The Visual Computer (2023) 39:1121–1136
https://doi.org/10.1007/s00371-021-02392-z

ORIGINAL ARTICLE

Structure–texture image decomposition via non-convex total generalized variation and convolutional sparse coding

Chunxue Wang[1] · Linlin Xu[2] · Ligang Liu[3]

Accepted: 23 December 2021 / Published online: 7 February 2022

Abstract
Image decomposition is a fundamental but challenging ill-posed problem in image processing and has been widely applied to compression, enhancement, texture removal, etc. In this paper, we introduce a novel structure–texture image decomposition model via non-convex total generalized variation regularization (NTGV) and convolutional sparse coding (CSC). NTGV aims to characterize the detailed-preserved structural component ameliorating the staircasing artifacts existing in total variation-based models, and CSC aims to characterize image fine-scale textures. Moreover, we incorporate both structure-aware and texture-aware measures to well distinguish structural and textural component. The proposed model is numerically implemented by an alternating minimization scheme based on alternating direction method of multipliers. Experimental results demonstrate the effectiveness of our approach on several applications including texture removal, high dynamic range image tone mapping, detail enhancement and non-photorealistic abstraction.

Keywords Structure–texture image decomposition · Non-convex total generalized variation regularization · Convolutional sparse coding · Alternating minimization scheme · Detail-preserving

1 Introduction

Image decomposition is a key step in image analysis to extract two semantically meaningful components from an image. That is, given an input image f, the image decomposition problem can be written as $f = u_C + v_T$, where the unknown u_C represents structural component modeling the well-structured homogeneous regions and the unknown v_T represents textural component defining oscillating patterns such as texture and/or noise. Image denoising and structure–texture image decomposition are two types of representative decomposition problems, focusing on how to formulate two parts with different priors. In this paper, we mainly focus on the structure–texture decomposition and the difficulties mainly reflect in two aspects. Firstly, this is an ill-posed problem since the number of unknowns is twice as many as the number of known variables, and how to employ appropriate constraints to reach a unique solution is challenging. Secondly, as there is no clear distinction between structures and textures among different images and even within one image, how to define the scales of both parts and extract structures and textures consistent with Human Visual System (HVS) is an another difficult problem.

Numerous techniques have been developed to implement decomposition models. For example, different variational techniques have been developed to separate image structures and textures by utilizing different regularizations. Early methods always applied total variation (TV) [50] and its extensions including to extract the piecewise constant structural part and characterized the textural part with different functional spaces or norms, such as TV-G [42], TV-div(L^p) [57], TV-H^{-1} [46], TV-H^{-s} [32], TV-div(BMO) [30], TV-L^1 [72], RTV [66], JCAS [19]. However, these methods often over-smooth the structural component. Filtering-based methods were proposed to filter out textures from the structural image according to edge strength [12,13,29,31,51,70,71] or spacial scale [17,24,67,75,77]. Commonly, the above work can remove weak or smaller-scale edges and preserve the strong or larger-scale ones. However,

✉ Chunxue Wang
chunxuewang2019@163.com

1 Dunhuang Academy, Jiuquan, Gansu, China

2 The School of Computer Information Management, Inner Mongolia University of Finance and Economics, Hohhot, Inner Mongolia, China

3 The School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui, China

these methods still cannot describe the distinction between image structure and texture, especially when weak edges belong to image structure, while strong ones correspond to the texture. Recently, deep neural networks (DNNs) have been proposed in the image decomposition [8,14,15,28,65]. Since there is no ground truth of image structure and texture, these methods are dependent on man-made datasets via sparse annotations or a large number of external samples and perform unstable on real-world data. In addition, STD [76] proposed a model of self-example, unsupervised and online learning, but still lost too many details of structural component due to the TV regularization, especially for the examples with undistinguishable image structures and textures.

In this paper, we take advantages of both the variational framework and sparse representation for structure–texture image decomposition. In particular, we propose to use non-convex total generalized variation regularization (NTGV) as a regularization term of image structure to capture more structural details and extract image fine-scale texture with convolutional sparse coding (CSC). To better distinguish textual edges from structural ones, both structure-aware and a texture-aware measures are incorporated. We develop an alternating minimization scheme based on alternating direction method of multipliers (ADMM) and show effectiveness of our method on several applications both quantitatively and qualitatively.

The rest of the paper is organized as follows. In Sect. 2, we review related works on structure–texture image decomposition. We present the proposed model in Sect. 3 and introduce the algorithm details in Sect. 4. A comprehensive analysis is provided in Sect. 5. In Sect. 6, we present the numerical experiments on several applications and make comparisons with the corresponding state-of-the-art methods.

2 Related work

Generally, image decomposition is also known as image layer separation including reflectance (albedo)-illumination (shading) decomposition and structure (cartoon)-texture decomposition. The former mainly involves intrinsic image decomposition and Retinex-based decomposition designed for image enhancement, surface re-texturing, object insertion and scene relighting, etc. While the latter aims to extract main information of the images successfully applied to HDR image tone mapping, detail enhancement, semantic image segmentation and non-photorealistic abstraction, etc. We have explored the total generalized variation regularization for Retinex-based decomposition in our previous work [58] and consider the non-convex TGV to separate a single image into structural and textural components in this paper. Therefore, we give a brief overview of related works on structure(cartoon)-texture decomposition in this section. For the past few decades, many implementations and improvements of structure–texture image decomposition have been studied, and we mainly discuss three classes of methodologies that are most relevant to ours: adaptive neighborhood filter, variational framework and deep learning-based method.

Adaptive neighborhood filter Image structure can be achieved easily by filtering out image details, and the well-known adaptive filters include bilateral filter [54], guided filter [22] and non-local filter [6]. To better preserve image edges, some extensions appeared subsequently, e.g., fast bilateral filter [13], gradient domain-guided filter [22], hashed non-local filter [6]. To suppress the structure halos, the filter of weighted least squares (WLS) [17] was proposed by introducing gradient-based smoothness weights and has been well applied to HDR tone mapping and detail manipulation. Inspired by WLS, Li et al. [31] proposed an edge-aware weighting strategy in the paradigm of guided filtering, the tree filtering [3] was designed by spatial, range, and tree distances to achieve strong image smoothing, and side window filtering (SWF) [70] was introduced to adaptively select the support domain of filtering. In addition, anisotropic diffusion equation [48] and zero crossings of the derivative of histogram [26] were proposed in a different mathematical form, and a pyramid of Laplacian filters [47] was presented to characterize edges by large-scale edges or small-scale details. Since the aforementioned methods are not designed to explicitly describe both image structures and textures, their performance on image decomposition is not satisfactory. Recently, more and more works [24,67,75,77] focused on the spatial scale of regions to distinguish image structures and textures. The rolling guidance filter (RGF) [75] employed scale space theory [33] to remove the small-scale edges and smoothed the large-scale edges with rolling guidance method iteratively. Jeon et al. [24] used patch-based statistics to find an optimal per-pixel smoothing scale. Zhou et al. [77] measured the scale with the help of a scanline along gradient direction through iterative global optimization (IGO). Xu et al. [67] adaptively adjusted the window size by the distances from pixels to structures. The neighborhood filters on spatial scale are intuitive and easy to implement, but still cannot well distinguish image structures and textures, especially for multiple scales of complex structures and textures.

Variational framework Variational models have been widely applied to image decomposition during the last two decades, aiming to find appropriate functional spaces to describe structures and textures, respectively. One of the most notable models is the TV approach [50], and early researches discussed the TV-based models characterizing textures by different functional spaces or norms [2], involving TV-G [42], TV-div(L^p) [57], TV-H^{-1} [46], TV-H^{-s} [32], TV-div(BMO) [30], TV-L^1 [72], etc. To well model textural component, sparse coding [19,52,74] and low rank [16]

were proposed in TV-based models. However, these models often over-smooth structural component as the property of TV. Beyond the TV [50], many other priors have also been proposed. Xu et al. [64] put forward L^0 gradient minimization to remove weak edges and retain strong edges. Similarly, Ono [45] formulated L^0-norm of gradients as the hard constraint solved by L^0 projection algorithm. The model of relative total variation (RTV) [66] used a pixel-wise windowed TV normalized by a windowed inherent variation, which is treated as a great progress in describing image textures. Many scale-ware measures in image decomposition models [21,24,67,77] essentially originated from RTV. Although different scale-aware measures were discussed, the resulting structure usually looked like piecewise constant, and thus structural edges were regarded as textural details. As a successful modified TV, total generalized variation (TGV) [4] has been applied in image decomposition to preserve structural edges. Following adaptive TV [34,37], Xu et al. [63] proposed the adaptive TGV to preserve the key features such as object boundaries and Liu [35] developed a weighted TGV-Gabor model. Following Meyer's TV-G model [42], Xu et al. [62] just used TGV regularization term in place of TV term in TV-G, aiming to capture structural component with more details by TGV, and Lu et al. [38] implemented TGV regularization term to model structural component and its dual TGV* for textural component. Although they all employed TGV to reduce the staircase effect and preserve the edges, they failed to recover the repetition or oscillation property of textures.

Deep learning-based method Deep learning-based methods have been developed for image decomposition recently. The deep edge-aware filters (DEF) [65] were proposed to approximate various filters based on a deep convolutional neural network with a gradient domain training procedure. Fan et al. [14] solved this problem by constructing two cascaded sub-networks, named as edge prediction network and image reconstruction network. Chen et al. [8] used a fully convolutional network to approximate a wide variety of variational models including the TV [50], RTV [66] and L^0 [64]. Kim et al. [28] learned the structure prior through context aggregation network (CAN) and generated training data by adding Gaussian noises to clean image patches. With the similar idea, Lu et al. [39] trained two prediction networks to highlight textural and structural parts on synthetic samples. However, all above data-driven approaches were highly relied on a large number of man-made training data. To overcome above difficulties, Fan et al. [15] employed a modified CAN on external samples and Zhou et al. [76] adopted modified U-net (MU-net) on self-example samples. Although both methods were in the unsupervised fashion, the learning variational priors still failed to distinguish image structures and textures well, especially for the complex cases.

3 NTGV + CSC decomposition model

In this section, we first review the TGV, NTGV regularization in Sect. 3.1 and CSC in 3.2 and then formulate our optimization for structure–texture image decomposition in Sect. 3.3.

3.1 Background of TGV and NTGV

TGV was first proposed by Bredies et al. [4] and has been widely applied in image processing [20,55,58] and geometry processing [36], written as follows:

$$\mathrm{TGV}^k_{\boldsymbol{\alpha}}(u) = \sup_{v} \left\{ \int_{\Omega} u \mathrm{div}^k(v) \mathrm{d}x \mid v \in \mathcal{C}^k_c(\Omega, \mathrm{Sym}^k(\mathbb{R}^d)), \right. \\ \left. \|\mathrm{div}^j(v)\|_\infty \le \alpha_j, \quad j = 0, \ldots, k-1 \right\}, \tag{1}$$

where $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_{k-1})$ is a fixed positive vector, $\mathrm{Sym}^k(\mathbb{R}^d)$ denotes the space of symmetric k-tensors on $\mathbb{R}^d$ as $\mathrm{Sym}^k(\mathbb{R}^d) = \{\xi : \mathbb{R}^d \times \cdots \times \mathbb{R}^d | \ \xi \, is \, k - linear \, and \, symmetric\}$, and $\mathcal{C}^k_c(\Omega, \mathrm{Sym}^k(\mathbb{R}^d))$ is the space of compactly supported symmetric tensor fields.

Meanwhile, $\mathrm{TGV}^k_{\boldsymbol{\alpha}}$ can be interpreted as a k-fold infimal convolution by employing the Fenchel−Rockafellar duality formula, as follows:

$$\mathrm{TGV}^k_{\boldsymbol{\alpha}}(u) = \inf_{u_j} \sum_{j=1}^{k} \boldsymbol{\alpha}_{k-j} \int_{\Omega} |\mathcal{E}(u_{j-1}) - u_j| \mathrm{d}x, \tag{2}$$

where $u_0 = u$, $u_k = 0$, $u_j \in \mathcal{C}^{k-j}_c(\Omega, \mathrm{Sym}^j(\mathbb{R}^d))$ for $j = 1, \ldots, k$, and $\mathcal{E}$ represents the distributional symmetrized derivative, i.e., $\mathcal{E}(u_{j-1}) = \frac{\nabla u_{j-1} + (\nabla u_{j-1})^T}{2}$. Thus, $\mathrm{TGV}^k_{\boldsymbol{\alpha}}$ automatically balances the first to the k-th derivatives of u among themselves and reduces the staircasing effects of the TV.

Recently, studies [25,43,44] have demonstrated that the non-convex variant of TGV performs better than the convex one for image restoration by preserving edges, which can be written as follows:

$$\mathrm{NTGV}^k_{\boldsymbol{\alpha}}(u) = \inf_{u_j} \sum_{j=1}^{k} \boldsymbol{\alpha}_{k-j} \int_{\Omega} \phi_i(|\mathcal{E}(u_{j-1}) - u_j|) \mathrm{d}x, \tag{3}$$

where $\{\phi_i\}$ are a list of non-convex prototype functions, such as the regularized ℓ_p-norm $\phi_i(s) = (s+\varepsilon_i)^p$ with $0 < p < 1$ and $\varepsilon_i > 0$, or log-function $\phi_i(s) = \frac{1}{\mu_i} \log(1 + \mu_i s)$ or $\phi_i(s) = \frac{1}{\mu_i} \log(1 + \mu_i s^2)$ with $\mu_i > 0$.

It is hard to solve the minimization problem involving the non-convex ℓ_p-norm regularizer because finding a limiting-supergradient of $\|\cdot\|_q$ at zero is difficult. Moreover, according

to the researches in [25,44,56], the log-function is suitable for the reconstruction of images being piecewise smooth and having some sharp jump discontinuities. Structural images are mostly piecewise-smooth; thus, we particularly utilize the second-order NTGV denoted by $\mathrm{NTGV}_{\boldsymbol{\alpha}}^2$ in this work.

3.2 Background of CSC

Sparse representation encodes a signal vector $\boldsymbol{y}$ as the linear combination of a few atoms in a dictionary $\boldsymbol{D}$ by solving the following inverse problem:

$$\arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{D}\boldsymbol{x}-\boldsymbol{y}\|_2^2+\lambda\|\boldsymbol{x}\|_1, \tag{4}$$

Over the past two decades, this optimization was solved independently on a set of overlapping image patches covering the image[9,53] and has achieved state-of-the-art results in various computer vision tasks [41,49,61,68]. As a category of sparse representations, the CSC was first proposed by Zeiler et al. [73] to decompose the input image into N sparse feature maps by N filters, instead of sparsely representing a vector by the linear combination of dictionary atoms, replacing (4) with

$$\arg\min_{\{\boldsymbol{x}_m\}} \frac{1}{2}\|\sum_m \boldsymbol{d}_m * \boldsymbol{x}_m-\boldsymbol{y}\|_2^2+\lambda\sum_m\|\boldsymbol{x}_m\|_1, \tag{5}$$

where $\{\boldsymbol{d}_m\}$ is a set of N dictionary filters, $*$ denotes convolution, and $\boldsymbol{x}_m$ is a set of coefficient maps, each of which is the same size as $\boldsymbol{y}$.

The convolutional decomposition avoids dividing the whole image into overlapped patches and can naturally utilize the consistency prior in the decomposition procedure. The convolutional decomposition has made great progress in numerical optimization. Zeile et al. [73] adopted an alternating minimization of two splitting subproblems, where one problem is how to solve a large linear system by an iterative method, and the other problem is about a simple shrinkage. Other algorithms operating in the spatial domain have been proposed including coordinate descent [27] and a proximal gradient method [7]. Bristow et al. [5] proposed a fast CSC algorithm by considering the property of block circulant with circulant block (BCCB) matrix in the Fourier domain, and Wohlberg [60] further improved above algorithm by introducing an efficient method for solving the linear systems that is linear in the number of filters, instead of cubic.

3.3 Proposed model

Here, we present our NTGV + CSC model for structure–texture decomposition, recovering the structural component with more details by NTGV and the textural component with repetition property by CSC, for an $m \times n$ input image $\boldsymbol{y}$, we characterize its structural part $\boldsymbol{u}$ with $\mathrm{NTGV}_{\boldsymbol{\alpha}}^2(\boldsymbol{u})$, approximate textural part $\boldsymbol{v}$ by the sum of N convolutions of $s \times s$ filter $\boldsymbol{f}_i$ and sparse feature map $\boldsymbol{Z}_i$ with size $(m+s-1)\times(m+s-1)$. The image decomposition is achieved by solving the following objective function:

$$\begin{aligned}&\min_{\boldsymbol{u},\boldsymbol{f},\boldsymbol{Z}} \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{u}-\sum_{i=1}^{N}\boldsymbol{f}_i*\boldsymbol{Z}_i\|_F^2+\omega_s\mathrm{NTGV}_{\boldsymbol{\alpha}}^2(\boldsymbol{u})+\lambda\sum_{i=1}^{N}\|\boldsymbol{Z}_i\|_1,\\ &\text{s.t. } \|\boldsymbol{f}_i\|_F^2\le 1 \quad for \quad i=1,\ldots,N\end{aligned} \tag{6}$$

where λ is a positive parameter controlling the L_1 penalty, ω_s is a weight function defining on each pixel to distinguish structural edges or textural edges, and the inequality constraints on the filter $\boldsymbol{f}_i$ prevent the dictionary from absorbing all the system's energy. The term $\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{u}-\sum_{i=1}^{N}\boldsymbol{f}_i*\boldsymbol{Z}_i\|_F^2$ is used for fidelity, and $\mathrm{NTGV}_{\boldsymbol{\alpha}}^2(\boldsymbol{u})$ is minimized over all gradients of the deformation field $\boldsymbol{p}=(p_1,p_2)$ on image space Ω, which reads

$$\inf_{\boldsymbol{p}\in\mathcal{C}_c^2(\Omega,\mathbb{R}^2)} \alpha_1\int_\Omega \phi_1(|\nabla\boldsymbol{u}-\boldsymbol{p}|)\,\mathrm{d}x+\alpha_0\int_\Omega\phi_2(|\mathcal{E}(\boldsymbol{p})|)\,\mathrm{d}x, \tag{7}$$

where α_0 and α_1 are positive parameters balancing between the first- and second-order derivative of $\boldsymbol{u}$, $\phi_i(s)=\frac{1}{\rho_i}\log(1+\rho_i s)$ with the parameters $\rho_i>0$ controlling the non-convexity of the first- and second-order regularization terms.

The Setting of ω_s The weight function ω_s plays an important role in highlighting the pixels from structural edges or textural edges and guiding a good image decomposition. Following the work [76], we introduce two measures: structure-aware measure and texture-aware measure. The former can reveal the main structural information of an image, and the latter describes the repetition property of textures.

Taking advantage of anisotropy property, structural gradients would have a dominant orientation, representing by one of the eigenvectors of the following positive semi-definite matrix $\mathbf{J}$ [1]:

$$\mathbf{J}(i)=\begin{bmatrix}\mathbf{g}_x^T(i)\mathbf{g}_x(i) & \mathbf{g}_x^T(i)\mathbf{g}_y(i)\\ \mathbf{g}_y^T(i)\mathbf{g}_x(i) & \mathbf{g}_y^T(i)\mathbf{g}_y(i)\end{bmatrix},$$

where i indicates the location of the patch, and $\mathbf{g}_x$ and $\mathbf{g}_y$ are the vectors containing gradients of each pixel in the patch along the abscissa and the ordinate, respectively. The matrix $\mathbf{J}$ has two non-negative eigenvalues denoted as λ_1 and λ_2, and its dominant direction can be represented by the eigenvectors corresponding to the lower one. For any patch, its anisotropy degree A can be calculated by [23]

$$A_{\mathbf{J}}=\frac{\max(\lambda_1,\lambda_2)-\min(\lambda_1,\lambda_2)}{\lambda_1+\lambda_2}, \tag{8}$$

where $\max(\cdot,\cdot)$ and $\min(\cdot,\cdot)$ return the maximum and minimum value of their two arguments, respectively. From Eq. 8, the range of $A_{\mathbf{J}}$ is from 0 to 1, and the larger $A_{\mathbf{J}}$ of one patch indicates that it is more likely to be a patch of image structure. Taking edge strength into consideration, the structure-aware measure M_s is designed as

$$M_s(i) = A_{\mathbf{J}}(i)\frac{\|\nabla f(i)\|_2}{R}, \tag{9}$$

where $f(i)$ is the pixel intensity at pixel i, $\|\nabla f(i)\|_2$ denotes the gradient magnitude in L_2 norm, R is fixed as 255 for 8-bit images.

To define texture-aware measure, we employ the histogram of oriented gradients (HOG) [10] for feature extraction same as [76]. Specifically, given a pixel i, we determine repetitive patterns by going through all its nearby pixels and calculate the texture-aware measure M_t as follows

$$M_t(i) = \frac{1}{\#(C_{N(i)})}\sum_{j\in C_{N(i)}}\cos\theta_{ij}\exp(-D_{KL}(\mathbf{h}_j||\mathbf{h}_i)), \tag{10}$$

where θ_{ij} is the angle between edge direction $\mathbf{v}_{ij}$ and gradient direction, $D_{KL}(\cdot||\cdot)$ denotes the Kullback-Leibler (KL) divergence, $\mathbf{h}_i$ and $\mathbf{h}_j$ are the HOG features of pixels i and j, respectively, $\#(C_{N(i)})$ counts the element number in the set consisted of pixels with noticeable gradients in the neighborhood of pixel i.

Based on above, it is straightforward to combine both structure-aware measure and texture-aware measure to guide our decomposition instead of using the same value among all pixels. For the edge pixel from structural edges, i.e., M_s is higher and M_t is lower, we hope they can be preserved as sharp as the input by reducing the impact of the $\mathrm{NTGV}^2_{\boldsymbol{\alpha}}(\boldsymbol{u})$. For the edge pixel from textural edges, i.e., M_s is lower and M_t is higher, we hope they can be removed by increasing the impact of $\mathrm{NTGV}^2_{\boldsymbol{\alpha}}(\boldsymbol{u})$. So the weight function ω_s on pixel i is defined as

$$\omega_s(i) = (1-t)(1-M_s(i)) + tM_t(i), \quad 0 < t < 1. \tag{11}$$

4 Optimization algorithm for image decomposition

In this section, we develop an efficient implementation for solving problem (6). First of all, we define linear operators $\boldsymbol{F}_i$ such that $\boldsymbol{F}_i z_i = \boldsymbol{f}_i * \boldsymbol{Z}_i$, where z_i is the vectorization of the feature map $\boldsymbol{Z}_i$. Based on this, problem (6) is equivalent to

$$\begin{aligned}&\min_{\boldsymbol{u},\boldsymbol{f},z}\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{u}-\sum_i^N \boldsymbol{F}_i z_i\|_F^2 + \omega_s\mathrm{NTGV}^2_{\boldsymbol{\alpha}}(\boldsymbol{u}) + \lambda\sum_{i=1}^N\|z_i\|_1,\\ &\text{s.t. } \|\boldsymbol{f}_i\|_F^2 \le 1 \, for \;\; i = 1,\ldots,N\end{aligned} \tag{12}$$

where $\boldsymbol{F}_i$ is the corresponding BCCB matrix of filter $\boldsymbol{f}_i$. Note that the three unknowns $\boldsymbol{u}$, $\boldsymbol{f}$ and $\boldsymbol{z}$ are coupled in (12), our idea is to update three variables alternatively and the details of each subproblem are summarized in Algorithm 1.

Algorithm 1 Image decomposition via NTGV and CSC.

Input: Image $\boldsymbol{y}$, regularization parameter λ.

Initialization: feature map $z^0 = \mathbf{0}$, weight function ω_s by (11).

for $t = 1 : K$ **do**

Update $\boldsymbol{u}^t$ by (16);

if $t == 1$, initialize $\{\boldsymbol{f}_i^1\}_{i=1,\ldots,N}$ as the PCA basis of the patches in the residual image $\boldsymbol{y}-\boldsymbol{u}^1$;

Update $\{z_i^t\}_{i,1,\ldots,N}$ by (21);

Update $\{\boldsymbol{f}_i^t\}_{i=1,\ldots,N}$ by (23);

end for

Output: Structural part $\boldsymbol{u}^t$ and textural part $\boldsymbol{v}^t = \sum_i^N \boldsymbol{F}_i^t z_i^t$.

Updating $\boldsymbol{u}$ Fixing $\{\boldsymbol{f}_i\}_{i=1,\ldots,N}$ and $\{z_i\}_{i=1,\ldots,N}$, we solve the $\boldsymbol{u}$ subproblem by optimizing the following problem:

$$\min_{\boldsymbol{u}}\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{u}-\sum_i^N \boldsymbol{F}_i z_i\|_F^2 + \omega_s\mathrm{NTGV}^2_{\boldsymbol{\alpha}}(\boldsymbol{u}) \tag{13}$$

where the $\mathrm{NTGV}^2_{\boldsymbol{\alpha}}(\boldsymbol{u})$ is introduced in Sect. 3.1. Denote $\boldsymbol{x} = \boldsymbol{y} - \sum_i^N \boldsymbol{F}_i z_i$, and we adopt the iteratively reweighted l_1 algorithm (IRLA) [43]. Introducing auxiliary variables $\boldsymbol{d}$ and $\boldsymbol{w}$, we can reformulate the unconstrained problem in (13) into the following constrained problem as

$$\begin{aligned}&\min_{\boldsymbol{u}}\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{u}\|_F^2 + \omega_s\alpha_1\phi_1(|\boldsymbol{d}|) + \omega_s\alpha_0\phi_2(|\boldsymbol{w}|)\\ &\text{s.t. } \boldsymbol{d} = \nabla\boldsymbol{u} - \boldsymbol{p}, \boldsymbol{w} = \mathcal{E}(\boldsymbol{p}).\end{aligned} \tag{14}$$

Hence, the IRLA can be applied to (14), and a convex minimization problem to update $(\boldsymbol{u}^{k+1}, \boldsymbol{p}^{k+1}, \boldsymbol{d}^{k+1}, \boldsymbol{w}^{k+1})$ is given by

$$\begin{aligned}&\min_{\boldsymbol{u},\boldsymbol{p},\boldsymbol{d},\boldsymbol{w}}\left\{\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{u}\|_F^2 + \omega_s\alpha_1\langle\tilde{\boldsymbol{d}^k}, |\boldsymbol{d}|\rangle + \omega_s\alpha_0\langle\tilde{\boldsymbol{w}^k}, |\boldsymbol{w}|\rangle\right\}\\ &\text{s.t. } \boldsymbol{d} = \nabla\boldsymbol{u} - \boldsymbol{p}, \boldsymbol{w} = \mathcal{E}(\boldsymbol{p}).\end{aligned} \tag{15}$$

where $\tilde{\boldsymbol{d}}^k = \frac{1}{\rho_1|\boldsymbol{d}^k|+1}$ and $\tilde{\boldsymbol{w}}^k = \frac{1}{\rho_2|\boldsymbol{w}^k|+1}$.

Problem (15) can be solved by ADMM algorithm, which yields

$$\begin{cases} \boldsymbol{d}^{k+1} = \underset{\boldsymbol{d}}{\operatorname{argmin}}\, \omega_s\alpha_1\langle\tilde{\boldsymbol{d}}^k, |\boldsymbol{d}|\rangle - (\lambda_1^k)^\intercal\boldsymbol{d} + \frac{\mu}{2}\|\boldsymbol{d} - \nabla\boldsymbol{u}^k + \boldsymbol{p}^k\|_2^2, \\ \boldsymbol{w}^{k+1} = \underset{\boldsymbol{w}}{\operatorname{argmin}}\, \omega_s\alpha_0\langle\tilde{\boldsymbol{w}}^k, |\boldsymbol{w}|\rangle - (\lambda_2^k)^\intercal\boldsymbol{w} + \frac{\mu}{2}\|\boldsymbol{w} - \mathcal{E}(\boldsymbol{p}^k)\|_2^2, \\ \boldsymbol{u}^{k+1} = \underset{\boldsymbol{u}}{\operatorname{argmin}}(\lambda_1^k)^\intercal\nabla\boldsymbol{u} + \frac{\mu}{2}\|\boldsymbol{d}^{k+1} - \nabla\boldsymbol{u} + \boldsymbol{p}^k\|^2 + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{u}\|_F^2, \\ \boldsymbol{p}^{k+1} = \underset{\boldsymbol{p}}{\operatorname{argmin}} -(\lambda_1^k)^\intercal\boldsymbol{p} + (\lambda_2^k)^\intercal\mathcal{E}(\boldsymbol{p}) + \frac{\mu}{2}\|\boldsymbol{w}^{k+1} - \mathcal{E}(\boldsymbol{p})\|_2^2, \\ \qquad + \frac{\mu}{2}\|\boldsymbol{d}^{k+1} - \nabla\boldsymbol{u}^{k+1} + \boldsymbol{p}\|_2^2, \\ \lambda_1^{k+1} = \lambda_1^k - \gamma\mu(\boldsymbol{d}^{k+1} - \nabla\boldsymbol{u}^{k+1} + \boldsymbol{p}^{k+1}), \\ \lambda_2^{k+1} = \lambda_2^k - \gamma\mu(\boldsymbol{w}^{k+1} - \mathcal{E}(\boldsymbol{p}^{k+1})). \end{cases} \tag{16}$$

where the penalty parameter $\mu > 0$ is a fixed constant, and the relaxation $\gamma \in (0, (\sqrt{5}+1)/2]$ is required for the convergence of the ADMM algorithm.

To solve the above problem, we first utilize forward differences to approximate $\nabla\boldsymbol{u}$ and the symmetrized derivative $\mathcal{E}(\boldsymbol{p})$ in the following, respectively.

$$\begin{aligned} \nabla\boldsymbol{u} &= D\boldsymbol{u} = (D_1\boldsymbol{u}, D_2\boldsymbol{u})^\intercal, \\ \mathcal{E}(\boldsymbol{p}) &= (D_1\boldsymbol{p}_1, (D_2\boldsymbol{p}_1 + D_1\boldsymbol{p}_2)/2, \\ &\quad (D_2\boldsymbol{p}_1 + D_1\boldsymbol{p}_2)/2, D_2\boldsymbol{p}_2)^\intercal. \end{aligned}$$

where D_1 and D_2 are the circulant matrices corresponding to the forward finite difference operators with periodic boundary conditions along x-axis and y-axis, respectively.

The subproblems for $\boldsymbol{d}^{k+1}$ and $\boldsymbol{w}^{k+1}$ in (16) have closed-form solutions using shrink operator as follows:

$$\boldsymbol{d}^{k+1} = \text{shrink}\left(\frac{\lambda_1^k}{\mu} + \nabla\boldsymbol{u}^k - \boldsymbol{p}^k, \frac{\omega_s\alpha_1\tilde{\boldsymbol{d}}^k}{\mu}\right), \tag{17}$$

$$\boldsymbol{w}^{k+1} = \text{shrink}\left(\frac{\lambda_2^k}{\mu} + \mathcal{E}(\boldsymbol{p}^k), \frac{\omega_s\alpha_0\tilde{\boldsymbol{w}}^k}{\mu}\right), \tag{18}$$

where the shrink operator is defined as $\text{shrink}(s,t) = \frac{s}{|s|_1}\max(|s|_1 - t, 0)$.

The subproblems for $\boldsymbol{u}^{k+1}$ and $\boldsymbol{p}^{k+1}$ in (16) are both least square problems, and their corresponding normal equations are as follows:

$$\begin{aligned} &(\mu D^\intercal D + I)\boldsymbol{u}^{k+1} = \mu(\boldsymbol{x} - D^\intercal\lambda_1^k) \\ &\quad + D^\intercal(\boldsymbol{d}^{k+1} + \boldsymbol{p}^{k+1}), \end{aligned} \tag{19}$$

$$\begin{aligned} &(\mathcal{E}^*\mathcal{E} + I)\boldsymbol{p}^{k+1} = \mathcal{E}^*(\boldsymbol{w}^{k+1} - \frac{\lambda_2^k}{\mu}) + \frac{\lambda_1^k}{\mu} \\ &\quad + (D\boldsymbol{u}^{k+1} - \boldsymbol{d}^{k+1}). \end{aligned} \tag{20}$$

where $\mathcal{E}^*$ denotes the adjoint operator of $\mathcal{E}$.

The block matrices $D^\intercal D$ and $\mathcal{E}^*\mathcal{E}$ can be diagonalized by the Fourier transform under the periodic boundary condition. Thus, solutions $\boldsymbol{u}^{k+1}$ and $\boldsymbol{p}^{k+1}$ in our algorithm can be achieved explicitly and easily using the Fourier transform and the block matrix inversion formula.

Finally, the IRLA with ADMM algorithm for $\boldsymbol{u}$-step in (16) is summarized in Algorithm 2.

Algorithm 2 The IRLA with ADMM algorithm for $\boldsymbol{u}$ subproblem

Set parameters: $\alpha_0, \alpha_1, \gamma, \mu, \rho_1$ and ρ_2.
Initialize: $k = 0$, $\boldsymbol{u}^0 = \boldsymbol{y} - \boldsymbol{v}^0$, $\boldsymbol{p}^0 = \boldsymbol{0}$, $\boldsymbol{d}^0 = \nabla\boldsymbol{u}^0$, $\boldsymbol{w}^0 = \boldsymbol{0}$ and $\lambda_1^k = \lambda_2^k = \boldsymbol{0}$.
Repeat

1) Compute $\boldsymbol{d}^{k+1}$ according to (17),
2) Compute $\boldsymbol{w}^{k+1}$ according to (18),
3) Compute $\boldsymbol{u}^{k+1}$ according to (19),
4) Compute $\boldsymbol{p}^{k+1}$ according to (20),
5) Update Lagrange multipliers λ_1^{k+1} and λ_2^{k+1} according to (16),

Until $\frac{\|\boldsymbol{u}^{k+1} - \boldsymbol{u}^k\|_2}{\|\boldsymbol{u}^{k+1}\|_2} < \epsilon_3$,
Return $\boldsymbol{u}^{k+1}$.

Updating z Fixing structural part $\boldsymbol{u}$ and filters $\{\boldsymbol{f}_i\}_{i=1,\ldots,N}$, we solve the following subproblem to obtain z^{t+1}:

$$\min_z \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{u} - \sum_i^N \boldsymbol{F}_i\boldsymbol{z}_i\|_F^2 + \lambda\sum_{i=1}^N\|\boldsymbol{z}_i\|_1. \tag{21}$$

The optimization in (21) is a standard convolutional sparse coding problem, which can be solved by the ADMM algorithm in the Fourier domain to substantially reduce computational cost [60].

Updating $\boldsymbol{f}$ To recover $\boldsymbol{f}$, we first apply an efficient variable reordering $\sum_i^N \boldsymbol{F}_i\boldsymbol{z}_i = \hat{\boldsymbol{Z}}\boldsymbol{f}$, where $\boldsymbol{f}$ is written as the vectorization of all the filters $\boldsymbol{f}_i, i = 1, \ldots, N$, $\hat{\boldsymbol{Z}} = [\hat{\boldsymbol{Z}}_1, \ldots, \hat{\boldsymbol{Z}}_i, \ldots, \hat{\boldsymbol{Z}}_N]$ and $\hat{\boldsymbol{Z}}_i$ is generated by collecting the patches in $\boldsymbol{Z}_i$. The $\boldsymbol{f}$ subproblem can be re-written as the following equivalent form:

$$\begin{aligned} &\min_{\boldsymbol{f}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{u} - \hat{\boldsymbol{Z}}\boldsymbol{f}\|_F^2, \\ &\text{s.t. } \|\boldsymbol{f}_i\|_F^2 \le 1 \quad for \quad i = 1, \ldots, N \end{aligned} \tag{22}$$

Following the optimization in JCAS [19], we adopt a proximal gradient descent scheme to solve (22), which namely

$$\boldsymbol{f}^{t+1} = Prox_{\|\cdot\|\le 1}(\boldsymbol{f}^t - \tau\hat{\boldsymbol{Z}}^\intercal(\boldsymbol{y} - \boldsymbol{u} - \hat{\boldsymbol{Z}}\boldsymbol{f}^t)). \tag{23}$$

Since our proposed model is non-convex, it is important to set an appropriate initialization and the optimization order of the variables to ensure the convergence of algorithm. In our implementation, we initialize the textural part as an all-zero matrix and solve the $\boldsymbol{u}^1$ subproblem first. Then, we initialize filters $\{\boldsymbol{f}_i^1\}_{i=1,\ldots,N}$ as the PCA dictionary of the residual image $\boldsymbol{y}-\boldsymbol{u}^1$. Based on $\boldsymbol{u}^1$ and $\boldsymbol{f}^1$, we recover the feature map $\boldsymbol{z}^1$ by solving convolutional sparse coding problem (21). These steps can guarantee that texture details are estimated from the residual image gradually, while keeping the details of the structural part.

5 Analysis

5.1 Implementation details

The proposed method is implemented using MATLAB R2018b on a windows 10 platform with an Intel Corei7 at 3.2 GHz and 8 GB RAM. We fix the regularization parameter $\lambda = 10^{-2}$ in objective function discussed in Sect. 5.4. For the parameters in Algorithm 2, the default setting is as follows: $\alpha_0 = 0.1$, $\alpha_1 = 0.2$, $\mu = 10$, $\gamma = 1.618$, $\rho_1 = 0.005$, $\rho_2 = 0.2$ and $\epsilon_3 = 5e^{-4}$. To calculate $\{\boldsymbol{f}_i\}_{i=1,\ldots,N}$, we fix the size and numbers of filters are 5×5 and 7, respectively. To calculate ω_s, we set balance parameter t to 0.6, the batch size of M_s to 3 and the neighborhood size of M_t to 6. Since our idea is to remove textual edges from structural image gradually, so we update M_s during the iterations and gradually reduce the impact of the second term by decreasing t in ω_s of (11). In our implementation, M_t is calculated once from the input and M_s is re-estimated from the updated structural result of previous step.

5.2 The detail-preserving property of our model

It is worthy to note that how our model removes textural edges while keeping sharp features of structural component. Firstly, the weight function ω_s plays an important role in locating structural and textural edges, which is a good guiding of our model to distinguish image structures and textures. Secondly, the non-convex TGV can well construct smooth regions without staircasing effects while preserving edges and details, especially for the images with much structures and strong edges or complex textures. Based above, our model can better preserve the details guided by ω_s. Here, we compare our proposed method with two most related works. Lu et al. [38] adopted the convex TGV to capture structural edges, and Gu et al. [19] employed synthesis sparse representation to capture textural edges, both methods either focused on recovering structural edges or textural edges, but cannot pursue both. We take account of both advantages and also combine the weight function ω_s to guide a good image decomposition. A visual example is given in Fig. 1, and the regions of red boxes are highlighted. The input images given in Fig. 1a, b and e show the image structures from different methods. From Fig. 1b, we can see that TGV-TGV* cannot remove the textural edges totally from structural component due to ignoring considering the repetition property of textures. For JCAS in Fig. 1c, small-scale structures are blurred or even removed because of the staircase effects from TV just as shown in the highlighted parts. Figure 1d shows the results of our decomposition model without ω_s, and NTGV improves the detail-preserving property greatly compared with JCAS. The best results are achieved by adding the weight function ω_s shown in Fig. 1e, from which we can observe that the structural edges are much sharper and the textural edges are removed clearly.

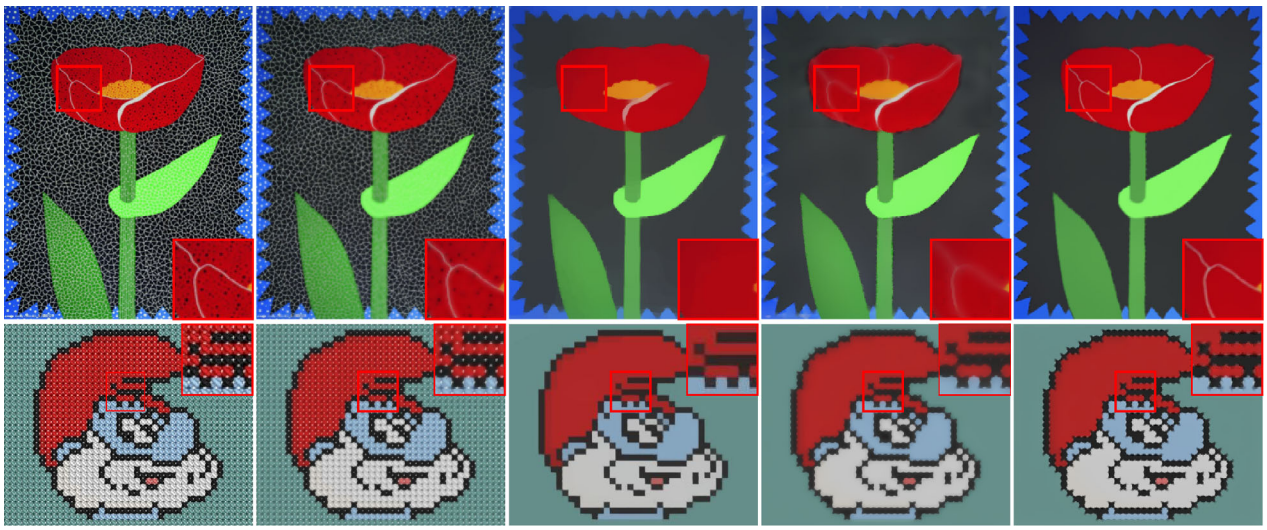
5.3 Convergence analysis

Note that the three unknowns $\boldsymbol{u}$, $\boldsymbol{f}$, $\boldsymbol{z}$ are coupled in (6), it is effective to apply an inner alternating minimization scheme to decrease the energy monotonically. However, it should be mentioned that such an alternating scheme does not guarantee the whole sequence convergence to the minimizer of the original problem. For fixed $\boldsymbol{f}$, $\boldsymbol{z}$, the problem is non-convex to $\boldsymbol{u}$ in (14) and we can obtain a partial convergence of the IRLA with ADMM algorithm in (15) (see [25,43,44] for more details). That is, the sequence $\{\boldsymbol{u}^k, \boldsymbol{v}^k, \boldsymbol{d}^k, \boldsymbol{w}^k\}$ generated by (15) is bounded and has at least one accumulation point. However, since the non-convex log function in (14) is not a sum of a convex function, we cannot even assure the global convergence of the algorithm in (15). For fixed $\boldsymbol{u}$, $\boldsymbol{f}$, the problem is convex to $\boldsymbol{z}$. And for fixed $\boldsymbol{u}$, $\boldsymbol{z}$, the problem is convex to $\boldsymbol{f}$. Since our objective function has a general lower bound 0, the $\boldsymbol{f}$ and $\boldsymbol{z}$ subproblems can be guaranteed to converge to their minimum, respectively.

Here, we compute average value of per-pixel intensity differences during iterations. Figure 2a shows the how the differences $\|\boldsymbol{u}^{k+1}-\boldsymbol{u}^k\|^2$ and $\|\boldsymbol{y}-\boldsymbol{u}^k-\boldsymbol{v}^k\|^2$ change over iterations. From the figure, we can see that our algorithm converges to a fixed point within a few iterations and both differences are less than 1.0×10^{-5} after 10 iterations. Meanwhile, Fig. 2a–d shows that textural edges are removed from structural edges gradually during iterations.

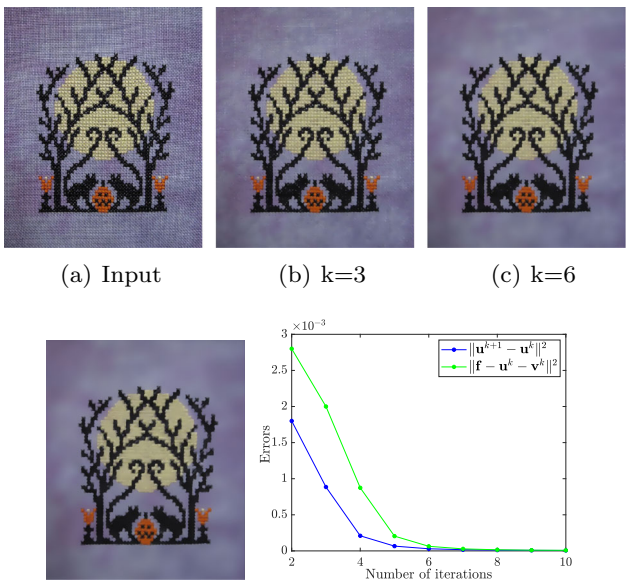
5.4 Parameters impact

The regularization parameter λ is important to distinguish structural edges and textural edges, controlling the sparsity of repetitive textures. In other words, the larger value of λ produces the more sparser textures, yielding less smoothing effects (and vice-versa). We show the resulting structural image $\boldsymbol{u}$ with various choices of λ in Fig. 3. The number of iterations K is fixed to 12 and feature map is initialized



(a) Inputs (b) TGV-TGV* [38] (c) JCAS [19] (d) Ours without ω_s (e) Ours

Fig. 1 Illustration on the detail-preserving property of our model. **a** input images; **b** image structures obtained by TGV–TGV* [38] where TGV regularization and its dual TGV* were adopted to model structural and textural component, respectively; **c** image structures obtained by JCAS [19] where ASR and CSC priors were integrated to separate two image layers, respectively; **d** image structures obtained by our decomposition model without ω_s; **e** image structures obtained by our proposed model. Our proposed model achieves more accurate and sharper structures and removes multiple-scale or extremely varying textures



(a) Input (b) k=3 (c) k=6

(d) k=10 (e) change of differences during iterations

Fig. 2 Illustration on the convergence of Algorithm 1. Our algorithm gradually separates textures from the input image and converges to a fixed point within few iterations

with zeros for all cases. As can be observed in Fig. 3, textures can be removed from the structural part gradually with the decreasing of λ. Different from JCAS [19], our method can well avoid over-smoothing of structural part thanks to the weight function ω_s and achieves almost the same results although the λ is less than 10^{-2}, just shown in Fig. 3e, f. Therefore, we fix λ to 10^{-2} for almost examples. Furthermore, compared with STD [76] in Fig. 3b, both results in Fig. 3e, f contain more sharper details and overcome the staircase effect of TV regularization.

5.5 Runtime

Since our algorithm is an alternating minimization scheme updating $\boldsymbol{u}$, $\boldsymbol{f}$ and $\boldsymbol{z}$, respectively, and the most time-consuming part is the $\boldsymbol{u}$ updating which consists of four subproblems. We compare the runtime of our method with those of the several decomposition methods including RTV [66], JCAS [19], IUL [15], SWF [70] and STD [76] and measure the runtime of the proposed method using gputimeit function built in MATLAB. We select 80 images with the same size of 350×350 and record the average running time over these images on the same computing machine provided in Table 1. Among the methods, IUL is the fastest due to its one forward propagation and RTV is the second since only the optimization of structures is considered. The efficiency of STD is dependent on the initialization of network, the scheme using standard initialization is time-consuming, while the one using pre-trained MU-net is comparable. Similar with traditional filters, SWF needs to perform calculations over multiple windows, and its computational cost is higher. JCAS and our method are both time-consuming mainly because

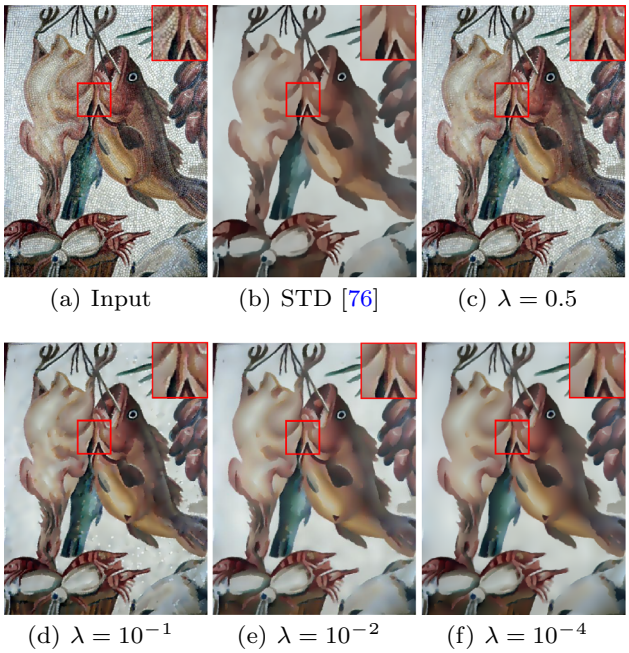


Fig. 3 Impact of parameter λ. **a** Input image; **b** structural image obtained by STD [76]; **c–f** structural images obtained by our method with various choices of λ. The smaller value of λ removes more textures, while almost similar results can be achieved when λ is less than or equal to $\lambda = 10^{-2}$

both involve the structural and textural image estimation via the alternating minimization scheme. However, both methods can guarantee a fast convergence in several iterations, so we set the maximum number of iterations as 10. With code optimization and implementing C programming, the computational speed may be significantly improved.

6 Applications

In this section, we demonstrate the effectiveness of our approach on several image decomposition applications, including texture removal, HDR tone mapping, detail manipulation and non-photorealistic abstraction. Meanwhile, we compare the proposed method with several state-of-the-art methods in each application. The code of the LowRank [16] is written by ourselves, while the results of other competing methods for comparison are from the source codes or already-trained models obtained from the original authors and parameters are set to be optimal according to the original papers.

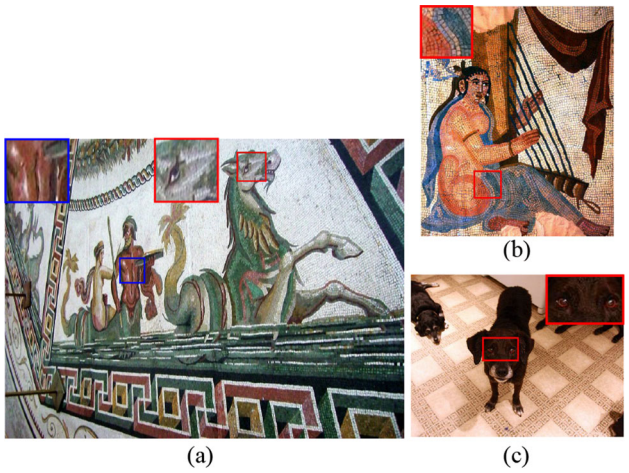


Fig. 4 Test images for the texture removal

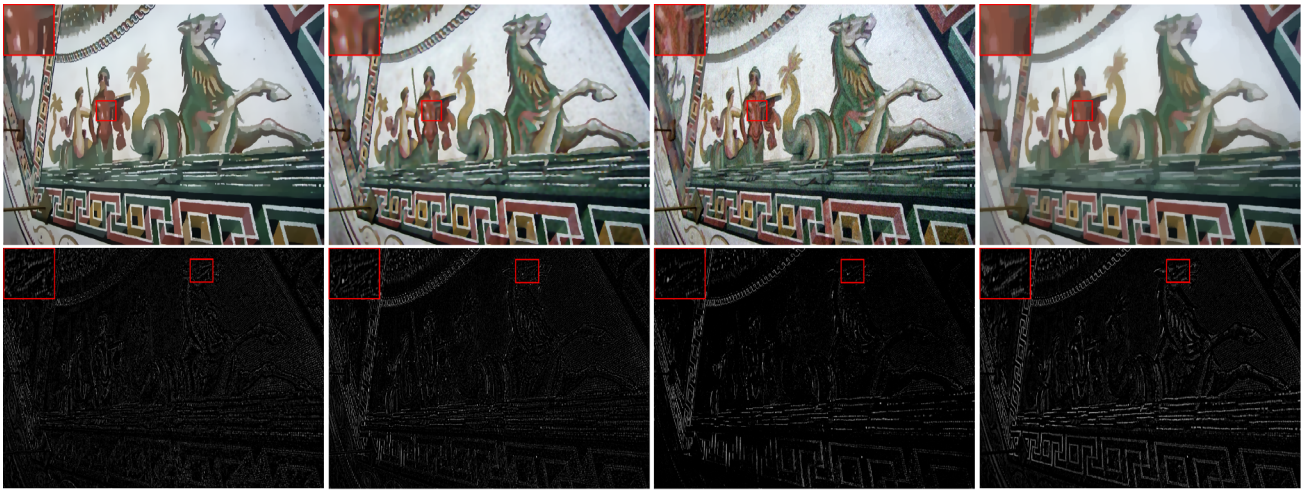
6.1 Texture removal

Texture removal is a direct application of image decomposition extensively used in image analysis, recognition and composition. To demonstrate the performance of our proposed model on textural removal, we compare it with seven popular methods. Among these competitors, RTV, LowRank and JCAS are the models of variational framework, RGF and SWF are filtering-based methods, and IUL and STD are two unsupervised methods based on DNNs. We use the test images which contain different kinds of textures in Fig. 4, and the comparison results of different methods are presented in Figs. 5 and 6.

In Fig. 5, we present both structural and textural components to visualize where textures are removed. The corresponding zoomed parts are shown with enlargements in red rectangles. One can observe that some structural edges are treated as textural edges removed in the textural components for most competitors. For example, the abdomen region of the man and the eye region of the horse are highlighted in Fig. 5. Visually, both parts belong to structures, but they are removed from structures leading to the smooth and blurred results in Fig. 5a–g. In Fig. 5h, our method removes the square patterns only and preserves these structural edges effectively.

Table 1 Average running time of different methods on 80 images of size 350×350

Methods	RTV [66]	JCAS [19]	IUL [15]	SWF [70]	STD [76]		Ours
					Standard INIT	pre-training INIT	
Time(s)	1.53	183.82	**0.012**	174.36	41.72	9.79	203.86



(a) RTV [66] (b) RGF [75] (c) LowRank [16] (d) JCAS [19]

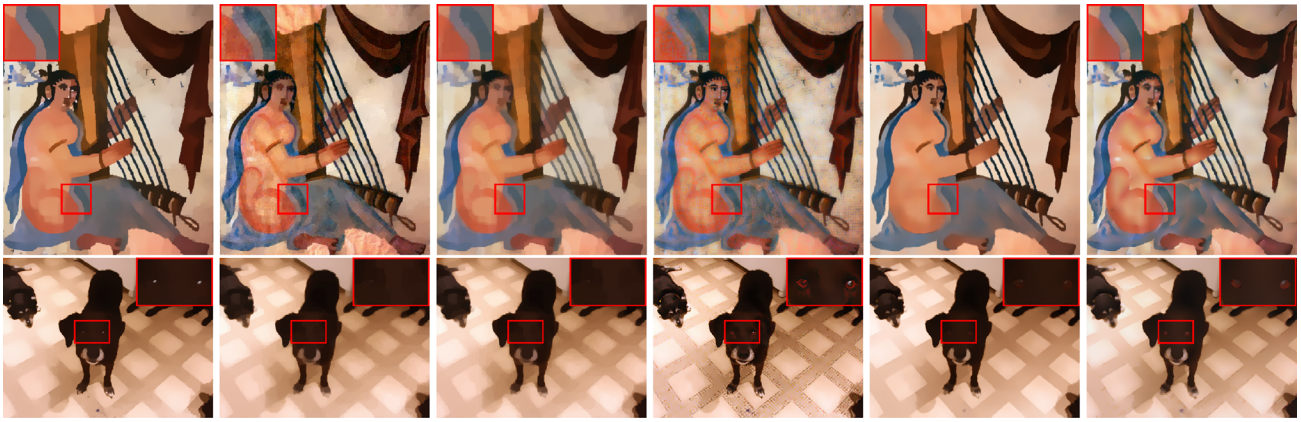
(e) IUL [15] (f) SWF [70] (g) STD [76] (h) Ours

Fig. 5 Texture removal results from different methods in Fig. 4a. For each method, the first row is the result of the texture removal and the second row is the result of the corresponding texture. The corresponding zoomed parts are shown with enlargements in red rectangles. Our method effectively preserves structural edges (first row), removes textural edges (second row) and does not suffer from staircasing artifacts

For other examples, we only exhibit the results of texture removal to save spaces in Fig. 6. From Fig. 6a, we can observe that RTV always fails to distinguish the small-scale structures and textures, and blurs the structural parts. The results of LowRank method always contain different scales of textures and blocking artifacts, which is insufficient to only extract complex textures with low rank theory, as illustrated in Fig. 6b. Suffering from the staircase effect of TV, JCAS and STD methods always remove the small-scale structures and achieve piecewise constant results, as shown in Fig. 6c, e. The drawback of SWF method is that some complex textures surrounding structural edges are always incorrectly preserved as shown in Fig. 6d. Our results are exhibited in Fig. 6f, from which we can observe that image textures can be clearly removed from image structures and structural edges are separated clearly and sharply.

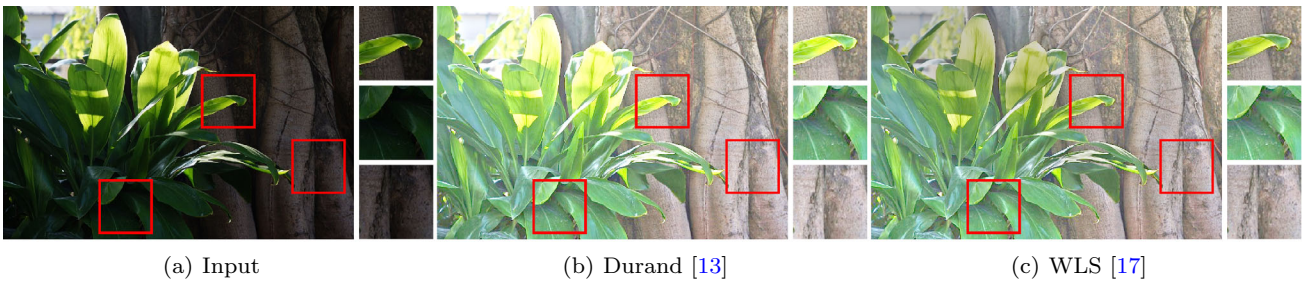
6.2 HDR tone mapping

Tone mapping aims to compress the intensity of a high dynamic range image, preserving the details and colors of the HDR image. To apply our method to HDR tone mapping problem, we first compute the luminance image using $\mathbf{l}^{in} = (0.299\mathbf{r}^{in} + 0.587\mathbf{g}^{in} + 0.114\mathbf{b}^{in})$; then, we apply our model to decompose the logarithm of luminance $\mathbf{l}^{in}$ into base and detail components. The output luminance $\mathbf{l}^{out}$ is obtained from the detail component plus the base compo-



(a) RTV [66] (b) LowRank [16] (c) JCAS [19] (d) SWF [70] (e) STD [76] (f) Ours

Fig. 6 More comparisons of the texture removal in Fig. 4b–c. The corresponding zoomed parts are shown with enlargements in red rectangles. Our method removes textural edges from image structures clearly and keeps much sharper structural edges



(a) Input (b) Durand [13] (c) WLS [17]

(d) JCAS [19] (e) STD [76] (f) Ours

Fig. 7 Comparison of several HDR tone-mapping methods. The corresponding zoomed parts are shown with enlargements in red rectangles. Our method achieves the best balance between detail-preserving and naturalness

nent compressed with a factor of 0.4. Finally, we adopt the color restoration [18] to reproduce chrominance information.

We compare our method with state-of-the-art tone mapping methods including filtering-based method [13,17], optimization-based method [19] and DNNs-based methods [76]. With a better image decomposition result, tone-mapped images can better preserve structural details and also be natural-looking. Visual examples of tone mapping results are provided in Fig. 7. As can be seen, for Durand and WLS, the results are over-enhanced and look unnatural, as shown in Fig. 7b, c. By contrast, the results from JCAS and STD do not suffer from such a problem, but they always lose structural details in relatively darker regions, as shown in Fig. 7d, e. In comparison with the competitors, the results from our method achieve the best balance between detail-preserving and naturalness, as shown in Fig. 7f. For example, the cracking bark and the leaves with white dots in red rectangles are recovered well. In addition, we use the tone-mapped image quality index (TMQI) [69] to compare different methods on a subject-rated image database of 15 HDR images quantitatively. From Table 2, one can see that our method achieves the highest TMQI values on most images (10 out of 15 images).

6.3 Detail enhancement

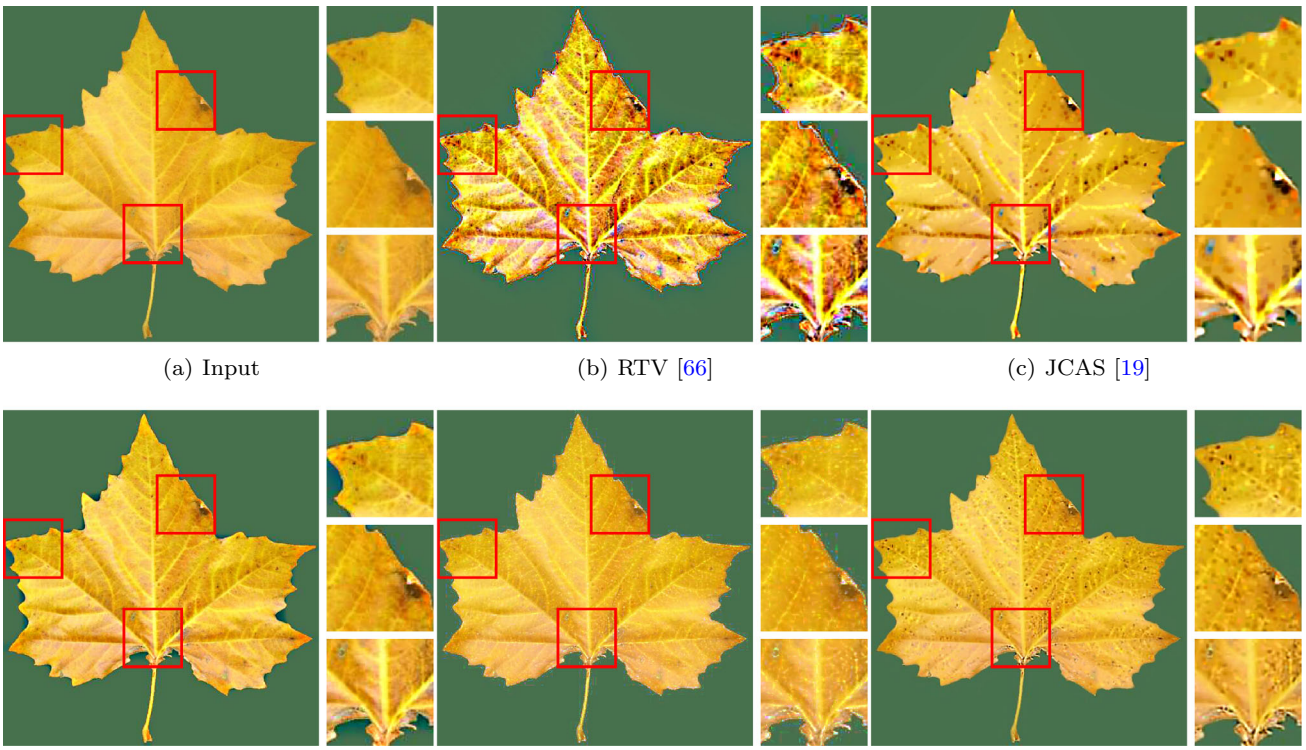
Given an image decomposition result, the detail enhancement can be achieved by superimposing the enhanced textural component on the structural component. Following the

Table 2 TMQI values of different methods on a subject-rated image database[69]

	DurandBF	WLS	JCAS	STD	Ours
1	0.8382	0.9107	0.9211	0.9517	**0.9529**
2	0.7733	0.8830	0.8997	0.9012	**0.9118**
3	0.8213	0.9357	0.9513	**0.9553**	0.9488
4	0.7890	0.9341	0.9786	0.9803	**0.9811**
5	0.8993	0.8495	**0.9717**	0.9548	0.9702
6	0.8092	0.8695	0.8758	**0.8961**	0.8873
7	0.8697	0.8567	0.8799	0.8691	**0.8834**
8	0.8003	0.8946	0.9169	0.9188	**0.9254**
9	0.8377	0.9073	0.9548	0.9818	**0.9835**
10	0.8526	0.8794	0.9287	**0.9516**	0.9409
11	0.8362	0.8588	0.9610	0.9810	**0.9827**
12	0.8217	0.8781	0.9701	0.9747	**0.9759**
13	0.8298	0.9086	0.9553	**0.9692**	0.9598
14	0.8171	0.8832	0.9177	0.9354	**0.9533**
15	0.8187	0.9305	0.9697	0.9586	**0.9614**

implementation of SWF [70] for detail enhancement, we obtained the enhanced image by $\boldsymbol{y}_e = \boldsymbol{u}+(\alpha+1)\boldsymbol{v}$, where α is an amplification parameter and is fixed to 5 in all experiments of this application. Here, the compared methods include RTV [66], JCAS [19], EGIF [40] and SWF [70]. If structural part and textural part are separated in a satisfactory way, edges of image details would be much sharper, while structural edges would be better preserved from being over-sharpened or overshot. On the contrary, if structural edges are extracted in textural part, they can be enhanced as the textural edges and halo artifacts will appear.

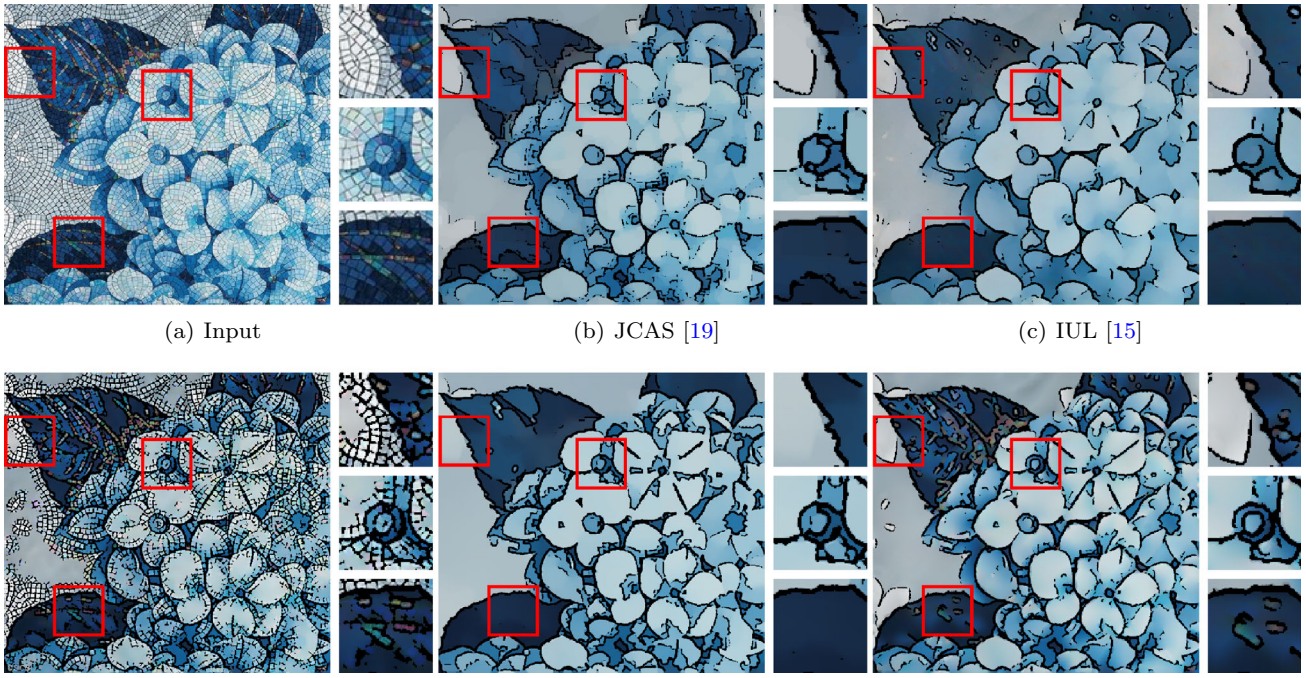
Visual examples of detail-enhanced experiments are provided in Fig. 8. As can be observed, for the results from RTV, JCAS and EGIF shown in Fig. 8b, c and d, respectively, structural edges of the leaves are over-enhanced as the structural edges are incorrectly separated in the textural component. Instead, SWF does not suffer from such a problem, but the detail edges of SWF in Fig. 8e are under-enhanced for the reason that the details needed to be enhanced are extracted in the structural component. In comparison with the competitors, our method can enhance the details in a better way while preserving the structural edges, as shown in Fig. 8f.



(a) Input (b) RTV [66] (c) JCAS [19]

(d) EGIF [40] (e) SWF [70] (f) Ours

Fig. 8 Results of detail enhancement from different methods. The corresponding zoomed parts are shown with enlargements in red rectangles. Our method can enhance the details in a better way while preserving the structural edges



(a) Input (b) JCAS [19] (c) IUL [15]

(d) SWF [70] (e) STD [76] (f) Ours

Fig. 9 Results of non-photorealistic abstraction from different methods. The corresponding zoomed parts are shown with enlargements in red rectangles. Our result fits non-photorealistic abstraction with simultaneous edge highlighting and texture suppressing

6.4 Non-photorealistic abstraction

Non-photorealistic abstraction aims to decrease the complexity of the scene while protecting important structures. Following the traditional methods [11,59], we perform the edge extraction with difference-of-Gaussian (DoG) on the structural component and highlight extracted edges to achieve non-photorealistic abstraction effect. If the structural edges are incorrectly removed in textural component, the structural component always is too over-smooth to detect in the abstraction results. Meanwhile, if the textural edges are incorrectly preserved in the structural component, redundant edge maps will appear in the abstraction results.

Figure 9 shows the visual examples of non-photorealistic abstraction results. It can be found that, for the methods of JCAS and IUL, as shown in Fig. 9b, c, respectively, the color and edges are over-smooth. For SWF, the textural edges retained clearly in the structural component lead to the mussy edges of abstraction results, as shown in Fig. 9d. For STD, the structural edges are more sharper than above, but still lose several structural details such as the white petals shown in Fig. 9e. However, our method removes the repetitive textures from the structural component but never smooth the structural edges, such as the colorful leaves shown in Fig. 9f. Furthermore, we recover the white petals and blue flower cores in a better way.

7 Conclusion

In this paper, we have presented an efficient structure–texture image decomposition model, which took advantages of both NTGV and CSC to separate the complex textures from richly detailed structural component. Meanwhile, the weight function has been incorporated to better distinguish structural and textural edges. We solved the model by an alternating minimization algorithm based on ADMM and gave a comprehensive analysis to the proposed algorithm. The experimental results have demonstrated the effectiveness of our approach on several applications including texture removal, HDR image tone mapping, detail enhancement and non-photorealistic abstraction.

In the future, we hope to increase the efficiency of our algorithm by GPU programming or DNNs. How to extend these techniques to decompose the digital images of wall paintings full of various diseases will also be a future work.

Funding This study was funded by the Youth Science and Technology Foundation of Gansu (20JR5RA050).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Aach, T., Mota, C., Stuke, I., Muhlich, M., Barth, E.: Analysis of superimposed oriented patterns. IEEE Trans. Image Process. **15**(12), 3690–3700 (2006)
2. Aujol, J.F., Gilboa, G., Chan, T., Osher, S.: Structure–texture image decomposition modeling, algorithms, and parameter selection. Int. J. Comput. Vis. **67**(1), 111–136 (2006)
3. Bao, L., Song, Y., Yang, Q., Yuan, H., Wang, G.: Tree filtering: efficient structure-preserving smoothing with a minimum spanning tree. IEEE Trans. Image Process. **23**(2), 555–569 (2013)
4. Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. SIAM J. Imaging Sci. **3**(3), 492–526 (2010)
5. Bristow, H., Eriksson, A., Lucey, S.: Fast convolutional sparse coding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 391–398 (2013)
6. Buades, A., Coll, B., Morel, J.M.: Nonlocal image and movie denoising. Int. J. Comput. Vision **76**(2), 123–139 (2008)
7. Chalasani, R., Principe, J.C., Ramakrishnan, N.: A fast proximal method for convolutional sparse coding. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN), pp. 1–5. IEEE (2013)
8. Chen, Q., Xu, J., Koltun, V.: Fast image processing with fully-convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2497–2506 (2017)
9. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Rev. **43**(1), 129–159 (2001)
10. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893. IEEE (2005)
11. DeCarlo, D., Santella, A.: Stylization and abstraction of photographs. ACM Trans. Graph. **21**(3), 769–776 (2002)
12. Dowson, N., Salvado, O.: Hashed nonlocal means for rapid image filtering. IEEE Trans. Pattern Anal. Mach. Intell. **33**(3), 485–499 (2010)
13. Durand, F., Dorsey, J.: Fast bilateral filtering for the display of high-dynamic-range images. ACM Trans. Graph. **21**(3), 257–266 (2002)
14. Fan, Q., Yang, J., Hua, G., Chen, B., Wipf, D.: A generic deep architecture for single image reflection removal and image smoothing. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3238–3247 (2017)
15. Fan, Q., Yang, J., Wipf, D., Chen, B., Tong, X.: Image smoothing via unsupervised learning. ACM Trans. Graph. (TOG) **37**(6), 1–14 (2018)
16. Fan, Y.R., Huang, T.Z., Ma, T.H., Zhao, X.L.: Cartoon-texture image decomposition via non-convex low-rank texture regularization. J. Franklin Inst. **354**(7), 3170–3187 (2017)
17. Farbman, Z., Fattal, R., Lischinski, D., Szeliski, R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. ACM Trans. Graph. (TOG) **27**(3), 1–10 (2008)
18. Fattal, R., Lischinski, D., Werman, M.: Gradient domain high dynamic range compression. In: Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques, pp. 249–256 (2002)
19. Gu, S., Meng, D., Zuo, W., Zhang, L.: Joint convolutional analysis and synthesis sparse representation for single image layer separation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1708–1716 (2017)
20. Gu, Y., Yang, X., Gao, Y.: A novel total generalized variation model for image dehazing. J. Math. Imaging Vis. **61**(9), 1329–1341 (2019)
21. Guo, X., Li, Y., Ma, J., Ling, H.: Mutually guided image filtering. IEEE Trans. Pattern Anal. Mach. Intell. **42**(3), 694–707 (2018)
22. He, K., Sun, J., Tang, X.: Guided image filtering. IEEE Trans. Pattern Anal. Mach. Intell. **35**(6), 1397–1409 (2013)
23. Jähne, B.: Spatio-Temporal Image Processing: Theory and Scientific Applications, vol. 751. Springer, Berlin (1993)
24. Jeon, J., Lee, H., Kang, H., Lee, S.: Scale-aware structure-preserving texture filtering. In: Proceedings of the Computer Graphics Forum, vol. 35, pp. 77–86. Wiley, New York (2016)
25. Kang, M., Kang, M., Jung, M.: Total generalized variation based denoising models for ultrasound images. J. Sci. Comput. **72**(1), 172–197 (2017)
26. Kass, M., Solomon, J.: Smoothed local histogram filters. In: ACM SIGGRAPH 2010 Papers, pp. 1–10 (2010)
27. Kavukcuoglu, K., Sermanet, P., Boureau, Y.L., Gregor, K., Mathieu, M., Cun, Y., et al.: Learning convolutional feature hierarchies for visual recognition. Adv. Neural. Inf. Process. Syst. **23**, 1090–1098 (2010)
28. Kim, Y., Ham, B., Do, M.N., Sohn, K.: Structure–texture image decomposition using deep variational priors. IEEE Trans. Image Process. **28**(6), 2692–2704 (2018)
29. Kou, F., Chen, W., Wen, C., Li, Z.: Gradient domain guided image filtering. IEEE Trans. Image Process. **24**(11), 4528–4539 (2015)
30. Le, T.M., Vese, L.A.: Image decomposition using total variation and div (bmo). Multiscale Model. Simul. **4**(2), 390–423 (2005)
31. Li, Z., Zheng, J., Zhu, Z., Yao, W., Wu, S.: Weighted guided image filtering. IEEE Trans. Image Process. **24**(1), 120–129 (2014)
32. Lieu, L.H., Vese, L.A.: Image restoration and decomposition via bounded total variation and negative Hilbert–Sobolev spaces. Appl. Math. Optim. **58**(2), 167 (2008)
33. Lindeberg, T.: Scale-space theory: a basic tool for analyzing structures at different scales. J. Appl. Stat. **21**(1–2), 225–270 (1994)
34. Lingling, J., Haiqing, Y., Xiangchu, F.: Adaptive variational models for image decomposition combining staircase reduction and texture extraction. J. Syst. Eng. Electron. **20**(2), 254–259 (2009)
35. Liu, X.: A new Tgv–Gabor model for cartoon-texture image decomposition. IEEE Signal Process. Lett. **25**(8), 1221–1225 (2018)
36. Liu, Z., Li, Y., Wang, W., Liu, L., Chen, R.: Mesh total generalized variation for denoising. IEEE Trans. Visual Comput. Graphics (2021). https://doi.org/10.1109/TVCG.2021.3088118
37. Lu, C., Song, G.: Image decomposition using adaptive regularization and div (bmo). J. Syst. Eng. Electron. **22**(2), 358–364 (2011)
38. Lu, C., Wang, M.: Alternating direction method for tgv-tgv* based cartoon-texture image decomposition. IET Image Proc. **10**(6), 495–504 (2016)
39. Lu, K., You, S., Barnes, N.: Deep texture and structure aware filtering network for image smoothing. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 217–233 (2018)
40. Lu, Z., Long, B., Li, K., Lu, F.: Effective guided image filtering for contrast enhancement. IEEE Signal Process. Lett. **25**(10), 1585–1589 (2018)
41. Mairal, J., Elad, M., Sapiro, G.: Sparse representation for color image restoration. IEEE Trans. Image Process. **17**(1), 53–69 (2007)
42. Meyer, Y.: Oscillating patterns in image processing and nonlinear evolution equations: the fifteenth Dean Jacqueline B. Lewis memorial lectures, vol. 22. American Mathematical Society (2001)
43. Ochs, P., Dosovitskiy, A., Brox, T., Pock, T.: An iterated l^1 algorithm for non-smooth non-convex optimization in computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1759–1766 (2013)
44. Ochs, P., Dosovitskiy, A., Brox, T., Pock, T.: On iteratively reweighted algorithms for nonsmooth nonconvex optimization in computer vision. SIAM J. Imaging Sci. **8**(1), 331–372 (2015)
45. Ono, S.: l^0 gradient projection. IEEE Trans. Image Process. **26**(4), 1554–1564 (2017)

46. Osher, S., Solé, A., Vese, L.: Image decomposition and restoration using total variation minimization and the h^{-1} norm. Multiscale Model. Simul. **1**(3), 349–370 (2003)
47. Paris, S., Hasinoff, S.W., Kautz, J.: Local Laplacian filters: edge-aware image processing with a Laplacian pyramid. Commun. ACM **58**(3), 81–91 (2015)
48. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990)
49. Rubinstein, R., Member, S., IEEE, Bruckstein, A.M., Member,: Dictionaries for sparse representation modeling. Proc. IEEE **98**(6), 1045–1057 (2010)
50. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**(1–4), 259–268 (1992)
51. Shu, L., Du, H.: Side window weighted median image filtering. In: Proceedings of the International Conference on Multimedia Systems and Signal Processing, pp. 26–30. Association for Computing Machinery (2020)
52. Starck, J.L., Elad, M., Donoho, D.L.: Image decomposition via the combination of sparse representations and a variational approach. IEEE Trans. Image Process. **14**(10), 1570–1582 (2005)
53. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. **73**, 273–282 (1996). https://doi.org/10.1111/j.1467-9868.2011.00771.x
54. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of the International Conference on Computer Vision, pp. 839–846. IEEE (1998)
55. Valkonen, T., Bredies, K., Knoll, F.: Total generalized variation in diffusion tensor imaging. SIAM J. Imaging Sci. **6**(1), 487–525 (2013)
56. Vese, L., Chan, T.F.: Reduced Non-convex Functional Approximations for Image Restoration and Segmentation. University of California, Department of Mathematics, Los Angeles (1997)
57. Vese, L., Osher, S.: Modeling textures with total variation minimization and oscillating patterns in image processing. J. Sci. Comput. **19**(1–3), 553–572 (2003)
58. Wang, C., Zhang, H., Liu, L.: Total generalized variation-based retinex image decomposition. Vis. Comput. **37**(1), 77–93 (2021)
59. Winnemöller, H., Olsen, S.C., Gooch, B.: Real-time video abstraction. ACM Trans. Graph. **25**(3), 1221–1226 (2006)
60. Wohlberg, B.: Efficient convolutional sparse coding. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7173–7177. IEEE (2014)
61. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **31**(2), 210–227 (2008)
62. Xu, J., Feng, X., Hao, Y., Han, Y.: Image decomposition and staircase effect reduction based on total generalized variation. J. Syst. Eng. Electron. **25**(1), 168–174 (2014)
63. Xu, J., Feng, X., Hao, Y., Han, Y.: Image decomposition using adaptive second-order total generalized variation. SIViP **8**(1), 39–47 (2014)
64. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via l^0 gradient minimization. In: Proceedings of the SIGGRAPH Asia Conference, pp. 1–12 (2011)
65. Xu, L., Ren, J., Yan, Q., Liao, R., Jia, J.: Deep edge-aware filters. In: Proceedings of the International Conference on Machine Learning, vol. 37, pp. 1669–1678. PMLR (2015)
66. Xu, L., Yan, Q., Xia, Y., Jia, J.: Structure extraction from texture via relative total variation. ACM Trans. Graph. (TOG) **31**(6), 1–10 (2012)
67. Xu, P., Wang, W.: Structure-aware window optimization for texture filtering. IEEE Trans. Image Process. **28**(9), 4354–4363 (2019)
68. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1794–1801. IEEE (2009)
69. Yeganeh, H., Wang, Z.: Objective quality assessment of tone-mapped images. IEEE Trans. Image Process. **22**(2), 657–667 (2012)
70. Yin, H., Gong, Y., Qiu, G.: Side window filtering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8758–8766 (2019)
71. Yin, H., Gong, Y., Qiu, G.: Combined window filtering and its applications. Multidimens. Syst. Signal Process. **32**(1), 313–333 (2021)
72. Yin, W., Goldfarb, D., Osher, S.: Image cartoon-texture decomposition and feature selection using the total variation regularized l^1 functional. In: Proceedings of the International Workshop on Variational, Geometric, and Level Set Methods in Computer Vision, pp. 73–84. Springer (2005)
73. Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2528–2535. IEEE (2010)
74. Zhang, H., Patel, V.M.: Convolutional sparse coding-based image decomposition. In: Proceedings of the British Machine Vision Conference (BMVC) (2016)
75. Zhang, Q., Shen, X., Xu, L., Jia, J.: Rolling guidance filter. In: Proceedings of the European Conference on Computer Vision, pp. 815–830. Springer (2014)
76. Zhou, F., Chen, Q., Liu, B., Qiu, G.: Structure and texture-aware image decomposition via training a neural network. IEEE Trans. Image Process. **29**, 3458–3473 (2019)
77. Zhou, Z., Wang, B., Ma, J.: Scale-aware edge-preserving image filtering via iterative global optimization. IEEE Trans. Multimed. **20**(6), 1392–1405 (2017)

Chunxue Wang is an associate researcher at the Cultural Heritage Digitization Institute, Dunhuang Academy, Dunhuang, China. She received her Ph.D. degree (2016) in computational mathematics from the University of Science and Technology of China, China. Her research interests include mathematical image processing, geometry modeling and processing, inverse problems and optimization.

Linlin Xu is a lecture at the School of Computer Information Management, Inner Mongolia University of Finance and Economics, Inner Mongolia, China. She received her Ph.D. (2018) in computational mathematics from the University of science and technology of China. Her research interests include sparse optimization, geometry modeling and processing.

Ligang Liu is a professor at the School of Mathematical Sciences, University of Science and Technology of China. He received his B.Sc. (1996) and his Ph.D. (2001) from Zhejiang University, China. Between 2001 and 2004, he worked at Microsoft Research Asia. Then he worked at Zhejiang University during 2004 and 2012. He paid an academic visit to Harvard University during 2009 and 2011. His research interests include digital geometric processing, computer graphics, and image processing. He serves as the associated editors for journals of IEEE TVCG, IEEE CG&A, CGF, CAGD, C&G, and The Visual Computer. He served as the conference co-chair of GMP 2017 and the program co-chairs of SIAM GD 2019, GMP 2018, CAD/Graphics 2017, CVM 2016, SGP 2015, and SPM 2014. His research works could be found at his research website: http://sta.ustc.edu.cn/lgliu.