derived from the approximation of a specific norm nor generated by a pre-defined function. Instead, they are generated dynamically by designing an update rule. Gazzola *et al.* [12] proposed an inner-outer iterative algorithm for restoring and reconstructing images with enhanced edges, emphasizing enhanced edges through the IRLS approach. In the k-th iteration, the minimization problem is given by

$$\min_{\mathbf{u}} \frac{1}{2}\|K\mathbf{u}-\mathbf{b}\|_2^2 + \frac{\lambda}{2}\left\|W^{(k)}L\mathbf{u}\right\|_2^2, \quad k = 1, 2, \cdots. \tag{21}$$

The sequence's weight matrix $W^{(k)}$ is updated at each outer iteration. This is achieved by premultiplying the weight diag($\mathbf{w}^{(k)}$) with a weight matrix that encodes all the edge information revealed by previous outer iterations, *i.e.*,

$$W^{(k)} = \operatorname{diag}\left(\mathbf{w}^{(k)}\right) W^{(k-1)}. \tag{22}$$

Here $\mathbf{w}^{(k)}$ is assigned a weight through the equation

$$\mathbf{w}^{(k)} = \mathbf{1} - {\mathbf{t}^{(k)}}^q. \tag{23}$$

Here $q > 0$, the vector $\mathbf{t}^{(k)}$ represents the weighted gradient of the $(k-1)$-th image $\mathbf{u}^{(k-1)}$ normalized by the following equation:

$$\mathbf{t}^{(k)} = \frac{\left|W^{(k-1)}L\mathbf{u}^{(k-1)}\right|}{\left\|W^{(k-1)}L\mathbf{u}^{(k-1)}\right\|_\infty}. \tag{24}$$

The process of determining the weights $\mathbf{w}^{(k)}$ as in (23) is close to the concept of diffusion coefficients in anisotropic diffusion. By incorporating the function $g(x) = 1 - x^q$, a decreasing function defined within the interval [0,1], g can be regarded as the edge-stopping function within anisotropic diffusion. The weight matrix $W^{(k)}$ not only extracts image edges from the $(k-1)$th iteration but also overlays information from $W^{(k-1)}$. In other words, the weight information accumulates with each iteration.

Specifically, given an initial weight matrix $W^{(0)} = I$, we compute the following sequence:

$$(\mathbf{u}^{(k)}, \mathbf{v}^{(k)}) = \operatorname*{argmin}_{\mathbf{u},\mathbf{v}} \quad \frac{1}{2}\|K(\mathbf{u}+\mathbf{v})-\mathbf{b}\|_2^2 + \frac{\lambda}{2}\left\|W^{(k)}L\mathbf{u}\right\|_2^2 + \mu\|\mathbf{v}\|_*, \tag{25}$$

Algorithm 1: The adaptive diagonal weight method for cartoon-texture image decomposition.

```
Input: Observed image $\mathbf{b}$, parameters $\lambda, \mu$ and $\beta$
Output: $\mathbf{u}, \mathbf{v}$
Initialize $W^{(1)} = I$;
for $k = 1, 2, \ldots$, until the stopping criterion is satisfied do
    Solving the problem (25) for given $W^{(k)}$ and return the optimial minimizers $\mathbf{u}^{(k)}$ and $\mathbf{v}^{(k)}$;
    $\mathbf{t}^{(k)} = \frac{\left|W^{(k)} L\mathbf{u}^{(k)}\right|}{\left\|W^{(k)} L\mathbf{u}^{(k)}\right\|_\infty}$;
    $\mathbf{w}^{(k)} = g(\mathbf{t}^{(k)})$ ;
    $W^{(k+1)} = \operatorname{diag}\left(\mathbf{w}^{(k)}\right) W^{(k)}$ ;
end
return $\mathbf{u} = \mathbf{u}^{(k)}, \mathbf{v} = \mathbf{v}^{(k)}$.
```

for $k = 0, 1, 2, \cdots$. Once we have $\mathbf{u}^{(k)}$, we update the weight matrix $W^{(k+1)}$ by

$$W^{(k+1)} = \operatorname{diag}\left(\mathbf{w}^{(k)}\right) W^{(k)}. \tag{26}$$

Here the weight $\mathbf{w}^{(k)}$ is obtained by $\mathbf{w}^{(k)} = g(\mathbf{t}^{(k)})$ with

$$\mathbf{t}^{(k)} = \left|W^{(k)} L\mathbf{u}^{(k)}\right| / \left\|W^{(k)} L\mathbf{u}^{(k)}\right\|_\infty.$$

In contrast to the function used in [12], we introduce a novel function $g(t)$ for computing the weights. This function is defined as $g(x) = \cos\left(\frac{\pi x}{2}\right)$. It is easy to check that the function $g(x)$ satisfies Assumption 2.

3.3 Inner-Outer Algorithm

In fact, the regularization term of cartoon images can also be regarded as an iteratively reweighted Tikhonov regularization, which is easy to solve numerically. We use an inner-outer iterative approach to solve the sequence of optimization problems (25) for cartoon-texture image decomposition. We first adaptively update the edge-aware weight matrix using equations (26) in the outer iteration and then solve the problem with fixed weights by (15) in the inner iteration. The whole inner-outer iteration for cartoon-texture image decomposition is summarized in Algorithm 1.

4 Convergence Analysis

In this section, we consider the convergence of the proposed algorithm, which entails two layers of iteration. The inner layer iteration derives decomposed cartoon and texture images given a specific weight matrix, while the outer layer iteration updates this weight matrix based on the cartoon images. Consequently, our convergence analysis includes both the convergence of the inner layer iteration and the convergence of the weight matrix.

Theorem 3 *If* Null(K)$\cap$ Null(WL) $= \emptyset$*, then the minimization problem* (4) *has a unique solution.*

Proof Given matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ with rank r, by the definition of the Frobenius norm and nuclear norm, we have $\|\mathbf{A}\|_*^2 = \left(\sum_{i=1}^r \sigma_i\right)^2 \geq \sum_{i=1}^r \sigma_i^2 = \|\mathbf{A}\|_F^2$. Taking the square root of both sides, we conclude $\|\mathbf{A}\|_* \geq \|\mathbf{A}\|_F$. Fixing the weight matrix W, therefore, the objective function satisfies

$$\begin{aligned}\mathcal{J}(\mathbf{u},\mathbf{v}) &= \frac{1}{2}\left\|K(\mathbf{u}+\mathbf{v})-\mathbf{b}\right\|_2^2 + \frac{\lambda}{2}\left\|WL\mathbf{u}\right\|_2^2 + \mu\left\|\mathbf{v}\right\|_* \\ &\geq \frac{1}{2}\left\|K(\mathbf{u}+\mathbf{v})-\mathbf{b}\right\|_2^2 + \frac{\lambda}{2}\left\|WL\mathbf{u}\right\|_2^2 + \mu\|\mathbf{v}\|_2 \\ &= \frac{1}{2}\left\|\begin{pmatrix} K & K \\ \sqrt{\lambda}WL & 0\end{pmatrix}\begin{pmatrix}\mathbf{u}\\ \mathbf{v}\end{pmatrix} - \begin{pmatrix}\mathbf{b}\\ 0\end{pmatrix}\right\|_2^2 + \mu\left\|\begin{pmatrix}0 & I\end{pmatrix}\begin{pmatrix}\mathbf{u}\\ \mathbf{v}\end{pmatrix}\right\|_2 .\end{aligned} \tag{27}$$

We note that the matrix

$$\begin{pmatrix} K & K \\ \sqrt{\lambda}WL & 0 \\ 0 & I\end{pmatrix} = \begin{pmatrix} I & 0 & K \\ 0 & I & 0 \\ 0 & 0 & I\end{pmatrix}\begin{pmatrix} K & 0 \\ \sqrt{\lambda}WL & 0 \\ 0 & I\end{pmatrix}.$$

The above matrix is full rank as $Null(K) \cap Null(WL) = \emptyset$. Therefore, when $\left\|\begin{pmatrix}\mathbf{u}\\ \mathbf{v}\end{pmatrix}\right\|_2$ tends to infinity, $\mathcal{J}(\mathbf{u},\mathbf{v})$ also tends to infinity (*i.e.*, the objective function is coercive). A unique solution exists since the objective function is also convex (see Proposition 3.2.1 in [1]).

Therefore, our convergence analysis includes the convergence of the inner layer iteration and the convergence of the weight matrix.

Next, we will discuss the convergence of the weight matrix. Due to the normalization process and the absolute value operation, we know that the entries of the weight are between 0 and 1, specifically $0 \leq \mathbf{w}_i^{(k)} \leq 1$. Starting from $W^{(1)} = I$, based on the update formula $W = \operatorname{diag}\left(\mathbf{w}^{(k)}\right) W^{(k-1)}$, it is easy to check that the diagonal edge-aware weight matrix has the following property.

Lemma 1 *As the number of iteration steps increases, the diagonal entries of the weight matrix defined in* (26) *are non-increasing, i.e., the i-th diagonal entry of two consecutive weight matrices satisfies* $[W^{(k)}]_{ii} \leq [W^{(k-1)}]_{ii}$.

Lemma 1 indicates that the sequence $W^{(k)}$ is element-wise non-increasing and bounded. Therefore, according to the definition of the weight matrix, we have the following lemma.

Lemma 2 *Suppose* $W^{(k)}$ *is generated by* (26)*, then there exist a diagonal matrix* $W^{(*)}$ *such that* $\lim\limits_{k\to\infty} W^{(k)} = W^{(*)}$.

Lemma 2 shows that the sequence $W^{(k)}$ converges as iterations proceed, which is important to prove the convergence of the proposed algorithm. Let us define

$$\mathcal{J}(\mathbf{u},\mathbf{v};W^{(k)}) := \frac{1}{2}\left\|K(\mathbf{u}+\mathbf{v})-b\right\|_2^2 + \frac{\lambda}{2}\left\|W^{(k)}L\mathbf{u}\right\|_2^2 + \mu\left\|\mathbf{v}\right\|_* . \tag{28}$$

We now analyze the convergence of Algorithm 1. First, we have the following theorem.

Theorem 4 *Let $\{\mathbf{u}^{(k)}, \mathbf{v}^{(k)}; W^{(k)}\}$ be the sequence generated by Algorithm 1. Then the objective function value of $\mathcal{J}$ is non-increasing, that is*

$$\mathcal{J}(\mathbf{u}^{(k+1)}, \mathbf{v}^{(k+1)}; W^{(k+1)}) \leq \mathcal{J}(\mathbf{u}^{(k)}, \mathbf{v}^{(k)}; W^{(k)}).$$

Proof Given $W^{(k+1)}$, $(\mathbf{u}^{(k+1)}, \mathbf{v}^{(k+1)})$ is updated by

$$(\mathbf{u}^{(k+1)}, \mathbf{v}^{(k+1)}) = \underset{\mathbf{u},\mathbf{v}}{\operatorname{argmin}} \mathcal{J}(\mathbf{u}, \mathbf{v}, W^{(k+1)}).$$

Therefore, we can easily get

$$\mathcal{J}(\mathbf{u}^{(k+1)}, \mathbf{v}^{(k+1)}; W^{(k+1)}) \leq \mathcal{J}(\mathbf{u}^{(k)}, \mathbf{v}^{(k)}; W^{(k+1)}). \tag{29}$$

Notice the diagonal weight matrix is generated by $W^{(k+1)} = \operatorname{diag}\left(\mathbf{w}^{(k)}\right) W^{(k)}$, see (26). According to the definition of $\mathbf{w}^{(k)}$, we have $0 \leq \mathbf{w}^{(k)} \leq 1$. Hence for any vector $\mathbf{x}$, we have $\left\|\mathbf{w}^{(k)}\mathbf{x}\right\|_2^2 \leq \|\mathbf{x}\|_2^2$, which implies

$$\left\|W^{(k+1)} L\mathbf{u}^{(k)}\right\|_2^2 = \left\|\operatorname{diag}\left(\mathbf{w}^{(k)}\right) W^{(k)} L\mathbf{u}^{(k)}\right\|_2^2 \leq \left\|W^{(k)} L\mathbf{u}^{(k)}\right\|_2^2. \tag{30}$$

By (28), we obtain

$$\mathcal{J}(\mathbf{u}^{(k)}, \mathbf{v}^{(k)}; W^{(k+1)}) - \mathcal{J}(\mathbf{u}^{(k)}, \mathbf{v}^{(k)}; W^{(k)}) = \frac{\lambda}{2}\left(\left\|W^{(k+1)} L\mathbf{u}^{(k)}\right\|_2^2 - \left\|W^{(k)} L\mathbf{u}^{(k)}\right\|_2^2\right). \tag{31}$$

Summing the inequalities (29) and (30), we obtain the

$$\mathcal{J}(\mathbf{u}^{(k+1)}, \mathbf{v}^{(k+1)}; W^{(k+1)}) \leq \mathcal{J}(\mathbf{u}^{(k)}, \mathbf{v}^{(k)}; W^{(k)}).$$

Hence, according to Theorem 4, we easily get the properties of the sequence $(\mathbf{u}^{(k)}, \mathbf{v}^{(k)})$.

Theorem 5 *If* $\operatorname{Null}(K) \cap \operatorname{Null}(WL) = \emptyset$*, then the sequences $\mathbf{u}^{(k)}$ and $\mathbf{v}^{(k)}$ obtained by Algorithm 1 are bounded. Therefore, there exist two convergent subsequences $\mathbf{u}^{(k_j)}$ and $\mathbf{v}^{(k_j)}$ such that* $\lim_{j\to\infty} \mathbf{u}^{(k_j)} = \mathbf{u}^{(*)}$, $\lim_{j\to\infty} \mathbf{v}^{(k_j)} = \mathbf{v}^{(*)}$.

Due to the coerciveness of the convex function $\mathcal{J}(\mathbf{u}, \mathbf{v})$, we can show the existence and uniqueness of the solution of the problem

$$\min_{u,v} \quad \mathcal{J}(\mathbf{u}, \mathbf{v}; W^{(*)}) \tag{32}$$

in the following theorems.

Theorem 6 *The limit point $(\mathbf{u}^{(*)}, \mathbf{v}^{(*)})$ is the unique minimum of minimization* (32).

Proof We know that $(\mathbf{u}^{(k)}, \mathbf{v}^{(k)})$ minimizes $\mathcal{J}(\mathbf{u}, \mathbf{v}, W^{(k)})$, *i.e.*,

$$(\mathbf{u}^{(k)}, \mathbf{v}^{(k)}) = \underset{\mathbf{u},\mathbf{v}}{\operatorname{argmin}} \mathcal{J}(\mathbf{u}, \mathbf{v}, W^{(k)}).$$

Hence $\mathbf{u}^{(k)}$ and $\mathbf{v}^{(k)}$ satisfy the optimal condition

$$\begin{cases} 0 = K^T K(\mathbf{u}^{(k)} + \mathbf{v}^{(k)}) - K^T\mathbf{b} + \lambda L^T W^{(k)T} W^{(k)} L\mathbf{u}^{(k)}, \\ 0 \in K^T K(\mathbf{u}^{(k)} + \mathbf{v}^{(k)}) - K^T\mathbf{b} + \mu\partial \left\|\mathbf{v}^{(k)}\right\|_* . \end{cases} \tag{33}$$

Further, there exists a subsequence $(\mathbf{u}^{(k_j)}, \mathbf{v}^{(k_j)})$ which subjects to

$$\begin{cases} 0 = K^T K(\mathbf{u}^{(k_j)} + \mathbf{v}^{(k_j)}) - K^T\mathbf{b} + \lambda L^T W^{(k_j)T} W^{(k_j)} L\mathbf{u}^{(k_j)}, \\ 0 \in K^T K(\mathbf{u}^{(k_j)} + \mathbf{v}^{(k_j)}) - K^T\mathbf{b} + \mu\partial \left\|\mathbf{v}^{(k_j)}\right\|_* . \end{cases} \tag{34}$$

Let $j \to \infty$, then we have

$$\begin{cases} 0 = K^T K(\mathbf{u}^{(*)} + \mathbf{v}^{(*)}) - K^T\mathbf{b} + \lambda L^T W^{(*)T} W^{(*)} L\mathbf{u}^{(*)}, \\ 0 \in K^T K(\mathbf{u}^{(*)} + \mathbf{v}^{(*)}) - K^T\mathbf{b} + \mu\partial \left\|\mathbf{v}^{(*)}\right\|_* . \end{cases} \tag{35}$$

Therefore, the conclusion holds.

5 Numerical Experiments

In this section, we conduct a series of numerical experiments to show the efficiency of the proposed edge-preserving WLS and low-rank regularization model and algorithm. Ideally, the outer iteration in Algorithm 1 stops when the cartoon component has no more edge information to be enhanced. We use the same stopping criterion as in [12], *i.e.*, $\frac{\|L\mathbf{u}^{(k)}\|_2}{\|L\mathbf{u}^{(k-1)}\|_2} < c$. Here $c > 1$ controls the smoothness of cartoon components to adjust this value in different images and tasks. If you want to make the cartoon image smoother, choose a larger c; correspondingly, a smaller c is a good choice if one wants to retain more details. Here we set $c = 1.005$ for synthetic images and set $c = 1.05$ for natural images. The original images used in the following experiments are shown in Fig. 2 and Fig. 3. Fig. 2 consists of synthetic images, and Fig. 3 consists of the natural images.

5.1 The Effectiveness of W on the Real Cartoon Image

In this subsection, we experiment to validate the effectiveness of our method in decomposing images and identifying edge locations in cartoon images using a weighted matrix that encodes the structural information related to the decomposition. We present the decomposition results and edge locations determined by the ideal weights derived from the true cartoon images. The updating rule of the true weighted matrix is given by

$$\widehat{W}^{(k)} = \operatorname{diag}\left(\widehat{\mathbf{w}}^{(k)}\right)\widehat{W}^{(k-1)}, \widehat{\mathbf{w}}^{(k)} = g(\widehat{\mathbf{t}}^{(k)}), \widehat{\mathbf{t}}^{(k)} = \frac{\left|\widehat{W}^{(k-1)}L\mathbf{u}_{true}\right|}{\left\|\widehat{W}^{(k-1)}L\mathbf{u}_{true}\right\|_\infty}. \tag{36}$$

Here $\mathbf{u}_{true}$ denotes the true cartoon image. The updating rule for the estimated weighted matrix of the proposed method is given as follows

$$W^{(k)} = \operatorname{diag}\left(\mathbf{w}^{(k)}\right)W^{(k-1)}, \mathbf{w}^{(k)} = g(\mathbf{t}^{(k)}), \mathbf{t}^{(k)} = \frac{\left|W^{(k-1)}L\mathbf{u}^{(k-1)}\right|}{\left\|W^{(k-1)}L\mathbf{u}^{(k-1)}\right\|_\infty}. \tag{37}$$

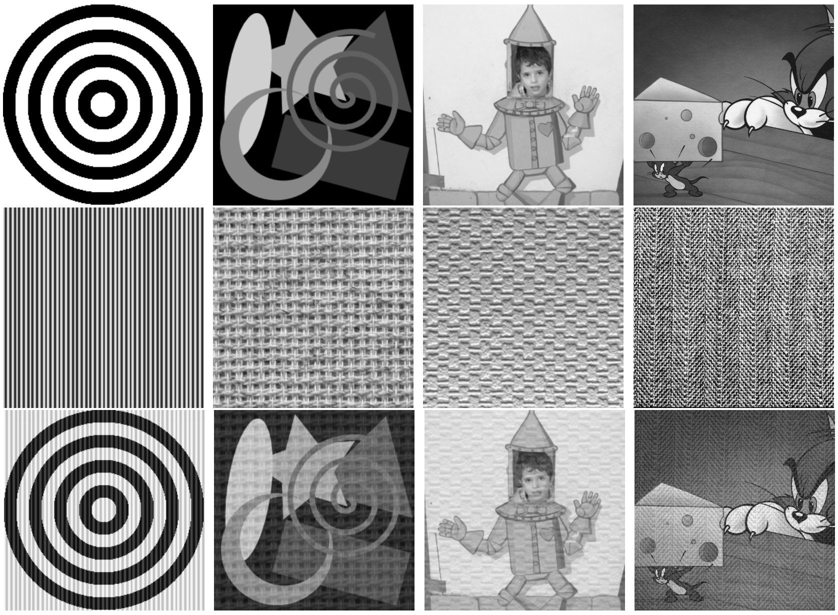


Fig. 2 The test original synthetic cartoon images and texture images. From the first column to the last column: “Circle”, “Geometry”, “Boy”, “TomJerry”. From the first row to the last row: cartoon images, texture images, and synthetic images (a cartoon image and a texture image of the ratio 7 : 3).

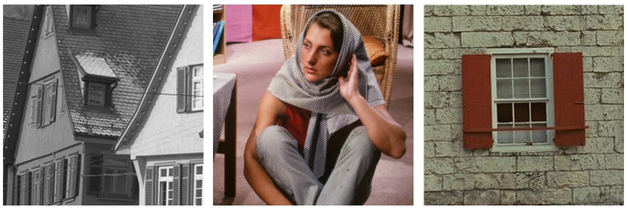


Fig. 3 Natural images for image decomposition. From left to right: “House”, “Barbara”, “Window”, respectively.

See (26). We remark that the true weighted matrix is obtained in the computation process by substituting $\mathbf{u}_{true}$ for $\mathbf{u}^{(k-1)}$.

In our study, we explore the decomposition results of synthetic images using two different weighted matrices: the true weighted matrix (36) and the estimated weighted matrix (37). These results are visually presented in Fig. 4. Both weighted matrices successfully separate the cartoon image from the texture image, validating the effectiveness of our weight estimation strategy. To understand the impact on edge locations, we compare the true cartoon image with the iteratively updated cartoon image. The results are similar for the four images, so we focus on the “Geometry” image for presentation purposes. We analyze the first and final estimated weight matrices. Fig. 5 shows both the horizontal and vertical weight matrices. Surprisingly, even when utilizing the true cartoon image, the edge information in the first iteration remains unclear. However, as the iterations progress, the cartoon image’s edge positions gradually become more distinct. A similar trend is observed when using the iteratively updated cartoon image for weight computation. The weight matrix definition accumulates edge information from each iteration,

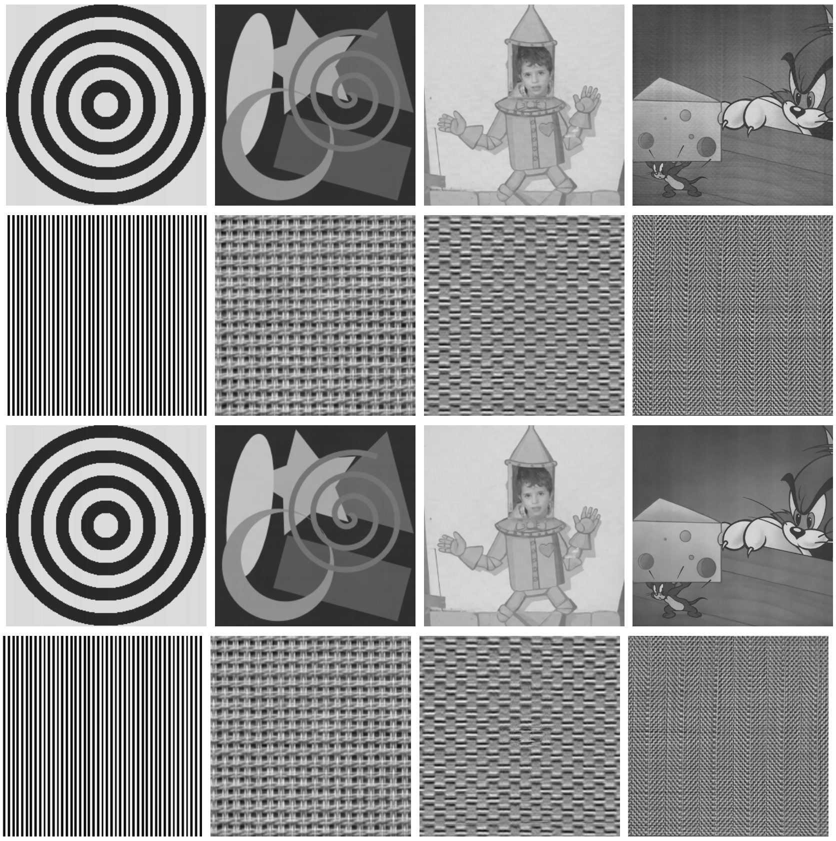


Fig. 4 Cartoon and texture images decomposed by different weights using the algorithm 1. The first two raws are the real weights W calculated by (36) using the true cartoon image $\mathbf{u}_{true}$, and the last two raws are the estimated weights $\widehat{W}$ calculated by (37) using the estimated cartoon image $\mathbf{u}$.

subsequently incorporating it into the regularization term of the next iteration. This iterative process significantly enhances the image's edges.

5.2 Comparison with the State-of-the-art Methods

To demonstrate the effectiveness of the proposed model, we compare the proposed method with the state-of-art cartoon-texture decomposition methods that utilize low-rank prior, including LPR [18], Log-det [11], CLRP [54], and GNCLR [48]. We obtained the source codes for these methods from the authors' websites and fine-tuned their parameters to achieve the best visual performance. To quantitatively evaluate the quality of the decomposed images, we employ the signal-to-noise ratio (SNR)[1].

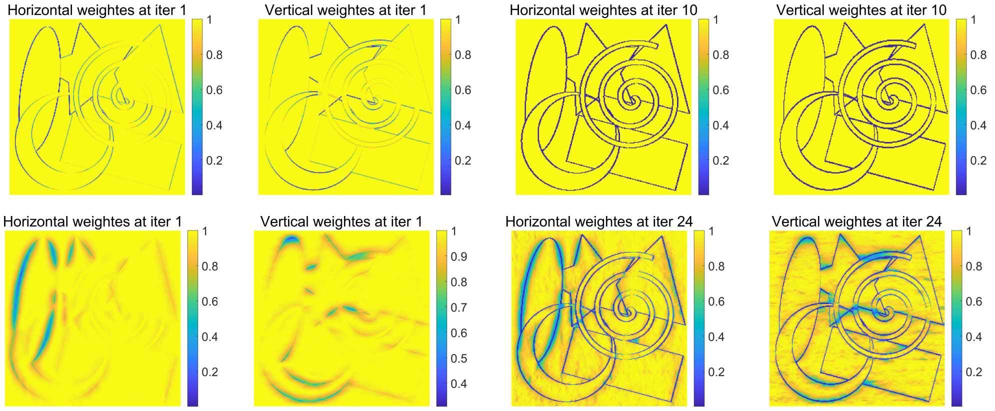


Fig. 5 The weights of vertical and horizontal directions for "Geometry". The first and second rows are the true weights (*i.e.,* (36)) and the weights estimated by the proposed method (*i.e.,* (37)) at the first and last iteration, respectively.

Table 1 SNR(dB) comparison for image decomposition without degradations.

		7:3			**5:5**		
Images	**Method**	**u**	**v**	**Time(s)**	**u**	**v**	**Time(s)**
Boy	**LPR**	24.14	16.53	1.20	20.95	19.50	1.18
	Log-det	26.08	17.74	2.57	21.51	20.17	2.55
	CLRP	30.35	21.27	0.29	25.04	23.57	0.28
	GNCLR	31.05	21.27	0.91	25.32	23.97	1.26
	Proposed	**32.61**	**23.91**	4.17	**26.21**	**24.97**	4.67
TomJerry	**LPR**	19.41	13.74	5.22	14.12	16.36	5.25
	Log-det	20.65	14.98	10.73	14.33	16.48	10.74
	CLRP	23.86	18.13	1.35	19.88	21.52	1.37
	GNCLR	24.17	18.08	9.50	18.20	20.02	13.30
	Proposed	**25.85**	**20.20**	45.30	**20.24**	**22.08**	35.00

5.2.1 Image Decomposition without Degradation

In this experiment, we consider to decompose an image without degradation, i.e., K is an identity matrix. We generate two synthetic images of different scales to evaluate the numerical results of color image decomposition for different methods. The synthetic images are superposed by a cartoon image and a texture image of the ratio $7:3$ and $5:5$. We set parameters $\beta = 10^{-3}, \mu = 10^{-4}$, and $\lambda = 2 \times 10^{-5}$.

We perform a quantitative evaluation of the quality of the decomposed images by comparing the SNRs for different methods. The results are presented in Table 1. For the "Boy" image, the CLRP method achieves an SNR of 30.35dB for the cartoon component and 21.27dB for the texture, with GNCLR slightly higher at 31.05dB and 21.91dB. The proposed method surpasses

[1] SNR is defined as $\text{SNR} = 20\log_{10}\frac{\|f\|_2}{\|\hat{f}-f\|_2}$, where f and $\hat{f}$ are the true images and the reconstructed image, respectively.

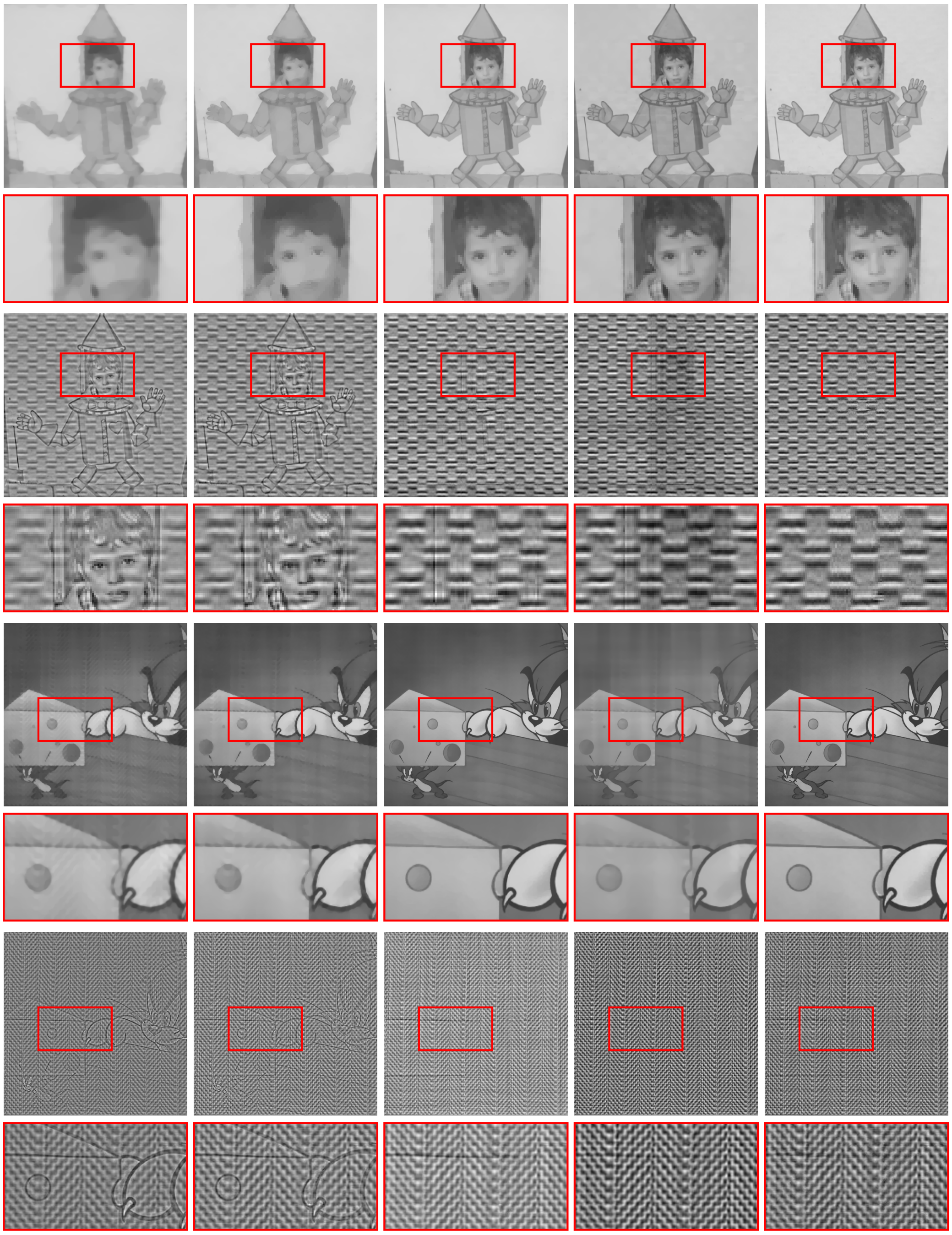


Fig. 6 Results of image decomposition without degradation. From left to right: the results of LPR, Logdet, CLRP, GNCLR, and Proposed method, respectively.

both, reaching 32.61dB for the cartoon and 23.91dB for the texture, demonstrating its superior quality. Although it requires 4.17s, longer than CLRP's, the SNR improvement is significant. Similar trends are observed in the 5:5 case for "Boy": the proposed method achieves the highest SNRs (26.21dB for cartoon and 24.97dB for texture) with a running time of 4.67s, justifying the trade-off for quality. For the "TomJerry" image, CLRP and GNCLR achieve competitive SNRs, but the proposed method leads with 25.85dB for cartoon and 20.20dB for texture. Though the computation time is longer at 45.30s, it consistently delivers the best results. In the 5:5 case, the proposed method maintains the highest SNRs.

For the 7:3 case, as shown in Fig. 6, we present the results of the cartoon and texture components decomposed using different methods. From the figure, we make the following observations. Both the LPR and Log-det methods tend to oversmooth the cartoon image during the separation process. Consequently, some structural edges that inherently belong to the cartoon image are erroneously assigned to the texture image. In contrast, the CLRP, GNCLR and our proposed method demonstrate superior decomposition performance. These methods effectively mitigate the over-smoothing issue, more accurately separating cartoon and texture components. Notably, they outperform LPR and Log-det regarding edge preservation and overall quality. To further compare the decomposition effects of these methods, we selected representative local regions from the decomposed cartoon and texture images. These regions are then magnified for detailed analysis. We also observe that the edges of cartoon images obtained using the CLRP and GNCLR appear less sharp than those from our proposed approach. Additionally, the texture images produced by CLRP exhibit some structural edges that should ideally belong to the cartoon component. In summary, our method effectively maintains sharp cartoon image edges while successfully separating texture components, resulting in superior decomposition performance.

Table 2 SNR(dB) comparison for simultaneous cartoon and texture decomposition in image deblurring

		7:3				**5:5**			
Images	**Method**	**u**	**v**	**u+v**	**Time**	**u**	**v**	**u+v**	**Time**
Boy	**LPR**	23.39	16.65	32.48	1.42	20.97	19.74	32.48	1.45
	Log-det	25.24	17.15	36.49	1.62	20.76	19.63	36.45	1.58
	CLRP	29.50	21.06	41.21	0.27	24.67	23.40	42.21	0.29
	GNCLR	30.85	21.17	43.94	0.95	25.37	24.02	44.67	1.36
	Proposed	**31.47**	**23.48**	**45.80**	3.40	**25.53**	**24.61**	**46.40**	2.56
TomJerry	**LPR**	18.84	14.03	28.68	5.74	14.84	16.77	28.18	5.77
	Log-det	19.72	14.11	33.29	6.11	13.49	16.19	32.96	6.12
	CLRP	23.63	18.41	36.92	1.33	19.76	21.60	38.51	1.37
	GNCLR	24.09	18.09	36.81	9.75	18.29	20.10	38.95	14.00
	Proposed	**25.09**	**19.79**	**38.94**	31.94	**19.97**	**21.70**	**40.58**	21.76

5.2.2 Simultaneous Cartoon-Texture Decomposition and Image Deblurring

In this experiment, we consider simultaneously performing image deblur and cartoon-texture decomposition. An out-of-focus blur with a radius of 3 is used to generate the blurred observed

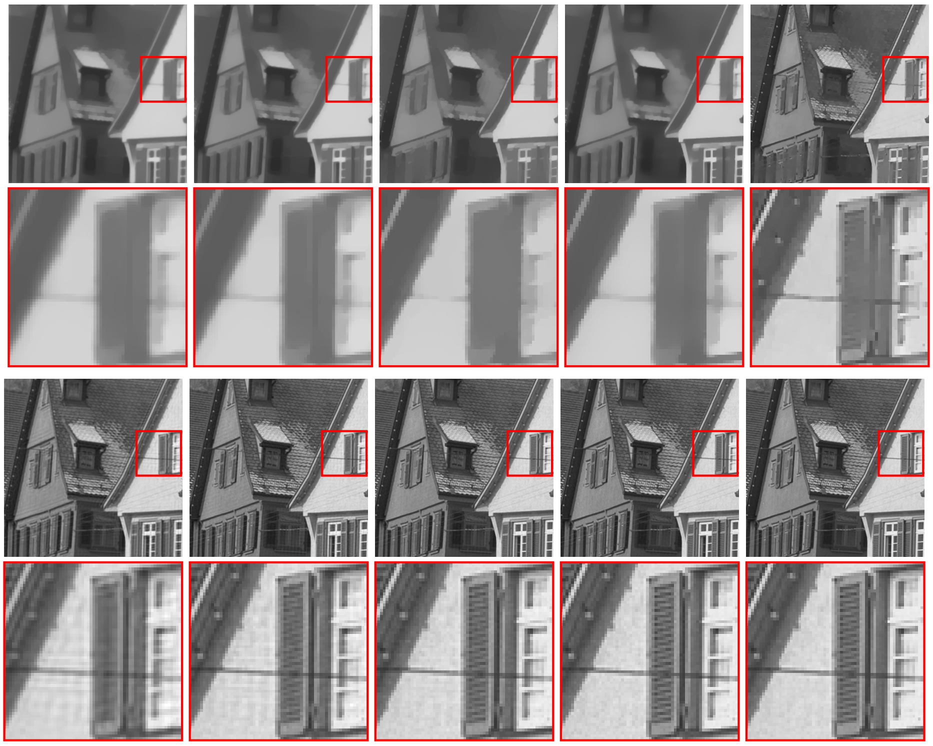


Fig. 7 Simultaneous cartoon and texture decomposition in image deblurring. From left to right: the results of LPR, Logdet, CLRP, GNCLR, and Proposed method, respectively. The first and third rows are the cartoon images and the reconstructed images. The second and fourth rows are the corresponding enlarged images of the red rectangle, respectively. The SNRs of the recovered images (*i.e.,* $\mathbf{u}+\mathbf{v}$) for these four methods are 22.10dB, 27.32dB, 30.08dB, 33.62, and 34.99dB, respectively.

images. The parameters are set as $\beta = 10^{-3}, \mu = 10^{-4}$, and $\lambda = 10^{-5}$ for the proposed method. The parameters of other methods are tested based on the values in the numerical section of these methods. The numerical results include both synthetic and natural images.

The comparison of various methods on synthetic images, where the ground truth cartoon and texture components are known, is summarized in Table 2. The table presents the SNR values for the decomposed cartoon $\mathbf{u}$ and texture $\mathbf{v}$ images, along with the SNR of the combined image $(\mathbf{u}+\mathbf{v})$. For the “Boy” image in the 7:3 case, the proposed method achieves the highest SNRs—31.47dB for the cartoon and 23.48dB for the texture, 45.80dB for the combined image. Although it requires 3.40s, it provides a clear quality improvement. In the 5:5 case, the proposed method maintains this advantage, with the highest total SNR of 46.40dB, again justifying its longer computation time. For the “TomJerry” image in the 7:3 case, the proposed method also leads with SNRs of 25.09dB (cartoon) and 19.79dB (texture), combined 38.94dB, despite a longer runtime of 31.94s. In the 5:5 case, it achieves the highest combined SNR of 40.58dB in 21.76s, underscoring its effectiveness in delivering superior decomposition results. Fig. 7 visually compares the cartoon and reconstructed images $(\mathbf{u}+\mathbf{v})$ obtained from the five methods. To further illustrate the decomposition quality, enlarged regions within the red rectangles are also provided. It is observed that the cartoon image obtained by our method exhibits sharp edges, while the edges in the cartoon images separated by the other three methods appear blurred. As

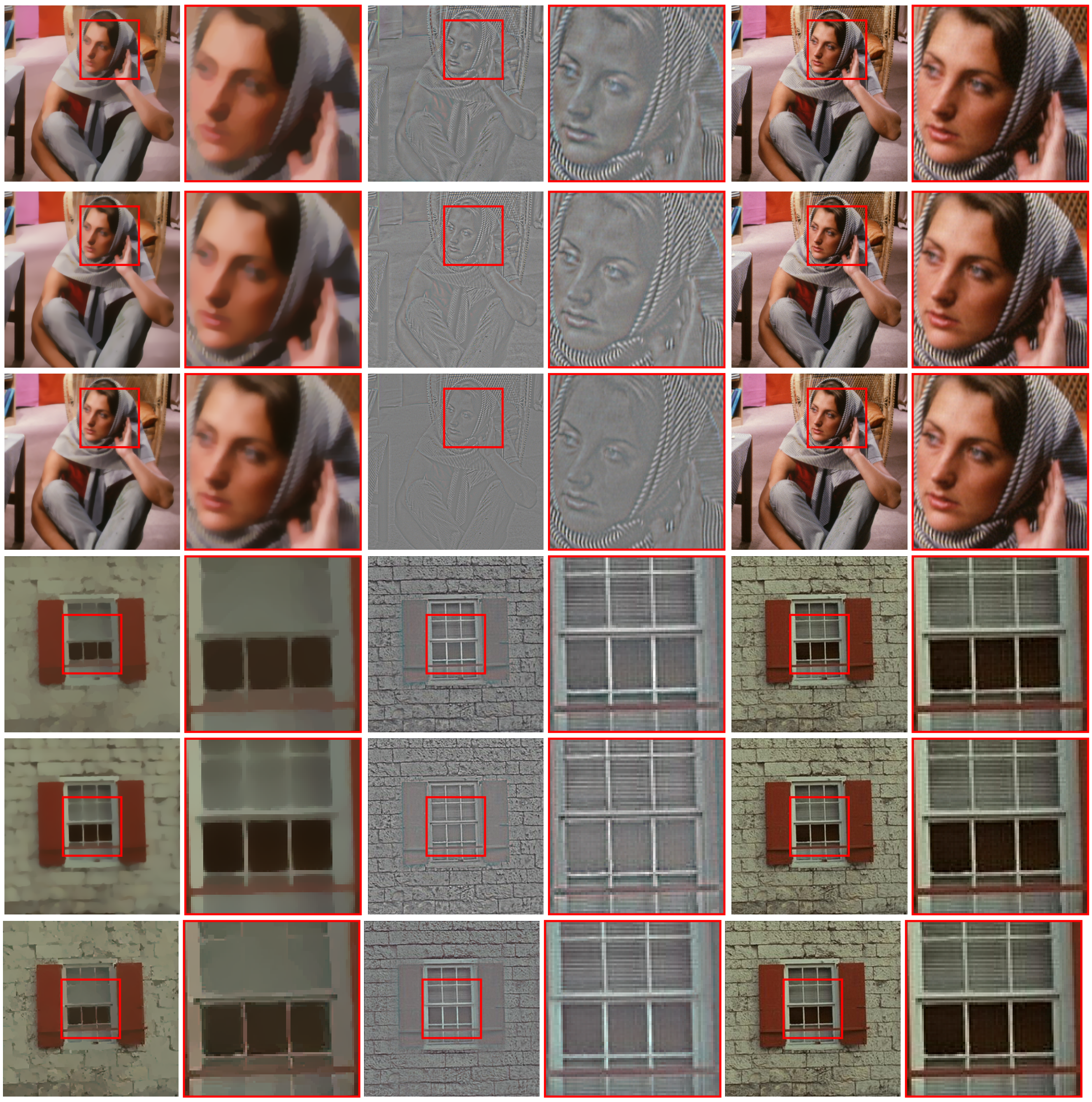
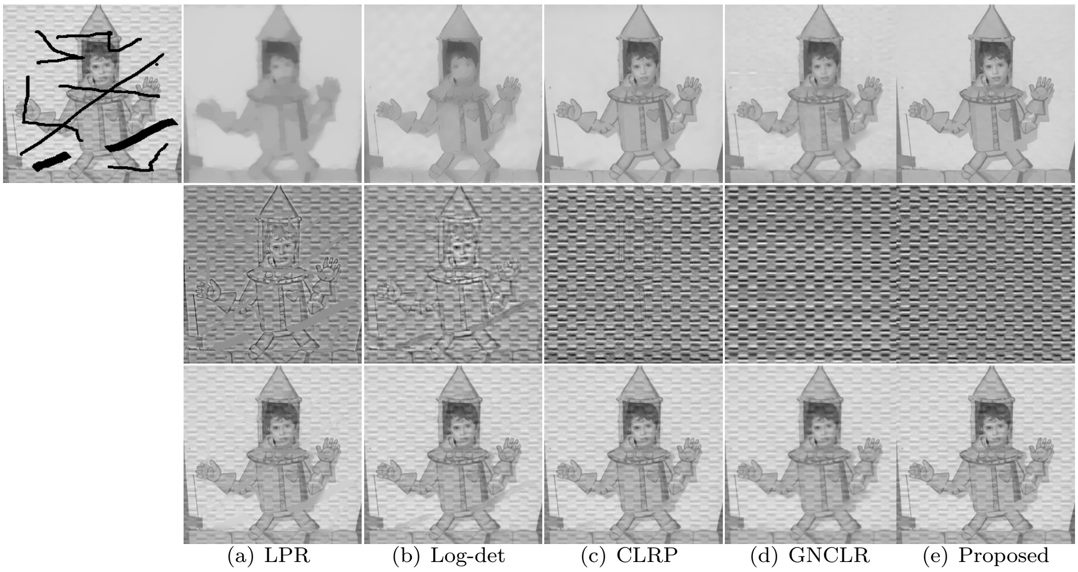


Fig. 8 Simultaneous cartoon and texture decomposition in image deblurring. The first and third rows display the results of CLRP, while the second and last rows show the results of the proposed method. For the recovered image "Barbara", the SNR is 32.35dB for CLRP, 36.13dB for GNCLR, and 35.36dB for our proposed method. For the recovered image "Window", the SNR is 28.11dB for CLRP, 32.61 for GNCLR, and 32.65dB for our proposed method.

is shown in Fig. 8, for the grayscale natural "House" image, the SNR values for the recovered images, using LPR, Log-det, CLRP, GNCLR, and the proposed method, are 22.10dB, 27.32dB, 30.08dB, 33.62 and 34.99dB, respectively. Our method achieves the highest SNR and the best visual effect.

For the recovered color image "Barbara", CLRP achieves an SNR of 32.35dB, and GNCLR improves on this with an SNR of 36.13dB. Our proposed method reaches a comparable SNR of 35.36dB. However, as shown in Fig. 8, our method provides a distinct advantage in visual



(a) LPR (b) Log-det (c) CLRP (d) GNCLR (e) Proposed

Fig. 9 Simultaneous cartoon-texture decomposition and image inpainting. From top to bottom: the cartoon, texture, and cartoon+texture. The SNRs for cartoon, texture, and cartoon + texture are as follows: LPR (23.09dB, 15.89dB, 31.54dB), Logdet (24.35dB, 16.86dB, 32.31dB), CLRP (26.72dB, 16.30dB, 34.52dB), GNCLR (27.61dB, 16.99dB, 32.75dB) and our method (29.14dB, 23.92dB, 34.30dB).

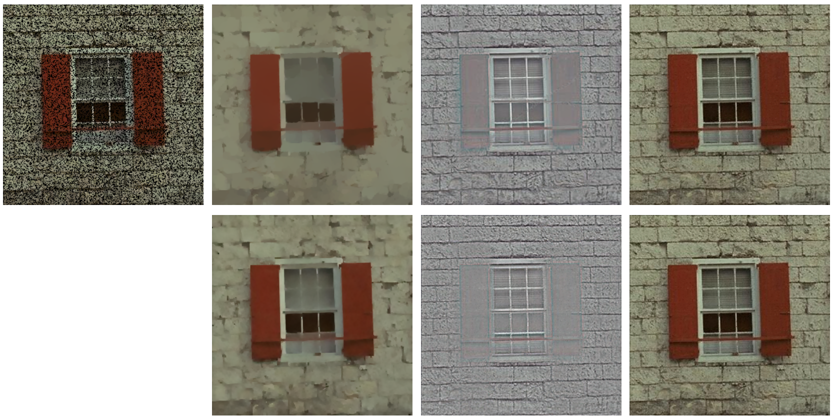


Fig. 10 Simultaneous cartoon-texture decomposition and image inpainting. The first and second rows show the results obtained by GNCLR, CLRP, and our method. From left to right: the cartoon, texture, and cartoon+texture. The SNRs of the recovered images for CLRP and Proposed are 22.49dB, 21.77dB and 22.09dB, respectively.

clarity: the facial features in the cartoon image "Barbar" are sharper and more defined with our approach, while the features from the CLRP and GNCLR methods appear blurred, resulting in reduced edge sharpness. This highlights the proposed method's effectiveness in preserving fine details. For the recovered image "Window", CLRP achieves an SNR of 28.11dB, and GNCLR increases this to 32.61dB. The proposed method achieves the highest SNR of 32.65dB, slightly surpassing GNCLR and demonstrating its effectiveness in producing high-quality decompositions for this image.

5.2.3 Simultaneous Cartoon-Texture Decomposition and Image Inpainting

In this experiment, we consider image inpainting problems through the application of cartoon and texture decomposition. The gray-textured “Boy” image is superimposed with a mask that covers 9.12% of its original area, resulting in information loss due to sketching. The decomposition results are shown in Fig. 9. It is found that compared with other methods, both cartoon and texture images separated by our method achieve the best visual results.

We also conduct an experiment on the color image “Window”, where 30% of the pixels are randomly missing, as depicted in Fig. 10. Consistent with previous experiments, the cartoon image decomposed by CLRP exhibits blurred edges, while our method yields a cartoon image with clear edges. Furthermore, from an SNR perspective, our reconstructed image has an SNR that is 0.32dB higher than that of CLRP, demonstrating the superior performance of our method.

6 Conclusion

In this paper, we developed a new image decomposition model for edge preservation. The model used Tikhonov-type edge-preserving regularization and nuclear norm to characterize the cartoon’s piecewise smoothness and the texture’s low rankness, respectively. In order to solve the problem, we have applied the inner-outer iterative scheme. We first adaptively update the edge-aware weight matrix in the outer loop and then solve the separable convex optimization problem by the ADMM-based prediction-correction method, in which the corresponding subproblems all have closed solutions. We proved the convergence of the algorithm. Numerical results showed that the cartoon images separated by our approach retained the complete edges. Meanwhile, the proposed method achieved the best performance compared with the state-of-the-art methods.

Acknowledgements This work was funded by the National Natural Science Foundation of China (Grant Nos. 12361089); the Scientific Research Fund Project of Yunnan Provincial Education Department (Grant No. 2024J0642); Yunnan Fundamental Research Projects (Grant No. 202401AU070104).

References

1. D. Bertsekas. *Convex optimization theory*, volume 1. Athena Scientific, 2009.
2. J. Biemond and R. L. Lagendijk. Regularized iterative image restoration in a weighted Hilbert space. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 11:1485–1488, 1986.
3. M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3):421–432, 1998.
4. J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.
5. X. Cai, D. Han, and X. Yuan. On the convergence of the direct extension of ADMM for three-block separable convex minimization models with one strongly convex function. *Computational Optimization and Applications*, 66(1):39–73, 2017.
6. F. Catté, P.-L. Lions, J.-M. Morel, and T. Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical analysis*, 29(1):182–193, 1992.
7. A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1):89–97, 2004.
8. A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock. An introduction to total variation for image analysis. *Theoretical foundations and numerical methods for sparse recovery*, 9(263-340):227, 2010.

9. C. Chen, B. He, Y. Ye, and X. Yuan. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, 155(1):57–79, 2016.
10. I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1):1–38, Jan. 2010.
11. Y.-R. Fan, T.-Z. Huang, T.-H. Ma, and X.-L. Zhao. Cartoon-texture image decomposition via non-convex low-rank texture regularization. *Journal of the Franklin Institute*, 354(7):3170–3187, 2017.
12. S. Gazzola, M. E. Kilmer, J. G. Nagy, O. Semerci, and E. L. Miller. An inner-outer iterative method for edge preservation in image restoration and reconstruction. *Inverse Problems*, 36(12):124004, 2020.
13. G. Gilboa, N. Sochen, and Y. Y. Zeevi. Variational denoising of partly textured images by spatially varying constraints. *IEEE Transactions on Image Processing*, 15(8):2281–2289, 2006.
14. G. H. Golub, P. C. Hansen, and D. P. O'Leary. Tikhonov regularization and total least squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1):185–194, 1999.
15. G. H. Golub and C. F. Van Loan. *Matrix Computations*. JHU press, 2013.
16. S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2862–2869, 2014.
17. B. Gupta, S. S. Lamba. An efficient anisotropic diffusion model for image denoising with edge preservation. *Computers & Mathematics with Applications*, 93:106–119, 2021.
18. D. Han, W. Kong, and W. Zhang. A partial splitting augmented Lagrangian method for low patch-rank image decomposition. *Journal of Mathematical Imaging and Vision*, 51(1):145–160, 2015.
19. Y. Han, C. Xu, G. Baciu, and M. Li. Lightness biased cartoon-and-texture decomposition for textile image segmentation. *Neurocomputing*, 168:575–587, 2015.
20. B. He, M. Tao, and X. Yuan. Alternating direction method with Gaussian back substitution for separable convex programming. *SIAM Journal on Optimization*, 22(2):313–340, 2012.
21. B. He and X. Yuan. A class of ADMM-based algorithms for three-block separable convex programming. *Computational Optimization and Applications*, 70(3):791–826, 2018.
22. G. Huang, A. Lanza, S. Morigi, L. Reichel, and F. Sgallari. Majorization–minimization generalized krylov subspace methods for ℓ_p–ℓ_q optimization applied to image restoration. *BIT Numerical Mathematics*, 57(2):351–378, 2017.
23. Y. Huang, M. K. Ng, and Y.-W. Wen. A fast total variation minimization method for image restoration. *Multiscale Modeling & Simulation*, 7(2):774–795, 2008.
24. V. Kamalaveni, R. A. Rajalakshmi, and K. Narayanankutty. Image denoising using variations of Perona-Malik model with different edge stopping functions. *Procedia Computer Science*, 58:673–682, 2015.
25. D. Karras and G. Mertzios. New pde-based methods for image enhancement using som and bayesian inference in various discretization schemes. *Measurement Science and Technology*, 20(10):104012, 2009.
26. A. Lanza, S. Morigi, L. Reichel, and F. Sgallari. A generalized Krylov subspace method for $\ell_p - \ell_q$ minimization. *SIAM Journal on Scientific Computing*, 37(5):S30–S50, 2015.
27. X. Liang, X. Ren, Z. Zhang, and Y. Ma. Repairing sparse low-rank texture. In *12th European Conference on Computer Vision*, pages 482–495. Springer, 2012.
28. Y. Luo, K. He, D. Xu, H. Shi, and W. Yin. Infrared and visible image fusion based on hybrid multi-scale decomposition and adaptive contrast enhancement. *Signal Processing: Image Communication*, page 117228, 2024.
29. Y. Meyer. *Oscillating patterns in image processing and nonlinear evolution equations: the fifteenth Dean Jacqueline B. Lewis memorial lectures*, volume 22. American Mathematical Society., 2001.
30. M. K. Ng, R. H. Chan, and W.-C. Tang. A fast algorithm for deblurring models with Neumann boundary conditions. *SIAM Journal on Scientific Computing*, 21(3):851–866, 1999.
31. S. Ono, T. Miyata, and I. Yamada. Cartoon-texture image decomposition using blockwise low-rank texture characterization. *IEEE Transactions on Image Processing*, 23(3):1128–1142, 2014.
32. C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982.
33. H. Pan, Y.-W. Wen, and Y. Huang. L_0 gradient-regularization and scale space representation model for cartoon and texture decomposition. *IEEE Transactions on Image Processing*, 2024.
34. Z.-F. Pang, G. Meng, H. Li, and K. Chen. Image restoration via the adaptive tvp regularization. *Computers & Mathematics with Applications*, 80(5):569–587, 2020.
35. Z.-F. Pang, H.-L. Zhang, S. Luo, and T. Zeng. Image denoising based on the adaptive weighted tvp regularization. *Signal Processing*, 167:107325, 2020.

36. P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.
37. L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.
38. H. Schaeffer and S. Osher. A low patch-rank interpretation of texture. *SIAM Journal on Imaging Sciences*, 6(1):226–262, 2013.
39. B. Shi, G. Meng, Z. Zhao, and Z.-F. Pang. Image decomposition based on the adaptive direction total variation and g-norm regularization. *Signal, Image and Video Processing*, 15(1):155–163, 2021.
40. M. Tao and X. Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM Journal on Optimization*, 21(1):57–81, 2011.
41. L. A. Vese and S. J. Osher. Image denoising and decomposition with total variation minimization and oscillatory functions. *Journal of Mathematical Imaging and Vision*, 20(1):7–18, 2004.
42. C. Vogel and M. Oman. Iterative method for total variation denoising. *SIAM Journal on Scientific Computing*, 17:227–238, 1996.
43. C. Wang, L. Xu, and L. Liu. Structure–texture image decomposition via non-convex total generalized variation and convolutional sparse coding. *The Visual Computer*, 39(3):1121–1136, 2023.
44. W. Wang and J. Wang. A variational model for cartoon-texture decomposition of a color image. *Journal of Computational and Applied Mathematics*, 449:115932, 2024.
45. Y.-W. Wen, H.-W. Sun, and M. K. Ng. A primal-dual method for the Meyer model of cartoon and texture decomposition. *Numerical Linear Algebra with Applications*, 26(2):e2224, 2019.
46. B. Wohlberg and P. Rodriguez. An iteratively reweighted norm algorithm for minimization of total variation functionals. *IEEE Signal Processing Letters*, 14(12):948–951, 2007.
47. R. Wolke and H. Schwetlick. Iteratively reweighted least squares: algorithms, convergence analysis, and numerical comparisons. *SIAM Journal on Scientific and Statistical Computing*, 9(5):907–921, 1988.
48. H.-Y. Yan and Z. Zheng. Image cartoon-texture decomposition by a generalized non-convex low-rank minimization method. *Journal of the Franklin Institute*, 361(2):796–815, 2024.
49. I. Zachevsky and Y. Y. J. Zeevi. Statistics of natural stochastic textures and their application in image denoising. *IEEE Transactions on Image Processing*, 25(5):2130–2145, 2016.
50. F. Zhang, X. Ye, and W. Liu. Image decomposition and texture segmentation via sparse representation. *IEEE Signal Processing Letters*, 15:641–644, 2008.
51. H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan. Hyperspectral image restoration using low-rank matrix recovery. *IEEE Transactions on Geoscience and Remote Sensing*, 52(8):4729–4743, 2013.
52. Y. Zhang, L. Chen, Z. Zhao, and J. Jia. Multi-focus image fusion based on cartoon-texture image decomposition. *Optik*, 127(3):1291–1296, 2016.
53. Z. Zhang, A. Ganesh, X. Liang, and Y. Ma. Tilt: Transform invariant low-rank textures. *International Journal of Computer Vision*, 99(1):1–24, 2012.
54. Z. Zhang and H. He. A customized low-rank prior model for structured cartoon–texture image decomposition. *Signal Processing: Image Communication*, 96:116308, 2021.
55. X. Zhou, C. Yang, H. Zhao, and W. Yu. Low-rank modelling and its applications in image analysis. *ACM Computing Surveys*, 47(2):1–33, 2014.
56. Z. Zhu, H. Yin, Y. Chai, Y. Li, and G. Qi. A novel multi-modality image fusion method based on image decomposition and sparse representation. *Information Sciences*, 432:516–529, 2018.