

# Structure-Texture Image Decomposition Using Discriminative Patch Recurrence

Ruotao Xu, Yong Xu, Yuhui Quan\*

**Abstract**—Morphology component analysis provides an effective framework for structure-texture image decomposition, which characterizes the structure and texture components by sparsifying them with certain transforms respectively. Due to the complexity and randomness of texture, it is challenging to design effective sparsifying transforms for texture components. This paper aims at exploiting the recurrence of texture patterns, one important property of texture, to develop a nonlocal transform for texture component sparsification. Since the plain patch recurrence holds for both cartoon contours and texture regions, the nonlocal sparsifying transform constructed based on such patch recurrence sparsifies both the structure and texture components well. As a result, cartoon contours could be wrongly assigned to the texture component, yielding ambiguity in decomposition. To address this issue, we introduce a discriminative prior on patch recurrence, that the spatial arrangement of recurrent patches in texture regions exhibits isotropic structure which differs from that of cartoon contours. Based on the prior, a nonlocal transform is constructed which only sparsifies texture regions well. Incorporating the constructed transform into morphology component analysis, we propose an effective approach for structure-texture decomposition. Extensive experiments have demonstrated the superior performance of our approach over existing ones.

**Index Terms**—Image Decomposition, Structure-Texture Separation, Patch Recurrence, Sparse representation

## I. INTRODUCTION

An image is usually the superposition of a structure component (layer) and a texture component (layer). The structure component, also called cartoon component, refers to the piecewise-constant geometrical parts of an image, including homogeneous regions, contours, and sharp edges. In contrast, the texture component is about the oscillating patterns of an image, such as fine structures and local repeating features. Structure-texture image decomposition is to separate

Ruotao Xu is with School of Computer Science and Engineering at South China University of Technology, China. (email: xu.ruotao@mail.scut.edu.cn)

Yong Xu is with School of Computer Science and Engineering at South China University of Technology, China, as well as with Peng Cheng Laboratory, Shenzhen, China. (email: yxu@scut.edu.cn)

Yuhui Quan is with School of Computer Science and Engineering at South China University of Technology, China, as well as with Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, China. (email: csyhquan@scut.edu.cn)

Corresponding author: Yuhui Quan.

This work was supported in part by National Nature Science Foundation of China under Grants 61872151, 62072188 and U1611461, in part by Science and Technology Planning Project of Guangdong Province under Grant 2019A050510010, in part by Natural Science Foundation of Guangdong Province under Grant 2020A1515011128, and in part by Science and Technology Program of Guangzhou under Grant 201802010055.

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org/>, provided by the author. The material includes a document containing additional results and details of the work. Contact csyhquan@scut.edu.cn for further questions about this work.

the structure and texture components from an image, which plays an important role in computer vision, with a wide range of applications to image restoration [1]–[3], motion analysis [4], image segmentation [5], image compression [6], stereo matching [7], image editing [8]–[10], color transfer [8], pattern recognition [11], biomedical engineering [12], remote sensing [13]–[15], etc.

The importance of structure-texture decomposition originates from the fact that the structure and texture components exhibit significantly distinct characteristics and they often involve different operations in the processing, analysis and recognition of images. For instance, the texture component is preferred in optical flow estimation as it is often free of shading reflections and shadows [4], while the structure component is often extracted for enhancing the stability of depth estimation [7]. In many image processing tasks (*e.g.* [1]), the cartoon regions and texture regions need separate treatments to guarantee the visual quality of results. Another example is object recognition, in which contour cues and texture features are extracted by different approaches (*e.g.* [16]).

In general, structure-texture decomposition requires solving the following underdetermined problem:

$$\mathbf{f} = \mathbf{u} + \mathbf{v}, \quad (1)$$

where  $\mathbf{f}$  is the given image,  $\mathbf{u}$  the structure layer, and  $\mathbf{v}$  the texture layer. Since the unknowns are much more than the equations, priors on both the structure and texture layers are needed to solve the problem. In existing approaches, the sparsity prior of spatial gradients (*e.g.* [17]–[21]) is often imposed on structure layers. The sparsity prior is also used for modeling texture layers [19], [20] which assumes texture layers can be sparsified under some system (transform or dictionary). One effective approach to utilizing the sparsity priors of image components for image decomposition is the Morphology Component Analysis (MCA) [22], which is about solving the following model:

$$\min_{\mathbf{u}, \mathbf{v}} \|\mathbf{S}_1 \mathbf{u}\|_1 + \|\mathbf{S}_2 \mathbf{v}\|_1, \text{ s.t. } \mathbf{u} + \mathbf{v} = \mathbf{f}, \quad (2)$$

where  $\mathbf{S}_1, \mathbf{S}_2$  are two transforms that sparsify the structure and texture layers respectively in terms of  $\ell_1$  norm.

## A. Motivation and Main Idea

The existing approaches often use a local transform, *i.e.* a transform with localized supports, for sparsifying texture components, *e.g.* local discrete cosine transform (DCT). The local transforms ignore one important property of texture called self-recurrence, *i.e.*, a texture region contains a number of similar

elements (patterns). Such a property can be cast into the patch recurrence in the texture layer. Indeed, patch recurrence is one prominent natural image prior used in the so-called nonlocal image recovery approaches (*e.g.* [23], [24]). Many nonlocal approaches use a nonlocal transform (*i.e.* a transform with non-localized support) that encodes the patch recurrence of a given image for processing. There are also benefits of using patch recurrence and nonlocal transforms for modelling texture layers. For instance, the patches of weakly-periodic or non-periodic texture can be well sparsified by the collaboration of their similar counterparts. This inspired us to investigate the exploitation of patch-recurrence-based nonlocal sparsifying transforms for modeling texture layers.

However, the plain patch recurrence prior used in existing image recovery methods may also hold on cartoon regions. For instance, similar patches can also be found along a straight contour edge. As a result, the nonlocal transforms used in existing image recovery methods can sparsify both texture and structure layers well. In this case, both  $\|S_1\mathbf{u}\|_1$  and  $\|S_1\mathbf{v}\|_1$  are small, and  $\|S_1(\mathbf{u} + \mathbf{v})\|_1 \leq \|S_1\mathbf{u}\|_1 + \|S_1\mathbf{v}\|_1$  is likely to be small as well. In other words, the recurrent patches (*e.g.* edge patches) in the structure layer may be wrongly assigned to the texture layer. This ambiguity is harmful to the decomposition accuracy. To effectively exploit the self-recurrence of texture for decomposition, we introduce a discriminative prior on patch recurrence. The prior comes from the observation that, the neighboring similar patches around cartoon contours tend to align along a major direction, while the ones in texture regions are likely to scatter around. See Fig. 1 for an illustration, where the neighboring similar patches in cartoon regions exhibit different spatial configurations from those in texture regions. Such a discriminative prior allows exploiting the patch recurrence of texture while avoiding the ambiguity of recurrent patches between the two components.

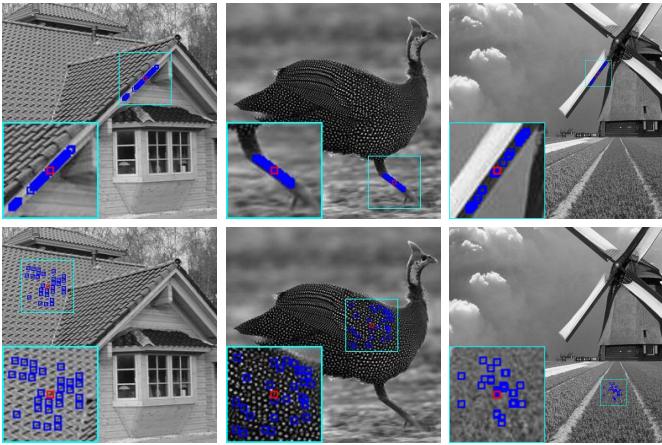


Fig. 1. Anisotropic patch recurrence in structure (upper row) and isotropic patch recurrence in texture (bottom row). The red squares indicate the source patches while the blue ones indicate the matched patches of the target patches.

Intuitively, the uni-directional distribution of neighboring similar patches around cartoon contours is attributed to the fact that the often-seen cartoon contours contain a large portion of straight edges. The patches along a straight edge have strong similarity to each other, and the patches on different sides

of the edge are often dissimilar. In contrast, the isotropic distribution of neighboring similar texture patches is due to the spatial homogeneity of textures, *e.g.* cobwebbing in fabrics, scales on snakes, and furs of lions. For convenience, the prior that neighboring similar texture patches scatter around, is referred to as the isotropic patch recurrence for texture components, while that similar patches of cartoon contour distribute along a major direction, is referred to as the anisotropic patch recurrence prior for structure components. Such a pair of priors composes of our discriminative patch recurrence prior. The isotropic patch recurrence can characterize texture components well and meanwhile distinguish itself from the anisotropic patch recurrence of structure components.

With the discriminative patch recurrence prior, we construct an isotropic nonlocal transform based on directional patch matching. The constructed transform can sparsify texture components well while not good at generating sparse representations on structure components. Incorporating such a nonlocal transform into MCA, we develop a patch-recurrence-based approach for structure-texture decomposition. Benefiting from exploitation of self-recurrence of texture, our approach shows improvement over the existing ones in both qualitative and quantitative evaluations.

### B. Contributions

The contributions of this paper are three-fold. Firstly, we introduce the discriminative patch recurrence prior for exploiting the self-recurrence of texture for structure-texture decomposition. The traditional patch recurrence prior designed for image recovery only exploits the repeating property of image patches, without any consideration on the discrimination between structure components and texture components. In contrast, the discriminative prior additionally considers the spatial distribution of recurrent patches, *i.e.*, how the image patches repeat over space. Therefore, it can distinguish texture components from structure components well, and allows constructing a patch-recurrence-based decomposition model without confusion on the recurrent patches on the two layers.

Secondly, we employ a directional patch matching scheme, built upon which an isotropic nonlocal sparsifying transform is proposed to exploit the discriminative patch recurrence prior for modeling texture components. Note that the discriminative patch recurrence prior, though promising, is non-trivial to be utilized formally in a decomposition model. Our directional patch matching scheme groups similar patch along different directions separately, by which the isotropy property of matching results is enforced. Then, the constructed nonlocal transform can encode the isotropic patch recurrence artfully.

Thirdly, based on the isotropic nonlocal sparsifying transform, a patch-recurrence-based approach to structure-texture decomposition is proposed with several advantages over existing ones. Compared to [25]–[27] which utilize the low rank prior on texture patches for modeling regular patterns, ours uses the isotropy prior of patch recurrence for better modeling non-regular texture. Compared with the patch-recurrence-based approach [28], the proposed one has additional considerations on the spatial configuration of patch recurrence. By subdividing the patch recurrence into the isotropic and anisotropic

versions for texture and structure respectively, our approach improves the discrimination between cartoon contours and texture patterns. In comparison to the sparse-coding-based approaches [20], [29] that use local sparsifying systems, ours can be viewed as employing a nonlocal transform to exploit similar patches for better sparse representation of textures.

### C. Notations and Organization

Throughout this paper, unless specified, bold upper letters are used for matrices, bold lower letters for column vectors, light lower letters for scalars, and hollow letters for sets. The notations  $\mathbf{0}$  and  $\mathbf{I}$  denote the zero matrix and the identity matrix with appropriate sizes respectively, and  $\text{diag}(\mathbf{x})$  denotes the square diagonal matrix with vector  $\mathbf{x}$  as its main diagonal. The  $\ell_p$  norm of a vector is denoted by  $\|\cdot\|_p$ . Given a sequence  $\{\mathbf{y}^{(t)}\}_{t \in \mathbb{N}}$ ,  $\mathbf{y}^{(t_0)}$  denotes the  $t_0$ -th element in the sequence. For a vector  $\mathbf{x} \in \mathbb{R}^N$  and a set  $\mathbb{S} \subset \{1, \dots, N\}$ ,  $\mathbf{x}(i)$  denotes the  $i$ -th element in  $\mathbf{x}$ , and  $\mathbf{x}_{\mathbb{S}}$  denotes the vector concatenating all  $\mathbf{x}(i)$  where  $i \in \mathbb{S}$ . For a matrix  $\mathbf{X}$ ,  $\mathbf{X}(i, j)$  denotes the element of  $\mathbf{X}$  at the  $i$ -th row and  $j$ -th column. For matrix concatenation, semicolons are for adding elements row-wisely and commas are for adding elements column-wisely.

The rest of this paper is organized as follows. Section II is for literature review. Section III is on the construction of our nonlocal sparsifying transform for texture components. Section IV is devoted to developing our method based on the constructed sparsifying transform. Section V is on experimental evaluation. Finally Section VI concludes the paper.

## II. RELATED WORK

There have been many approaches proposed for structure-texture image decomposition. Their main differences lie in the mechanisms of modeling cartoon regions and texture patterns. Regarding cartoon characterization, a majority of approaches use the total variation (TV) penalties (e.g. [1], [13], [14], [17]–[19], [30]–[35]), since TV penalties can efficiently induce piece-wise constant signals with bounded variations (BV). In [34], the windowed relative TV is used for improvement. See also [8] for a generalized TV penalty. Instead of using TV, some approaches impose sparse representation on cartoon parts under some systems such as wavelet frames [16], [36], [37] and curvelet transforms [22], [38], which can effectively characterize the smoothness of structure components. TV-based approaches have deep connections to the sparse-coding-based ones using spline wavelets [39]. Compared to the multi-scale extension of TV-based approach [36], wavelet systems naturally allow multi-resolution analysis on cartoon structures. Recently, inspired from the classical TV minimization, Kim *et al.* [40] introduce a deep variational prior for cartoon parts by employing a pre-trained deep denoising neural network for the estimation of structure components.

In comparison to structure components, texture components are much more challenging to characterize due to their high complexities and large variations. In the past, many studies modeled texture by the patterns with strong oscillations, and they focused on exploiting the spaces of oscillatory functions which are discriminative to the BV space, as well as exploring

the TV-antagonistic norms, *i.e.* their induced functionals have opposite behaviors to the TV-induced ones. A seminal work can be traced back to Meyer's proposal [30], whose numerical implementation is first given in [31]. Following this line of research, many studies have been conducted on addressing numerical issues; see *e.g.* [13], [14], [17], [33], [41]–[43]. There are also other studies developing new models such as linear filtering models [44], [45] and multi-scale model [46], or focusing on applications [1], [3], [6]. Since noises also exhibit oscillatory patterns, the aforementioned approaches cannot well handle noisy images. Furthermore, as pointed out in [25], these approaches may fail in discovering small-magnitude but well-patterned textures.

One promising alternative is the sparse-representation-based approach (*e.g.* [5], [16], [19], [20], [29], [37], [38]), which assumes textures can be sparsified under certain systems, *i.e.*, a texture patch can be represented by few of atoms in a certain dictionary (synthesis model) or sparsified under some transform (analysis model). With a proper system, either analytic (*e.g.* local Fourier frame [38]) or learned (*e.g.* [19], [20]), the texture component can be effectively extracted by promoting the sparsity of representation coefficients. In [29], analysis and synthesis models are combined for further improvement. Since cartoon features such as edges and contours may also have sparse representations under proper dictionaries, the sparse-co-based approaches do not work well when the sparsifying system for cartoon has high coherence with that for texture. In other words, one key in the sparse-representation-based approaches is designing effective sparsifying systems whose ambiguity between structure and texture is minimized. Interested readers can refer to [16].

Another line of research on texture modeling is built upon the low-rankness prior of texture patches. An early inspiring attempt is given by Schaeffer and Osher [25]. They assumed the texture patches are almost linearly dependent after certain alignment and applied the low-rank regularization with nuclear norm to aligned texture patches. In [27], the *logdet* function is used to replace the nuclear norm in [25] for a better approximation to the rank. Though capable of processing regular textures with global homogeneity, these methods cannot work well on the textures with distortion or with spatially-varying patterns. As a result, such methods may produce undesirable artifacts in the results. To address this issue, Ono and Miyata [26] adapted the low rank prior from the whole image to a local window, and the low-rank regularization is applied to the patches of texture component within a local block. Moreover, multiple shear operators are employed for better alignment. Nevertheless, this approach may suffer from the under-sampling problem, *i.e.*, the local block may not contain sufficient similar patterned patches for stable low-rank approximation, especially for the patch locating near the boundary of a texture region. Besides, the shear operations may involve additional computational cost.

Instead of sampling patches within a local block, Ma *et al.* [28] proposed to group similar patches and apply low-rank regularization to each patch group for extracting textures. The structure component is extracted by enforcing structured sparsity on each group. It is noted that, though with low-rank

approximation, this method employs a different prior from the low rank prior in [25]–[27]: it can be viewed as the nonlocal patch recurrence prior which has been successfully applied to many image restoration tasks [23]. The nonlocal patch recurrence is also exploited by Sur *et al.* [47]. They extracted the texture component using a statistical model built upon the power spectrum of similar texture patches. However, the direct use of the nonlocal patch recurrence prior in structure-texture decomposition can cause ambiguity, as the prior is applicable to both the structure component and texture component. In this paper, we aim at further exploiting the nonlocal patch recurrence for structure-texture decomposition.

There is a close-related topic called structure-preserving filtering [48]–[51]. These methods construct guidance from images and use it to smooth the images. Compared to structure-texture decomposition, such methods do not explicitly model the characteristics of texture or cartoon. In addition, these methods are often designed to allow user interaction for parameter tune-up on different images.

### III. NONLOCAL SPARSIFYING TRANSFORM FOR TEXTURE

In this section, we construct a nonlocal sparsifying transform for texture layers based on the isotropic patch recurrence in texture components, which is built upon a directional patch matching scheme.

#### A. Directional Patch Matching

Existing nonlocal approaches for image recovery exploit patch recurrence by running patch matching that collects the top- $K$  similar patches of each source patch within a neighborhood. Unlike these methods, we use a directional patch matching to find top- $K$  similar patches along different directions respectively, by which the isotropic distribution of similar patches can be exploited.

We first define the banded regions for directional matching. Given a source patch  $\mathbf{p}$  centered at  $c$ , its neighborhood  $\mathbb{N}_\mathbf{p}$  is defined as an  $S \times S$  window  $\mathbb{M}_\mathbf{p}$  centered at  $c$ . The neighborhood  $\mathbb{N}_\mathbf{p}$  is partitioned into  $D + 1$  regions, which includes (i)  $D$  bi-directional bands  $\{\mathbb{N}_\mathbf{p}^{(d)}\}_{d=1}^D$  with the width of  $B$  pixels along different directions and (ii) one central region  $\mathbb{N}_\mathbf{p}^{(0)}$ . The partition scheme is done as follows. In  $\mathbb{N}_\mathbf{p}$ , we find  $D$  banded regions  $\{\mathbb{M}_\mathbf{p}^{(d)} \subset \mathbb{N}_\mathbf{p}\}_{d=1}^D$  which go through  $\mathbb{M}_\mathbf{p}$  with center at  $c$  and are aligned at  $0^\circ, \frac{180^\circ}{D}, \dots, (D-1)\frac{180^\circ}{D}$  respectively. Let  $\bar{\mathbb{M}}_\mathbf{p}^{(0)} = \bigcap_{d=1}^D \mathbb{M}_\mathbf{p}^{(d)}$  denote the overlap of  $\mathbb{M}_\mathbf{p}^{(1)}, \dots, \mathbb{M}_\mathbf{p}^{(D)}$ . The bands  $\{\mathbb{N}_\mathbf{p}^{(d)}\}_{d=1}^D$  are defined as follows:

$$\mathbb{N}_\mathbf{p}^{(d)} = \mathbb{M}_\mathbf{p}^{(d)} \cap \bar{\mathbb{M}}_\mathbf{p}^{(0)}, d = 1, \dots, D, \quad (3)$$

where  $\bar{\mathbb{M}}_\mathbf{p}^{(0)}$  is the complement of  $\mathbb{M}_\mathbf{p}^{(0)}$ . The patches whose center pixels fall into the banded regions  $\mathbb{N}_\mathbf{p}^{(d)}, d = 1, \dots, D$  are considered as the candidates for searching. Note that while the excluded central region  $\mathbb{M}_\mathbf{p}^{(0)}$  is likely to correspond to the most similar patches to  $\mathbf{p}$  in  $\mathbb{M}_\mathbf{p}$ , we empirically found that an adequate number of sufficiently-similar patches can still be found in outside this central region.

In principle, a proper width  $B$  should make the banded regions  $\{\mathbb{N}_\mathbf{p}^{(d)}\}_d$  sufficiently large while keeping the overlapped

region  $\mathbb{M}_\mathbf{p}^{(0)}$  small. In implementation, we set  $D = 4, B = 8$  so that the banded regions occupy more than 50% of the whole search region  $\mathbb{N}_\mathbf{p}$  and at the same time the overlapped region occupies less than 10%. See Fig. 2(a) for the partition used in our implementation with  $D = 4, B = 8$ .

The directional patch matching for  $\mathbf{p}$  conducts the searching process on each banded region separately. For an image with  $N$  pixels, define  $\mathbf{p}_i$  as the patch centered at the  $i$ -th pixel for  $i = 1, \dots, N$ . Symmetric image boundary extension is applied for ensuring that  $N$  patches are sampled. For each patch  $\mathbf{p}_i$ , we search its top- $K$  similar patches based on  $\mathbb{N}_{\mathbf{p}_i}^{(1)}, \dots, \mathbb{N}_{\mathbf{p}_i}^{(D)}$  respectively. The indices of the similar patches are collected as  $\mathbb{S}_i^{(d)}$  for all  $d, i$ . In other words,  $\mathbb{S}_i^{(d)}$  stores the indices of the similar patches of the  $i$ -th patch along the  $d$ -th direction. The similarity of two patches  $\mathbf{p}, \mathbf{q}$  is measured by

$$\omega(\mathbf{p}, \mathbf{q}) = e^{-\|\mathbf{p}-\mathbf{q}\|_2^2/h}, \quad (4)$$

where  $h$  is a scalar set to 0.3 in practice.

See Fig. 2(b)-(c) for an illustration of the directional patch matching results on a texture regions and a cartoon contour respectively. In both case, the matched patches are scattered with the same number in different directions. The matched patches along all directions are very similar to the source patch in the texture region. In contrast, similar patches are found along one direction around the cartoon contour. The anisotropic distribution of similar contour patches is explained as follows. Consider an edge lying in the direction of some predefined band, its similar patches can only be found in one band but fail in the others. For an edge lying in the region of between two predefined bands, its similar patches cannot be found in any band. In addition, for a highly-bended curve, few similar patches can be found in all the bands. In short, for edges in different shapes and conditions, dissimilar patches can be found in most directions.

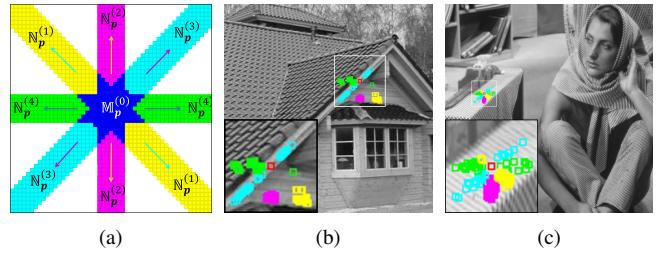


Fig. 2. Directional patch matching with  $D = 4$  and  $B = 8$ . (a) Schematic diagram of the partition strategy. (b)-(c) Matching result of a source patch on a cartoon contour and a texture region respectively. The red boxes indicate the source patches and the boxes in other colors indicate the matched patches along different directions.

#### B. Isotropic Nonlocal Sparsifying Transform

Given an image  $\mathbf{x} \in \mathbb{R}^N$ , the directional patch matching constructs  $D$  index sets:  $\mathbb{S}_i^{(d)}, d = 1, \dots, D, i = 1, \dots, N$ . Let  $\tilde{\mathbf{L}}^{(d)} \in \mathbb{R}^{N \times N}$  denote the directional nonlocal transform of  $\mathbf{x}$  for the  $d$ -th direction, which is defined by

$$\tilde{\mathbf{L}}^{(d)}(i, j) = \begin{cases} 1, & i = j; \\ -\frac{\omega(\mathbf{p}_i, \mathbf{p}_j)}{\sum_{k \in \mathbb{S}_i^{(d)}} \omega(\mathbf{p}_i, \mathbf{p}_k)}, & j \in \mathbb{S}_i^{(d)}; \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

The transform  $\tilde{\mathbf{L}}^{(d)}$  can be rewritten as  $\tilde{\mathbf{L}}^{(d)} = \mathbf{I} - \mathbf{E}^{(d)}$  where  $\mathbf{E}^{(d)}(i, j) = \omega(\mathbf{p}_i, \mathbf{p}_j)/\sum_{k \in \mathbb{S}_i^{(d)}} \omega(\mathbf{p}_i, \mathbf{p}_k)$  if  $j \in \mathbb{S}_i^{(d)}$  and 0 otherwise. With this form, it is straightforward to show

$$(\tilde{\mathbf{L}}^{(d)} \mathbf{x})(i) = \mathbf{x}(i) - \sum_{j \in \mathbb{S}_i^{(d)}} \frac{\omega(\mathbf{p}_i, \mathbf{p}_j)}{\sum_{k \in \mathbb{S}_i^{(d)}} \omega(\mathbf{p}_i, \mathbf{p}_k)} \mathbf{x}(j). \quad (6)$$

It can be seen that  $\tilde{\mathbf{L}}^{(d)}$  is a nonlocal difference operator which relates each  $\mathbf{x}_i$  to the weighted sum of its nonlocal similar ones lying at the  $d$ -th direction. We can also view  $\mathbf{E}^{(d)}$  as a directional version of the well-known nonlocal mean (NLM) [52], which calculates the NLM in the  $d$ -th banded region. Recall that the NLM is constructed based on the similar patches in an isotropic neighborhood. In contrast, only similar patches lying in the a banded region are used for constructing  $\mathbf{E}^{(d)}$ . In the case of perfect patch recurrence along the  $d$ -th direction, *i.e.*  $\mathbf{p}_j = \mathbf{p}_i$  for all  $j \in \mathbb{S}_i^{(d)}$  and all  $i$ , we have  $\omega(\mathbf{p}_i, \mathbf{p}_j) = 1$  and  $\mathbf{x}(i) = \mathbf{x}(j)$  for all  $j \in \mathbb{S}_i^{(d)}$ . Then, it is easy to show that  $(\tilde{\mathbf{L}}^{(d)} \mathbf{x})(i) = 0$  for all  $i$ , implying that  $\tilde{\mathbf{L}}^{(d)}$  can sparsify  $\mathbf{x}$  perfectly. When the patch recurrence of texture along the  $d$ -th direction is strong but imperfect, it is reasonable to assume  $(\tilde{\mathbf{L}}^{(d)} \mathbf{x})(i) \approx 0$  for most  $i$ , and thus  $\tilde{\mathbf{L}}^{(d)}$  can still sparsify  $\mathbf{x}$  well. In conclusion, the directional nonlocal transform  $\tilde{\mathbf{L}}^{(d)}$  can well sparsify an image with strong patch recurrence along the  $d$ -th direction.

The nonlocal sparsifying transform we construct has the following form:

$$\mathbf{L} = [\tilde{\mathbf{L}}^{(1)}; \dots; \tilde{\mathbf{L}}^{(D)}] \in \mathbb{R}^{ND \times N}. \quad (7)$$

In other words,  $\mathbf{L}$  stacks all directional nonlocal transforms. Thus,  $\mathbf{L}$  is referred to as an isotropic nonlocal sparsifying transform. Since  $\mathbf{L}\mathbf{x} = [\tilde{\mathbf{L}}^{(1)}\mathbf{x}; \dots; \tilde{\mathbf{L}}^{(D)}\mathbf{x}]$ , applying  $\mathbf{L}$  is about calculating the nonlocal difference over the input image along different directions separately. If the isotropic patch recurrence of a region is strong, all  $\tilde{\mathbf{L}}^{(d)}$ 's can sparsify the region and thus  $\mathbf{L}$  inheriting the sparsifying capability from  $\tilde{\mathbf{L}}^{(d)}$ 's will generate sparse results. When the patch recurrence is anisotropic, some of  $\tilde{\mathbf{L}}^{(d)}$ 's cannot sparsify the region well, and the results by  $\mathbf{L}$  have to be dense.

It is interesting to compare our isotropic nonlocal sparsifying transform with the one built upon plain patch recurrence without considering the isotropy property. Let  $\mathbb{P}_i$  denote the index set of patch  $\mathbf{p}_i$  constructed by finding the top- $K$  similar patches within  $\mathbb{M}_{\mathbf{p}_i}$ , *i.e.* the neighborhood of  $\mathbf{p}_i$ . Consider the plain nonlocal transform defined by

$$\hat{\mathbf{L}}(i, j) = \begin{cases} 1, & i = j; \\ -\frac{\omega(\mathbf{p}_i, \mathbf{p}_j)}{\sum_{k \in \mathbb{P}_i} \omega(\mathbf{p}_i, \mathbf{p}_k)}, & j \in \mathbb{P}_i; \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

Think of a cartoon patch  $\mathbf{p}$  on a straight edge whose similar patches lie along the direction of the edge. The index sets  $\mathbb{P}_i$ 's contains the locations of such similar patches. Thus,  $\hat{\mathbf{L}}$  can generate sparse results from  $\mathbf{p}$  by using the similar patches along the edge. In contrast,  $\mathbf{L}$  cannot generate sufficiently sparse results, as all  $\tilde{\mathbf{L}}^{(1)}, \dots, \tilde{\mathbf{L}}^{(D)}$  except the one corresponding to the edge direction, explicitly enforce using the patches along different directions which are dissimilar to  $\mathbf{p}$ .

TABLE I  
NOTATION/PARAMETER LIST FOR THE PROPOSED METHOD

Notations	Definition	Suggested Value
$M$	The number of wavelet filters	9 (8 high + 1 low)
$Q$	The number of local DCT filters	25
$N$	The number of pixels of image $\mathbf{f}$	-
$S$	The size of the whole searching regions	51
$D$	The number of directional banded regions	4
$B$	The width of directional banded regions	8
$\gamma$	The penalty weight from Bregman iteration	0.5
$\delta$	The step size from Bregman iteration	1
$\eta$	The scaling factor using in (13)	0.05
	(Key Parameters)	
$\beta_1$	The weight for the regularization of cartoon	Noiseless 2.5 Noisy 0.10
$\beta_2$	The weight for the regularization of texture	15 0.35

When  $\mathbf{p}$  is a patch in a texture region with similar patches scattering in the neighborhood, both  $\mathbb{P}_i$  and  $\mathbb{S}_i^{(d)}$  for all  $d$  contain the locations of similar patches. Thus, both  $\mathbf{L}$  and  $\hat{\mathbf{L}}$  can lead to sparse results. See Fig. 3 for a demonstration of the sparsification effects of  $\tilde{\mathbf{L}}^{(d)}$  and  $\hat{\mathbf{L}}$ . In summary, the plain transform  $\hat{\mathbf{L}}$  generates sparse results on both texture and cartoon patches and thus it is not discriminative enough to distinguish structure components from texture components. In contrast, the isotropic nonlocal transform  $\mathbf{L}$  is discriminative and only generates sparse results on texture components.

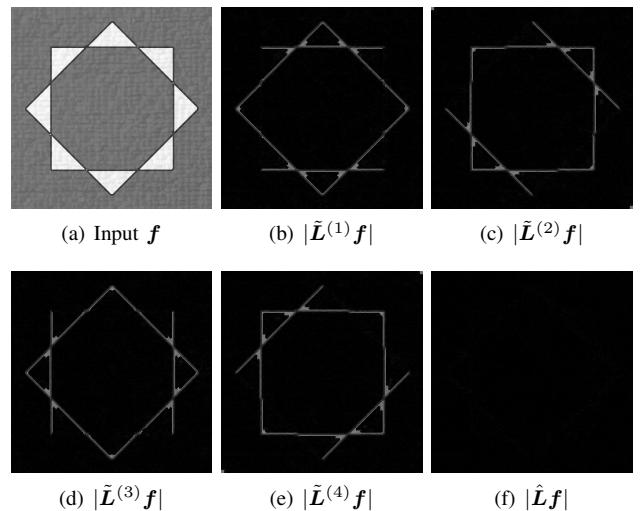


Fig. 3. Illustration of sparsification effects of  $\tilde{\mathbf{L}}^{(d)}$  and  $\hat{\mathbf{L}}^{(d)}$ . The notation  $|\cdot|$  denotes element-wise magnitude. It can be seen that both  $\tilde{\mathbf{L}}^{(d)}$  ( $d = 1, \dots, D$ ) and  $\hat{\mathbf{L}}$  sparsify the textural regions well. However, they show different behaviors on the cartoon edges. The plain nonlocal sparsifying transform  $\hat{\mathbf{L}}$  kills all the edges, while the isotropic nonlocal sparsifying transform  $\tilde{\mathbf{L}}^{(d)}$  only eliminates the edge along the  $d$ -th direction.

## IV. METHOD

### A. Model

Based on the isotropic nonlocal sparsifying transform constructed in the previous section, we propose a structure-texture decomposition method in the section. For convenience, the parameters of our method are summarized in Table I. Let  $*$  denote the 2D convolution operator and  $\vec{\cdot}$  denote the vectorization operator. Given a 2D convolutional kernel  $\mathbf{K}$ ,

we define the associated convolution matrix by  $\mathcal{S}_K \in \mathbb{R}^{N \times N}$ , which satisfies

$$\mathcal{S}_K \vec{\mathbf{X}} = \vec{\mathbf{K}} * \vec{\mathbf{X}}, \quad (9)$$

for a given 2D image  $\mathbf{X} \in \mathbb{R}^{H \times W}$  where  $HW = N$ .

Given a gray-scale image  $\mathbf{f} \in \mathbb{R}^N$  in the vectorized form, our goal is to extract its structure component  $\mathbf{u} \in \mathbb{R}^N$  and texture component  $\mathbf{v} \in \mathbb{R}^N$ . Towards this end, we solve the following MCA problem:

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{v}} & \|\text{diag}(\boldsymbol{\lambda}_1) \mathbf{W} \mathbf{u}\|_1 + \|\text{diag}(\boldsymbol{\lambda}_2) \mathbf{J} \mathbf{v}\|_1, \\ \text{s.t. } & \mathbf{u} + \mathbf{v} = \mathbf{f}, \end{aligned} \quad (10)$$

where  $\boldsymbol{\lambda}_1 \in \mathbb{R}^{MN}$ ,  $\boldsymbol{\lambda}_2 \in \mathbb{R}^{DQN}$  are two weighting vectors, and  $\mathbf{W}, \mathbf{J}$  are defined by

$$\begin{aligned} \mathbf{W} &= [\mathcal{S}_{G_1}; \dots; \mathcal{S}_{G_M}] \in \mathbb{R}^{MN \times N}, \\ \mathbf{J} &= [\mathbf{L} \mathcal{S}_{H_1}; \dots; \mathbf{L} \mathcal{S}_{H_Q}] \in \mathbb{R}^{DQN \times N}, \end{aligned} \quad (11)$$

where  $\{\mathbf{G}_m\}_{m=1}^M$  denotes a 2D wavelet filter bank and  $\{\mathbf{H}_q\}_{q=1}^Q$  denotes a set of 2D DCT filters. In implementation, we use the single-level linear spline wavelet filter bank [53], which contains  $M = 9$  two-dimensional filters  $\{\mathbf{G}_m\}_{m=1}^9 = \{\mathbf{g}_{k_1} \mathbf{g}_{k_2}^\top : 1 \leq k_1, k_2 \leq 3\}$  constructed by following 1D filers:

$$\mathbf{g}_1 = \frac{1}{4}(1, 2, 1)^\top; \mathbf{g}_2 = \frac{\sqrt{2}}{4}(1, 0, -1)^\top; \mathbf{g}_3 = \frac{1}{4}(-1, 2, -1)^\top.$$

See the supplementary materials for more details on the 2D wavelet filters. Regarding the DCT filters, we set the filter size to  $5 \times 5$  which corresponds to  $Q = 25$  2D-DCT filters.

There are two terms in the model (10). The first term is for the characterization of structure component using a wavelet filter bank. It is motivated by the advantages of generalized TV and wavelet-based methods in modelling piece-wise smooth signals. The filter bank corresponds to different operators with different orders along different orientations for sparsifying structure components. Since textures often contain dense small edges whose wavelet representation is unlikely to be sparse, the weighted  $\ell_1$  norm on  $\mathbf{W} \mathbf{u}$  can distinguish cartoon from texture well. In addition, the associated transform  $\mathbf{W}$  has the tight frame property [53]:  $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$ , by which the calculation of  $\mathbf{W}^\top \mathbf{W}$  can be avoided and the related computation can be accelerated. Note that since the low-pass filter does not sparsify images generally, it is omitted by setting the corresponding parts in  $\boldsymbol{\lambda}_1$  to zeros. We empirically found that using multi-level wavelet brings little improvement and even yield worse results. The reason is probably that more patterns are introduced into the wavelet filters as the level of wavelet increases, which decreases the accuracy the wavelet systems for modeling structure components. In fact, existing TV-based methods can also viewed as single-scale methods.

The second term is for texture characterization. The sparsification transform  $\mathbf{J}$  for texture first applies local DCT to sparsifying the image and then applies the nonlocal transform  $\mathbf{L}$  of (7) to sparsifying the DCT coefficients. In traditional approaches, local DCT is often employed for sparsifying the texture component. However, local DCT is not good at characterizing complex textures, e.g. weakly/non-periodic

patterns. In our model, the transform  $\mathbf{L}$  exploits the self-recurrence of texture for better sparsifying texture components. Even though the local DCT coefficients are not sparse, they can be sparsified under  $\mathbf{L}$ . It is also noted that while  $\mathbf{L}$  is not directly applied to sparsifying the texture component  $\mathbf{v}$ , but to the local DCT coefficients instead for generating a better sparse representation.

When there is noticeable image noise, the model (10) can be adapted to an unconstrained form for improving the robustness to noise. Assuming additive Gaussian white noise (AWGN) exists, the model becomes

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{v}} & \|\text{diag}(\boldsymbol{\lambda}_1) \mathbf{W} \mathbf{u}\|_1 + \|\text{diag}(\boldsymbol{\lambda}_2) \mathbf{J} \mathbf{v}\|_1 \\ & + \|\mathbf{f} - (\mathbf{u} + \mathbf{v})\|_2^2. \end{aligned} \quad (12)$$

### B. Calculation of Weighting Parameters

The regularization parameters  $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2$  are important to the performance of the proposed model. We exploit the isotropy of patch recurrence to define  $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2$  for further improvement. First, given an input image of  $N$  pixels, we define two spatial maps  $\bar{\boldsymbol{\lambda}}_1, \bar{\boldsymbol{\lambda}}_2 \in \mathbb{R}^N$  as follows:

$$\bar{\boldsymbol{\lambda}}_1(k) = 1 + e^{-\frac{\phi(k)}{\eta}}, \quad \bar{\boldsymbol{\lambda}}_2(k) = 1 - e^{-\frac{\phi(k)}{\eta}} \in (0, 1] \quad (13)$$

where  $\eta$  is a scaling factor set to 0.05, and  $\phi(k)$  measures the “anisotropicness” of patch recurrence around the  $k$ -th pixel based on our nonlocal transform, which is defined as follows:

$$\phi(k) = \frac{1}{DQ} \sum_{d=1}^D \sum_{i=1}^Q |(\tilde{\mathbf{L}}^{(d)} \mathbf{S}_{H_i} \mathbf{f})(k)|^2. \quad (14)$$

According to the discriminative patch recurrence prior, a patch  $\mathbf{p}_k$  in a textured region with strong isotropy property of patch recurrence will have small value of  $\phi$  and large value of  $e^{-\phi(k)/\eta}$ , as it can be well expressed by its similar patches along the each direction, which makes  $\tilde{\mathbf{L}}^{(d)} \mathbf{S}_{H_i} \mathbf{f}(k)$  close to zero for all  $d, i$ . In contrast, a patch  $\mathbf{p}_k$  lying on a cartoon contour will have relatively-large value of  $\phi$  and small value of  $e^{-\phi(k)/\eta}$ . See Fig. 4 for an illustration of  $e^{-\phi(k)/\eta}$ . Since  $\bar{\boldsymbol{\lambda}}_1, \bar{\boldsymbol{\lambda}}_2$  have small values on the cartoon contours and texture regions respectively, they are used as the weights for the regularizations of structure and texture components. Then, we define the weighting vectors  $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2$  in model (10) as follows:

$$\boldsymbol{\lambda}_1 = [\underbrace{\beta_1 \bar{\boldsymbol{\lambda}}_1; \dots; \beta_1 \bar{\boldsymbol{\lambda}}_1}_M], \quad \boldsymbol{\lambda}_2 = [\underbrace{\beta_2 \bar{\boldsymbol{\lambda}}_2; \dots; \beta_2 \bar{\boldsymbol{\lambda}}_2}_{DQ}] \quad (15)$$

where  $\beta_1, \beta_2$  are the weights of the regularization terms. In other words,  $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2$  only change spatially and they have repeating values across different channels of  $\mathbf{W}$  and  $\mathbf{J}$ . Recall that since the coefficients of low-pass filtering on cartoon part are generally not sparse, the corresponding values in  $\boldsymbol{\lambda}_1$  are set to 0 in implementation.

**Remark [Revisiting discrimination of proposed model].** The discrimination of our model comes from various sources, including the isotropy of  $\mathbf{L}$ , the adaption of  $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2$ , and the difference between filters, which play different roles. (i) The proposed transform  $\mathbf{L}$  exploits discriminative patch recurrence to sparsify texture layers but not structure layers, which is

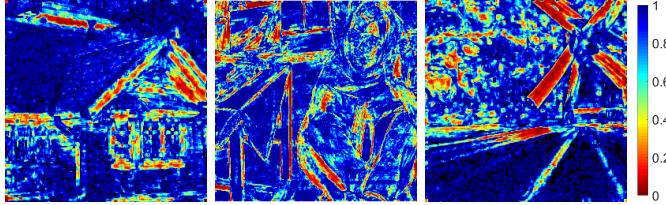


Fig. 4. Illustration of the exponential response  $e^{-\phi(\cdot)/\eta}$  on the images 'Cabin' (left), 'Barbara' (middle) and 'Windmill' (right). It can be seen that the values are high on textured regions and low on cartoon contours.

the main source of discrimination. (ii) The weighting parameters  $\lambda_1, \lambda_2$  also exploit the discriminative patch recurrence by adjusting the penalty over regions for better preserving weak edges/patterns in structure/texture components, which is another main source of discrimination. (iii) The wavelet and DCT filters are used to exploit different local sparsity patterns of structure and texture components respectively, bringing additional discrimination.

### C. Numerical Algorithm

Following the MCA framework, we define an augmented system  $\mathbf{D} = [\mathbf{W} \ \mathbf{0}; \ \mathbf{0} \ \mathbf{J}]$  and then rewrite the minimization problem (10) and (12) into

$$\min_{\mathbf{x}} \|\text{diag}(\boldsymbol{\lambda})\mathbf{D}\mathbf{x}\|_1, \quad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{f}, \quad (16)$$

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{f}\|_2^2 + \|\text{diag}(\boldsymbol{\lambda})\mathbf{D}\mathbf{x}\|_1, \quad (17)$$

where  $\mathbf{x} = [\mathbf{u}^\top, \mathbf{v}^\top]^\top$ ,  $\mathbf{A} = [\mathbf{I}, \mathbf{I}]$ ,  $\boldsymbol{\lambda} = [\lambda_1^\top, \lambda_2^\top]^\top$ . The problem can be efficiently solved by the ADMM (also known as split Bregman [54]). The overall algorithm is summarized in Algorithm 1, and each of its step is detailed in the following.

Regarding the unconstrained problem (17), we first rewrite it as the constrained problem

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{f}\|_2^2 + \|\text{diag}(\boldsymbol{\lambda})\mathbf{d}\|_1, \quad \text{s.t. } \mathbf{d} = \mathbf{D}\mathbf{x}, \quad (18)$$

which is the ‘‘splitting’’ step to separate  $\mathbf{x}$  from the  $\ell_1$  penalty. Next, the Bregman iteration is used to solve (18) as follows:

$$\begin{cases} (\mathbf{x}^{(k+1)}, \mathbf{d}^{(k+1)}) = \underset{\mathbf{x}, \mathbf{d}}{\text{argmin}} \|\mathbf{A}\mathbf{x} - \mathbf{f}\|_2^2 \\ \quad \|\text{diag}(\boldsymbol{\lambda})\mathbf{d}\|_1 + \frac{\gamma}{2} \|\mathbf{D}\mathbf{x} - \mathbf{d} + \mathbf{e}^{(k)}\|_2^2 \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} + \delta(\mathbf{D}\mathbf{x}^{(k+1)} - \mathbf{d}^{(k+1)}) \end{cases} \quad (19)$$

for  $k = 0, 1, \dots$ , where  $\gamma > 0$  and  $0 < \delta \leq 1$  are two parameters arising from the Bregman iteration. By decomposing the first two problem in (19) into another two, we further rewrite (19) as follows:

$$\begin{cases} \mathbf{x}^{(k+1)} = \underset{\mathbf{x}}{\text{argmin}} \|\mathbf{A}\mathbf{x} - \mathbf{f}\|_2^2 + \frac{\gamma}{2} \|\mathbf{D}\mathbf{x} - \mathbf{d}^{(k)} + \mathbf{e}^{(k)}\|_2^2 \\ \mathbf{d}^{(k+1)} = \underset{\mathbf{d}}{\text{argmin}} \|\text{diag}(\boldsymbol{\lambda})\mathbf{d}\|_1 + \frac{\gamma}{2} \|\mathbf{D}\mathbf{x}^{(k+1)} - \mathbf{d} + \mathbf{e}^{(k)}\|_2^2 \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} + \delta(\mathbf{D}\mathbf{x}^{(k+1)} - \mathbf{d}^{(k+1)}) \end{cases} \quad (20)$$

In (20), the subproblem regarding  $\mathbf{x}$  is a least-squares regression which is about solving the linear system:

$$(\mathbf{A}^\top \mathbf{A} + \gamma \mathbf{D}^\top \mathbf{D}) \mathbf{x}^{(k+1)} = \mathbf{A}^\top \mathbf{f} + \gamma \mathbf{D}^\top (\mathbf{d}^{(k)} - \mathbf{e}^{(k)}). \quad (21)$$

This system is solved by the conjugate gradient method. The subproblem regarding  $\mathbf{d}$  is separable to each dimension of  $\mathbf{d}$  and thus has the analytic solution given by

$$\mathbf{d}^{(k+1)} = T_{\frac{\lambda}{\gamma}}(\mathbf{D}\mathbf{x}^{(k+1)} + \mathbf{e}^{(k)}), \quad (22)$$

where  $T_{\boldsymbol{\lambda}}(\cdot)$  is the soft-thresholding operation defined by

$$(T_{\boldsymbol{\lambda}}(\mathbf{x}))(k) = \text{sign}(\mathbf{x}(k)) \max(|\mathbf{x}(k)| - \lambda_k, 0). \quad (23)$$

Combining (20), (21) and (22), the problem (17) is solved by the following iteration:

$$\begin{cases} \mathbf{x}^{(k+1)} = (\mathbf{A}^\top \mathbf{A} + \gamma \mathbf{D}^\top \mathbf{D})^{-1} (\mathbf{A}^\top \mathbf{f} + \gamma \mathbf{D}^\top (\mathbf{d}^{(k)} - \mathbf{e}^{(k)})) \\ \mathbf{d}^{(k+1)} = T_{\frac{\lambda}{\gamma}}(\mathbf{D}\mathbf{x}^{(k+1)} + \mathbf{e}^{(k)}) \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} + \delta(\mathbf{D}\mathbf{x}^{(k+1)} - \mathbf{d}^{(k+1)}) \end{cases} .$$

Regarding the noiseless model (16) which has an additional constraint compared to (17), we can solve it with a similar scheme which applies an additional Bregman iteration to handle the fidelity constraint.

---

### Algorithm 1 Structure-Texture Image Decomposition

---

**Input:** Image  $f$

**Output:** Structure component  $u$ , Texture component  $v$

**Main procedure:**

- 1) Find the similar patches with directional grouping scheme and construct  $L$  according to (5) and (7);
  - 2) Calculate  $\lambda_1$  and  $\lambda_2$  according to (15);
  - 3) Construct the system  $\mathbf{D} = [\mathbf{W} \ \mathbf{0}; \ \mathbf{0} \ \mathbf{J}]$  accordingly;
  - 4)  $\mathbf{d}^{(0)} := \mathbf{D}\mathbf{x}^{(0)}$ ,  $\mathbf{e}^{(0)} := \mathbf{0}$ ;
  - 5) For  $k = 0, \dots, K-1$ ,
    - a) 
$$\begin{cases} \mathbf{x}^{(k+1)} = (\mathbf{A}^\top \mathbf{A} + \gamma \mathbf{D}^\top \mathbf{D})^{-1} (\mathbf{A}^\top \mathbf{f} + \gamma \mathbf{D}^\top (\mathbf{d}^{(k)} - \mathbf{e}^{(k)})) \\ \mathbf{d}^{(k+1)} = T_{\frac{\lambda}{\gamma}}(\mathbf{D}\mathbf{x}^{(k+1)} + \mathbf{e}^{(k)}) \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} + \delta(\mathbf{D}\mathbf{x}^{(k+1)} - \mathbf{d}^{(k+1)}) \end{cases}$$
  - 6)  $u = [\mathbf{I}, \mathbf{0}] \mathbf{x}^{(K)}$  and  $v = [\mathbf{0}, \mathbf{I}] \mathbf{x}^{(K)}$ .
- 

## V. EXPERIMENTS

### A. Protocols and Settings

The performance of the proposed approach for structure-texture image decomposition is evaluated with both the noiseless and noisy settings. In the noiseless setting, input images are clean with an ignorable amount of noise, and we solve the problem (10). In comparison, the noisy setting involves noisy images for testing the performance of (12), which also can be viewed as the application to image denoising. When ground-truth images available, we use PSNR and SSIM to measure the visual quality of the decomposition results.

Several representative or state-of-the-art methods are selected for comparison, including

- ADMGB (Alternating Direction Method with Gaussian Back Substitution) [3]: using the TV norm and dual TV norm for structure and texture respectively;
- RTV (Relative Total Variation) [34]: a filtering based method using inherent variation and relative TV measures;
- BNN (Block-wise Nuclear Norm) [26]: imposing low-rank property on texture blocks;

- RGF (Rolling Guidance Filter) [48]: a filtering-based method using an iterative guidance;
- GID3D (Group-based Image Decomposition 3D) [28]: using low-rank decomposition on grouped similar patches;
- SCSC (Slice-based Convolutional Sparse Coding) [20]: sparse dictionary learning on structure/texture layer;
- JCAS (Joint Convolutional Analysis and Synthesis) [29]: joint analysis and synthesis sparse models;
- NLDD (Non-local Dual Domain Decomposition) [47]: a very recent nonlocal method with theoretical analysis.

To test whether the discriminative prior of patch recurrence is useful, we construct a baseline denoted by 'Baseline', which is the same as our method except that it uses the plain nonlocal sparsifying transform  $\hat{\mathbf{L}}$  defined in (5) instead of the new one. We keep  $\{\mathbf{G}_m\}_{m=1}^M, \{\mathbf{H}_q\}_{q=1}^Q$  of Baseline the same as our method and also use the same parameters for patch matching. For the model parameters  $\beta_1, \beta_2$ , we search their best values within the ranges used for tuning up our method.

In implementation, we use local DCT filters of size  $5 \times 5$  (*i.e.*  $Q = 25$ ). The maximal number of iteration is set to  $K = 50$ . In the construction of  $\mathbf{L}$ , the size of search window is set to  $S = 51$  with 16 matched patches in each direction the number of bands is set to  $D = 4$  and the width of each band is set to  $B = 8$ . The size of patches used in matching is  $5 \times 5$ . For the ADMM iteration, the parameter  $\gamma$  is set to 0.5 and the update step  $\delta$  is set to 1. The model parameters are set as  $\beta_1 = 2.5, \beta_2 = 0.1$  for pure image decomposition and  $\beta_1 = 15, \beta_2 = 0.35$  for noisy image decomposition. A pre-denoising same as [28] is applied before matching for fair comparison. Implemented in Matlab with GPU acceleration, our method takes around 26 seconds to process a  $256 \times 256$  image on an Intel i7-6700K CPU and an RTX 2080Ti GPU.

### B. Decomposition on Synthetic Images

We first test the proposed method on 100 synthetic images, whose details are given in our supplementary materials. With the truth of each component, we calculate the PSNR/SSIM on the structure and texture components respectively. The average PSNR/SSIM values are listed in Table II. Our method performs the best in terms of both PSNR and SSIM.

See Fig. 5 for some decomposition results. The benefit of using our discriminative patch recurrence prior is demonstrated by our superior results over Baseline. In the close-ups of structure components, the edges of the cross are broken and blurred by Baseline, while well preserved in our result. In the close-ups of texture components, the contour edges of the cross appear clearly in the texture component of Baseline, while correctly rejected by our method. In the comparison with other methods, ours also shows noticeable improvement. Regarding the structure component, BNN, GID3D, NLDD and SCSC mistakenly preserve some texture patterns in their cartoon parts. Regarding the texture components, all the compared methods expect BNN and ours produce visible counter edges.

**Influence of parameters  $\beta_1, \beta_2$ .** Varying  $\beta_1, \beta_2$  within a moderate range, we test their influence to the decomposition results. It is observed that increasing  $\beta_1$  or decreasing  $\beta_2$  will move the elements from texture components to structure components, and vice versa. We also observe that the

decomposition is stable to the small changes of  $\beta_1, \beta_2$ . See the supplementary materials for more details.

**Influence of patch size in patch matching.** We set the patch size in patch matching to some common values respectively and then test the performance of the proposed method. The quantitative results are given in Table III. See also the supplementary materials for the visual comparison on some recovered results using different patch sizes. Overall, the patch size has only a little influence to the performance.

TABLE II  
AVERAGE PSNR (DB) AND SSIM VALUES OF THE DECOMPOSITION RESULTS ON SYNTHETIC IMAGES BY DIFFERENT METHODS.

Criterion	Structure		Texture	
	PSNR	SSIM	PSNR	SSIM
ADMGB	27.29	0.940	27.68	0.874
RTV	30.78	0.969	30.78	0.937
BNN	30.25	0.737	30.26	0.805
RGF	30.82	0.940	30.81	0.931
GID3D	27.77	0.697	27.81	0.769
JCAS	29.09	0.885	28.31	0.841
SCSC	27.56	0.846	27.57	0.879
NLDD	29.31	0.799	29.32	0.836
Baseline	24.41	0.909	24.41	0.890
Ours	<b>33.48</b>	<b>0.977</b>	<b>33.32</b>	<b>0.938</b>

### C. Decomposition on Natural Images

1) *Gray-scale images:* We further test the proposed method on the five natural images shown in Fig. 6(a), where 'House' and 'Barbara' are quoted from [20], [26], [28]. These images contain different types of textures (*e.g.* handmade vs. natural and regular vs. random), which can test different aspects of a method. Since PSNR and SSIM are unavailable without ground truths, we compare the decomposition results by visual inspection with three main criteria. Firstly, contour edges should only appear in the structure component. Secondly, well-patterned features should only appear in the texture component. Thirdly, no artifacts are present in both the components.

Due to space limitation, we only show the results of five most recent methods, and the results of the other methods are given in supplementary materials. The results on 'Barbara' given in the first row shows that the proposed method can extract well-patterned features in the texture layer while rejecting them in the structure layer. To show the advantages of our method over others, we zoom in the table cloth and the face of Barbara. In the structure components of both table cloth, only ADMGB, JCAS, Baseline and our method can completely remove the textures. Compared to JCAS which

TABLE III  
AVERAGE PSNR (DB) AND SSIM VALUES OF THE DECOMPOSITION RESULTS ON SYNTHETIC IMAGES WITH DIFFERENT PATCH SIZES

Criterion	3	5	7	9	11	13
Structure	PSNR	33.37	33.48	33.31	33.23	33.19
	SSIM	0.972	0.977	0.974	0.970	0.967
Texture	PSNR	33.20	33.32	33.10	32.95	32.85
	SSIM	0.937	0.938	0.933	0.929	0.927

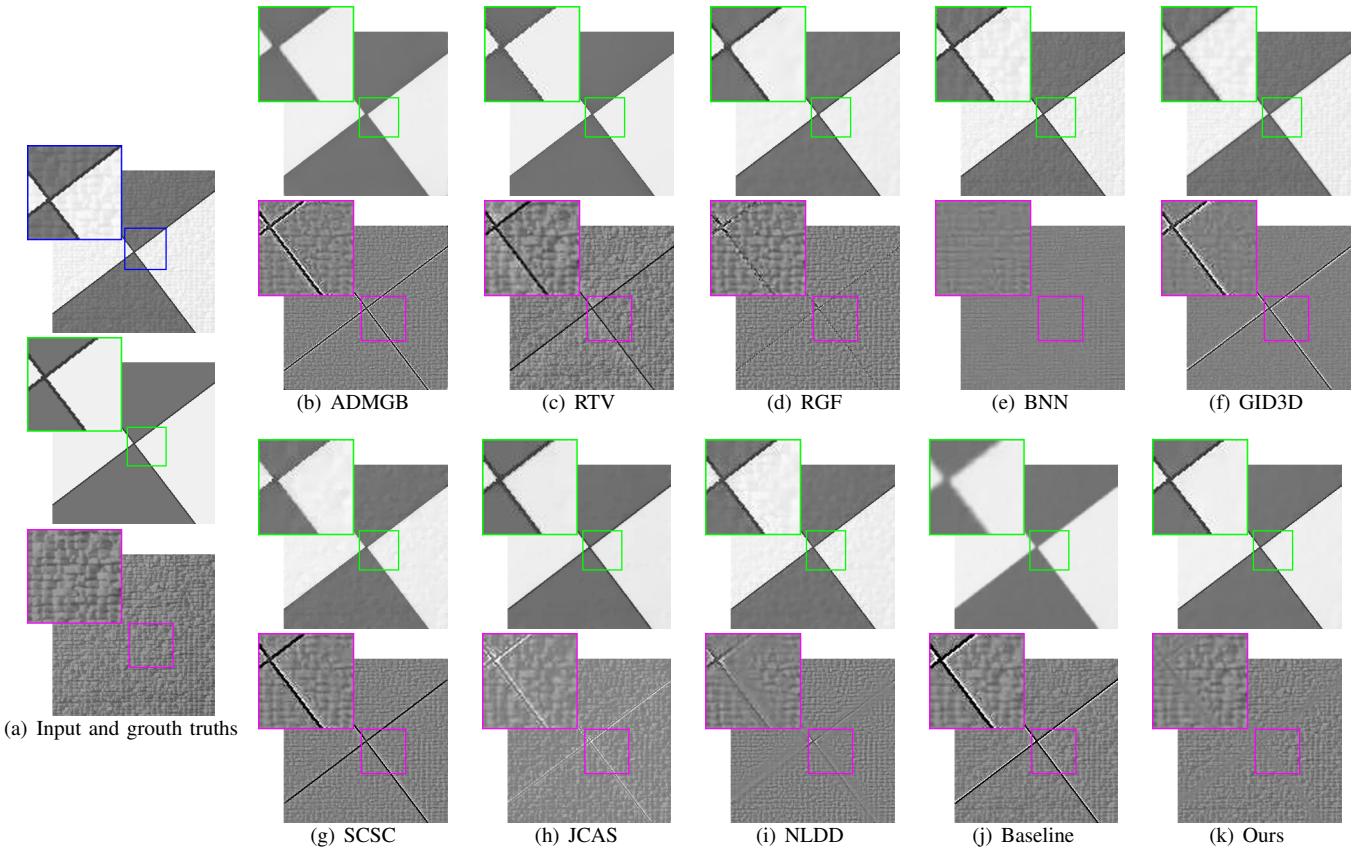


Fig. 5. The structure component (top) and the texture component (bottom) of the decomposition results on 'Cross' (*i.e.* the 1st synthetic images). (a) Input and ground-truths of structure and texture components. (b)-(k) decomposition results of different methods.

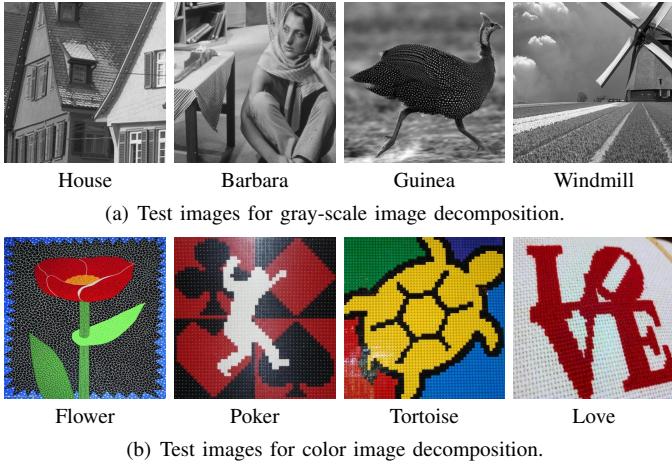


Fig. 6. Some test images used in our experiments.

fails to extract the cloth textures, our method correctly handles the textured patterns. Compared with ADMGB and Baseline that blur the structure component, our method preserves more details in cartoon.

Fig. 7 also gives the results on 'House', which shows the capability of our method in preserving contour edges in structure layers and rejecting them in texture layers. This can be more clearly demonstrated in the close-ups of the eave that contains sharp cartoon edges. In our results, the

eave is completely kept in the structure layer. In contrast, the other compared methods except BNN assign a strong edge along the eave to the texture layer, resulting in blur or even removal of the eave in cartoon. In particular, the three patch-recurrence-based methods (*i.e.* GID3D, NLDD and Baseline) obviously blur the eave, without exploiting the discriminative patch recurrence prior.

Similar phenomena can be found in the results of 'Guinea' and 'Windhill', where the proposed method handles both the structure and texture layers well.

2) *Color images*: We also extend the proposed method to color images, by processing each color channel separately. We use four images quoted from [34] for test, which are shown in Fig. 6(b). In Fig. 8, we compare the results of the proposed method with RTV, RGF and NLDD, as they provided codes for color image decomposition. For other methods that only provide codes for gray-scale images, we also extend them in the same way as ours, but their results are shown in our supplementary materials. As the ground truths are unavailable, the results are also compared by visual inspection with the same criteria as that we use on gray-scale images.

The results of 'Flower' are showed in the first row of Fig. 8(b). It can be observed that RTV and Baseline fail to preserve the edge of the petal in the structure component while RGF and NLDD do better but still blur the edge. On the other hand, NLDD has a problem in getting rid of the small texture from the structure layer while RTV and RGF do better but

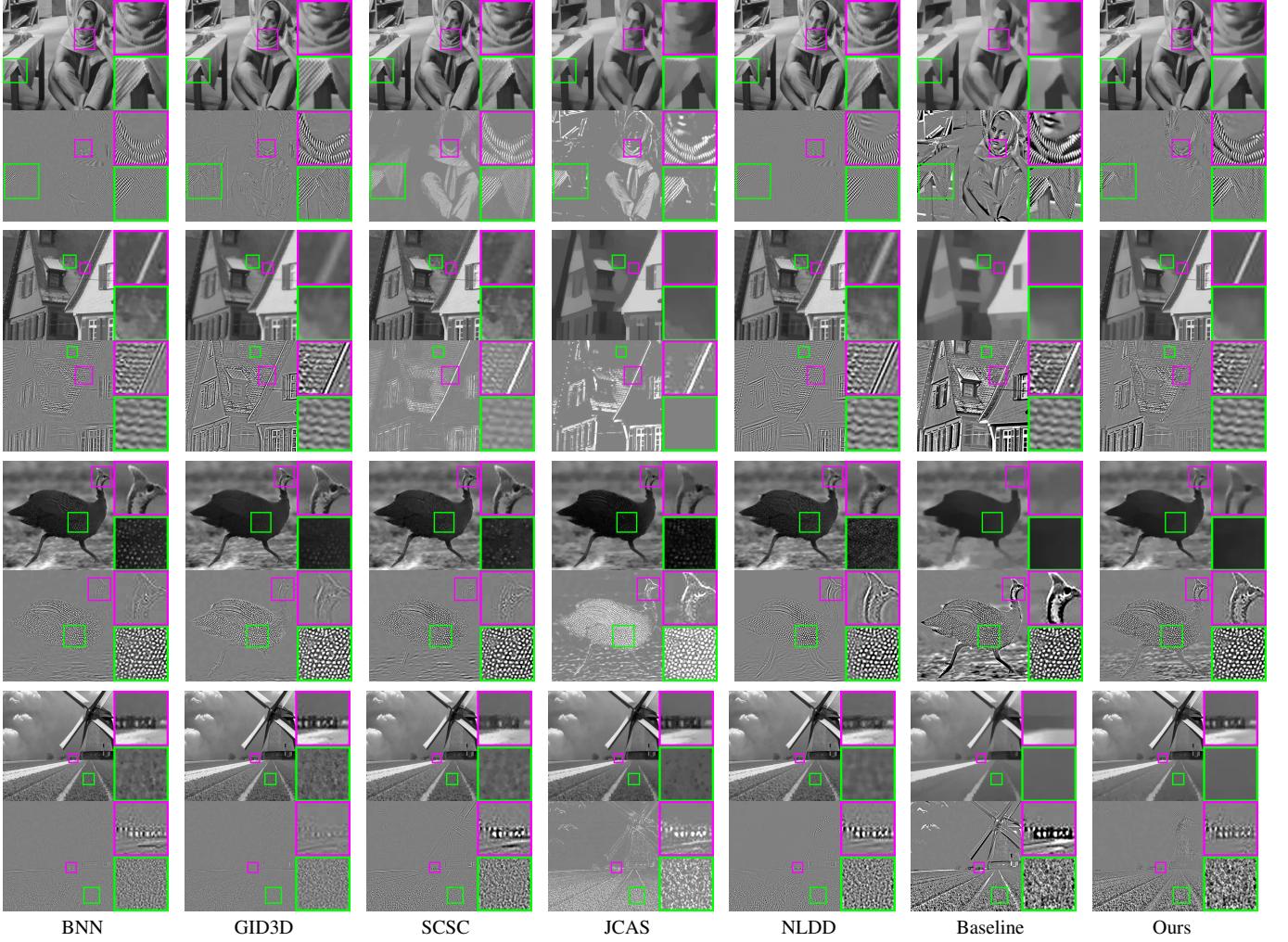


Fig. 7. The structure component (top) and the texture component (bottom) of the decomposition results on different images.

still leave some spots from texture in the structure layer. In comparison, the proposed method does not suffer from either of the above problems. Similar phenomena can also be found in the other test cases, where the proposed method preserves the edge in the structure component well and correctly extracts the texture in the meanwhile.

To conclude, all the visual results on natural images have demonstrated the superior performance of our method over the existing ones, as well as the effectiveness of the discriminative patch recurrence prior in structure-texture decomposition.

#### D. Decomposition on Noisy Images

To evaluate the robustness of our method in noisy settings, we use four noisy images for test, which are shown in Fig. 9. Among the test images, ‘Barbara-1’, ‘Barbara-2’ and ‘Roof’ are quoted from [26], [28]. Following [28], the original images are degraded by adding the AWGN with noise level  $\sigma = 0.1$  and then used as inputs. As a result, each input image contains three components: cartoon, texture and noise. We decompose an input image into these three components by solving (12). For comparison, BNN and GID3D are selected, as they have available results and codes. The evaluation is done in terms of visual inspection, as well as PSNR and SSIM.

The decomposition results are shown in Fig. 11. Our method performs well even in the presence of noises, with better visual quality than other methods. Notice the results of BNN and GID3D. The textures of the chair in Fig. 11(a) and the tiles on the cabin in Fig. 11(d), are present in the cartoon instead of the texture components. In contrast, our method extracts the texture components correctly in these cases. Fig. 11 also shows the denoised images, which are calculated by removing the noise component from the input. Compared with BNN and GID3D, our method preserves more details with less artifacts.

Note that our method may assign a few unexpected large structures to the texture components, *e.g.* Fig. 11(a). See also supplementary materials for the results using different patch sizes for matching. This is probably due to the improper patch size used in the directional patch matching on the corresponding regions. When using a fixed patch size across the image, the patch size should be set to strike the balance between stability and effectiveness. On one hand, a small patch size may make the patch matching sensitive to noise and decrease the stability of matching. On the other hand, a large patch size may include many different patterns to the patches, lowering both accuracy and effectiveness of patch matching. One possible scheme for improvement is using adaptive patch

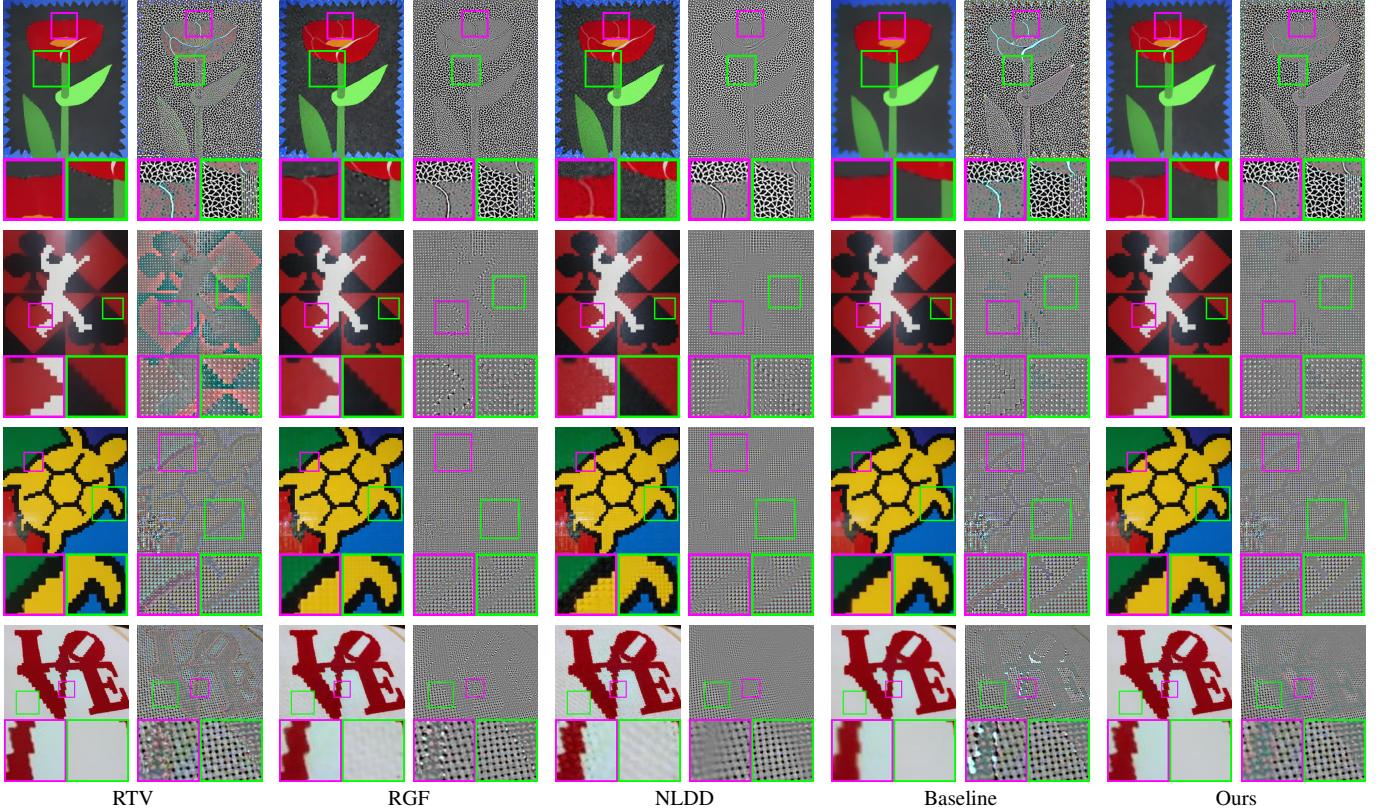


Fig. 8. The structure component (top) and the texture component (bottom) of the decomposition results on different images.

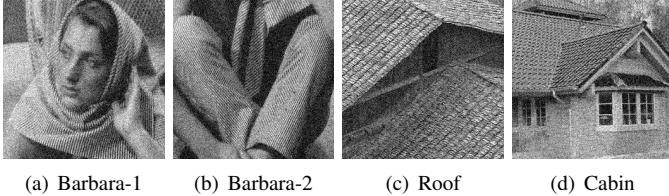


Fig. 9. Test images for noisy decomposition.

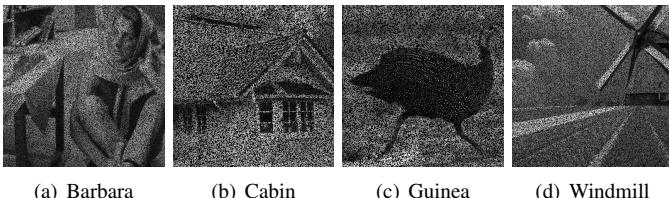


Fig. 10. Test images for inpainting.

matching which sets spatially-varying patch sizes according to some properties (*e.g.* distribution of gradients) of local regions.

Following [52], we also evaluate the results in terms of the randomness in the noise component. The less structures appear in the noise component (implying higher degree of randomness), the lower coherence exists between noise components and other components, and the better the denoising performance is. It can be seen that, compared with those of other methods, our produced noise components are more random with lower correlation to the input images. The PSNR and SSIM values of the denoised images are listed in Table IV,

where our method achieves the highest ones on all the images except 'Roof'. Additionally, we also vary the noise level for the test. The results are shown in the supplementary materials, where our method produces reasonable results for images degraded by the AWGN of different levels.

In order to quantitatively measure the decomposition performance, we conduct the noisy decomposition on the synthetic images used in the previous subsection where the AWGN with noise level  $\sigma = 0.1$  is added. See Table V for the comparison with GID3D. Our method outperformed GID3D in terms of PSNR and SSIM on both structure and texture layers.

#### E. Decomposition with Missing Pixels

Following [26], we further extend our method to handling the decomposition with missing pixels (*i.e.* image inpainting). The extension is simply done by modifying the constraint  $\mathbf{u} + \mathbf{v} = \mathbf{f}$  to  $\mathbf{u}_{\mathbb{I}} + \mathbf{v}_{\mathbb{I}} = \mathbf{f}_{\mathbb{I}}$ , where  $\mathbb{I}$  contains the indexes of the known pixels in  $\mathbf{f}$ . Accordingly, the numerical solution is straightforward by setting  $\mathbf{A} = [\mathbf{P}_{\mathbb{I}}, \mathbf{P}_{\mathbb{I}}]$  in place of  $\mathbf{A} = [\mathbf{I}, \mathbf{I}]$  where  $\mathbf{P}_{\mathbb{I}}$  is the projection matrix such that  $(\mathbf{P}_{\mathbb{I}}\mathbf{x})(i) = \mathbf{x}(i)$  if  $i \in \mathbb{I}$  and 0 otherwise.

The test images are shown in Fig. 10, which are generated using 'Barbara', 'Jackstraw', 'House' and 'Snake' with 40% pixels damaged. Our method is compared with BNN which has available results and code. The results are evaluated both visually and quantitatively. See Fig. 12 for the visual results, where BNN fails to remove the textures in the restored structure layer, making its texture layer rather 'weak'. In comparison, our method produces much clearer structure and texture layers.

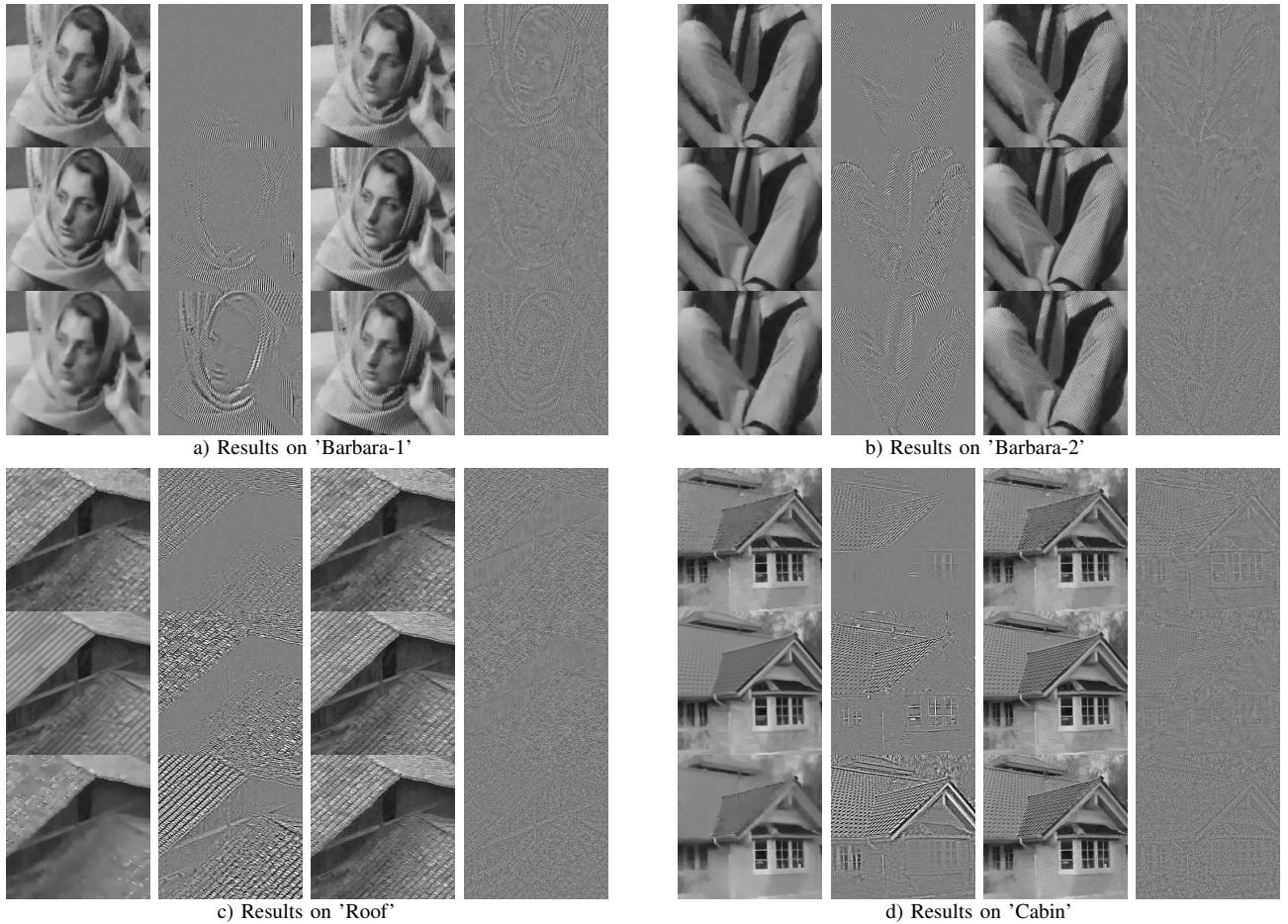


Fig. 11. Decomposition results on noisy images. For each image, the rows from top to bottom correspond to BNN, GID3D and the proposed method. The columns from left to right to the structure components, texture components, denoised images, and noise components.

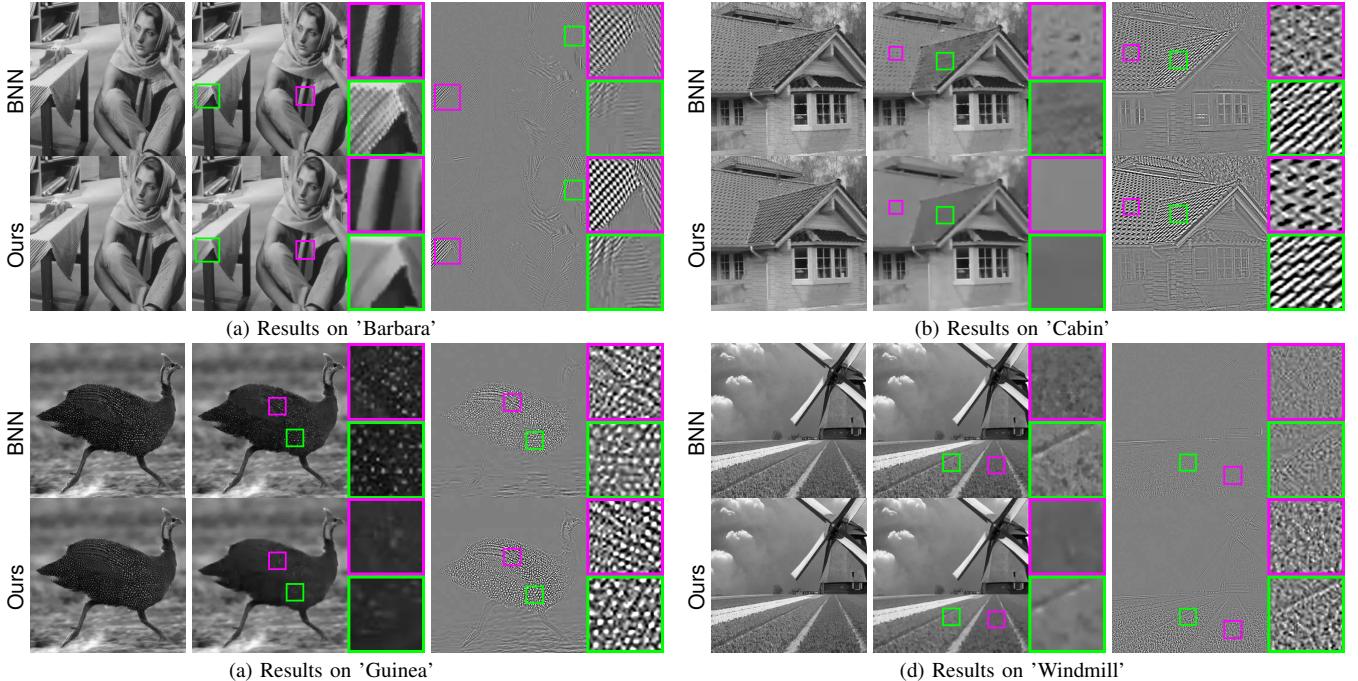


Fig. 12. Results on images with 40% pixels missing. For each image, the rows from top to down correspond to BNN and the proposed method. The columns from left to right correspond to and the recovered image, the structure and the texture components.

TABLE IV  
PSNR (dB) AND SSIM VALUES OF DENOISED IMAGES

Method	Barabara-1	Barabara-2	Roof	Cabin	Average
BNN	26.72/0.765	25.43/0.778	23.32/0.740	24.06/0.699	24.88/0.746
GID3D	27.99/0.820	26.51/0.824	<b>23.92</b> /0.779	25.91/0.782	26.08/0.802
Ours	<b>28.30/0.825</b>	<b>27.44/0.847</b>	23.80/ <b>0.780</b>	<b>25.98/0.786</b>	<b>26.38/0.810</b>

TABLE V  
COMPARISON OF GID3D AND OUR METHOD ON NOISY DECOMPOSITION  
ON SYNTHETIC IMAGES WITH NOISE LEVEL  $\sigma = 0.1$

Method	PSNR		SSIM	
	Structure	Texture	Structure	Texture
GID3D	27.00	26.18	0.800	0.550
Ours	<b>29.59</b>	<b>28.28</b>	<b>0.959</b>	<b>0.647</b>

TABLE VI  
PSNR (dB) AND SSIM VALUES OF INPAINTED IMAGES

Method	Measure	Barbara	House	Guinea	Windmill	Average
BNN	PSNR	32.87	28.99	26.24	32.40	30.13
	SSIM	0.959	0.922	0.937	0.943	0.940
Ours	PSNR	<b>35.50</b>	<b>30.82</b>	<b>26.99</b>	<b>33.31</b>	<b>31.66</b>
	SSIM	<b>0.971</b>	<b>0.943</b>	<b>0.944</b>	<b>0.952</b>	<b>0.953</b>

The PSNR and SSIM values are listed in Table VI. Our method outperforms BNN by a large margin on the test images.

## VI. SUMMARY AND FUTURE WORK

In this paper, we introduced the discriminative patch recurrence prior to structure-texture image decomposition. The prior can be viewed as a refinement of the traditional patch recurrence prior, which states that, though the patch recurrence property exists in both the structure and texture components, the directions of recurrences, or say the spatial distributions of similar patches, are much different between the two components. Similar patches on cartoon contours are usually distributed along one direction, while similar patches in texture regions tend to scatter in multiple directions. Such a prior enables distinguishing textures from cartoons by considering the spatial configuration of recurrent patches.

According to the discriminative patch recurrence prior, we constructed an isotropic nonlocal sparsifying transform for texture components, built upon which an effective approach is developed for structure-texture decomposition. The performance of the proposed approach was evaluated in both noiseless and noisy settings, as well as in inpainting. The experimental results showed the superior performance of the proposed approach to the state-of-the-art ones, which demonstrated the power of the proposed approach as well as the effectiveness of the discriminative patch recurrence prior.

In future, we would like to further exploit the discriminative patch recurrence in four directions. Firstly, though multi-level wavelet did not bring improvement to our method, multi-scale information is undoubtedly useful for structure-texture decomposition. Thus, we will investigate how to utilize multi-scale or scale-adaptive patch matching and the behaviors of discriminative patch recurrence over scales for further improvement. Secondly, we have shown the possible applications

of the discriminative prior to image denoising and inpainting. To continue, we will investigate the adaption of the proposed method to general image inverse problem. Thirdly, our method processes color images by treating the color channels independently. Such a strategy cannot fully exploit the additional cues existing in color channels. Thus, we will study how to work better on decomposing color images.

## REFERENCES

- [1] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Proc.*, vol. 12, no. 8, pp. 882–889, 2003.
- [2] G. Gilboa, N. Sochen, and Y. Y. Zeevi, "Variational denoising of partly textured images by spatially varying constraints," *IEEE Trans. Image Proc.*, vol. 15, no. 8, pp. 2281–2289, 2006.
- [3] M. K. Ng, X. Yuan, and W. Zhang, "Coupled variational image decomposition and restoration model for blurred cartoon-plus-texture images with missing pixels," *IEEE Trans. Image Proc.*, vol. 22, no. 6, pp. 2233–2246, 2013.
- [4] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, "An improved algorithm for  $TV-\ell_1$  optical flow," in *Statistical and Geometrical Approaches to Visual Motion Analysis*. Springer, 2009, pp. 23–45.
- [5] K. Cao, E. Liu, and A. K. Jain, "Segmentation and enhancement of latent fingerprints: A coarse to fine ridgestructure dictionary," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 9, pp. 1847–1859, 2014.
- [6] X. Hu, W. Xia, S. Peng, and W.-L. Hwang, "Multiple component predictive coding framework of still images," in *Proc. IEEE Int. Conf. Multimedia and Expo*. IEEE, 2011, pp. 1–6.
- [7] F. Calderero and V. Caselles, "Recovering relative depth from low-level features without explicit t-junction detection and interpretation," *Int. J. Comput. Vision*, vol. 104, no. 1, pp. 38–68, 2013.
- [8] Y. Han, C. Xu, G. Baciu, M. Li, and M. R. Islam, "Cartoon and texture decomposition-based color transfer for fabric images," *IEEE Trans. Multimedia*, vol. 19, no. 1, pp. 80–92, 2017.
- [9] Z. Liang, J. Xu, D. Zhang, Z. Cao, and L. Zhang, "A hybrid 11-l0 layer decomposition model for tone mapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition*, 2018, pp. 4758–4766.
- [10] B. Cornelis, H. Yang, A. Goodfriend, N. Ocon, J. Lu, and I. Daubechies, "Removal of canvas patterns in digital acquisitions of paintings," *IEEE Trans. Image Proc.*, vol. 26, no. 1, pp. 160–171, 2016.
- [11] J. Qian, J. Yang, and Y. Xu, "Local structure-based image decomposition for feature extraction with applications to face recognition," *IEEE Trans. Image Proc.*, vol. 22, no. 9, pp. 3591–3603, 2013.
- [12] I. N. Figueiredo, S. Kumar, C. M. Oliveira, J. D. Ramos, and B. Engquist, "Automated lesion detectors in retinal fundus images," *Computers in biology and medicine*, vol. 66, pp. 47–65, 2015.
- [13] J.-F. Aujol, G. Aubert, L. Blanc-Féraud, and A. Chambolle, "Image decomposition into a bounded variation component and an oscillating component," *J. Math. Imag. and Vis.*, vol. 22, no. 1, pp. 71–88, 2005.
- [14] J. Gilles and Y. Meyer, "Properties of  $bv - g$  structures + textures decomposition models. application to road detection in satellite images," *IEEE Trans. Image Proc.*, vol. 19, no. 11, pp. 2793–2800, 2010.
- [15] I. Yanovsky and A. B. Davis, "Separation of a cirrus layer and broken cumulus clouds in multispectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2275–2285, 2015.
- [16] M. J. Fadili, J.-L. Starck, J. Bobin, and Y. Moudden, "Image decomposition and separation using sparse representations: an overview," *Proc. The IEEE*, vol. 98, no. 6, pp. 983–994, 2010.
- [17] S. Osher, A. Solé, and L. Vese, "Image decomposition and restoration using total variation minimization and the  $h^{-1}$  norm," *Multiscale Model. Simul.*, vol. 1, no. 3, pp. 349–370, 2003.
- [18] W. Yin, D. Goldfarb, and S. Osher, "Total variation based image cartoon-texture decomposition," COLUMBIA UNIV NEW YORK DEPT OF INDUSTRIAL ENGINEERING AND OPERATIONS RESEARCH, Tech. Rep., 2005.
- [19] H. Zhang and V. M. Patel, "Convolutional sparse coding-based image decomposition," in *Proc. British Machine Vis. Conf.*, 2016.
- [20] V. Papyan, Y. Romano, J. Sulam, and M. Elad, "Convolutional dictionary learning via local processing," *arXiv preprint*, 2017.
- [21] H. Zhang and V. M. Patel, "Convolutional sparse and low-rank coding-based image decomposition," *IEEE Trans. Image Proc.*, vol. 27, no. 5, pp. 2121–2133, 2017.

- [22] J.-L. Starck, Y. Moudden, J. Bobin, M. Elad, and D. Donoho, "Morphological component analysis," in *Wavelets XI*, vol. 5914. International Society for Optics and Photonics, 2005, p. 59140Q.
- [23] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering," in *Image Process.: Algorithms and Syst., Neural Networks, and Mach. Learn.*, vol. 6064. International Society for Optics and Photonics, 2006, p. 606414.
- [24] Y. Quan, H. Ji, and Z. Shen, "Data-driven multi-scale non-local wavelet frame construction and image recovery," *Journal. Scientific Computing*, vol. 63, no. 2, pp. 307–329, 2015.
- [25] H. Schaeffer and S. Osher, "A low patch-rank interpretation of texture," *SIAM J. Imaging Sci.*, vol. 6, no. 1, pp. 226–262, 2013.
- [26] S. Ono, T. Miyata, and I. Yamada, "Cartoon-texture image decomposition using blockwise low-rank texture characterization," *IEEE Trans. Image Proc.*, vol. 23, no. 3, pp. 1128–1142, 2014.
- [27] Y.-R. Fan, T.-Z. Huang, T.-H. Ma, and X.-L. Zhao, "Cartoon–texture image decomposition via non-convex low-rank texture regularization," *J. the Franklin Institute*, vol. 354, no. 7, pp. 3170–3187, 2017.
- [28] T.-H. Ma, T.-Z. Huang, and X.-L. Zhao, "Group-based image decomposition using 3-d cartoon and texture priors," *Inform. Sci.*, vol. 328, pp. 510–527, 2016.
- [29] S. Gu, D. Meng, W. Zuo, and L. Zhang, "Joint convolutional analysis and synthesis sparse representation for single image layer separation," in *Proc. IEEE Int. Conf. Comput. Vision*. IEEE, 2017, pp. 1717–1725.
- [30] Y. Meyer, *Oscillating patterns in image processing and nonlinear evolution equations: the fifteenth Dean Jacqueline B. Lewis memorial lectures*. American Mathematical Soc., 2001, vol. 22.
- [31] L. A. Vese and S. J. Osher, "Modeling textures with total variation minimization and oscillating patterns in image processing," *Journal. Scientific Computing*, vol. 19, no. 1-3, pp. 553–572, 2003.
- [32] W. Yin, D. Goldfarb, and S. Osher, "Image cartoon-texture decomposition and feature selection using the total variation regularized  $l_1$  functional," in *Variational, Geometric, and Level Set Methods in Computer Vision*. Springer, 2005, pp. 73–84.
- [33] V. Duval, J.-F. Aujol, and L. A. Vese, "Mathematical modeling of textures: Application to color image decomposition with a projected gradient algorithm," *J. of Math. Imaging and Vision*, vol. 37, no. 3, pp. 232–248, 2010.
- [34] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Trans. Graph.*, vol. 31, no. 6, p. 139, 2012.
- [35] Y. Sun, S. Schaefer, and W. Wang, "Image structure retrieval via  $l_0$  minimization," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 7, pp. 2129–2139, 2017.
- [36] E. Tadmor, S. Nezzar, and L. Vese, "A multiscale image representation using hierarchical (BV,  $L^2$ ) decompositions," *Multiscale Model. Simul.*, vol. 2, no. 4, pp. 554–579, 2004.
- [37] J.-L. Starck, M. Elad, and D. L. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," *IEEE Trans. Image Proc.*, vol. 14, no. 10, pp. 1570–1582, 2005.
- [38] P. Maurel, J.-F. Aujol, and G. Peyré, "Locally parallel texture modeling," *SIAM J. Imaging Sci.*, vol. 4, no. 1, pp. 413–447, 2011.
- [39] J.-F. Cai, B. Dong, S. Osher, and Z. Shen, "Image restoration: total variation, wavelet frames, and beyond," *J. The American Mathematical Society*, vol. 25, no. 4, pp. 1033–1089, 2012.
- [40] Y. Kim, B. Ham, M. N. Do, and K. Sohn, "Structure-texture image decomposition using deep variational priors," *IEEE Trans. Image Proc.*, vol. 28, no. 6, pp. 2692–2704, 2018.
- [41] J.-L. Starck, M. Elad, and D. L. Donoho, "Image decomposition: Separation of texture from piecewise smooth content," in *Wavelets: Applications in Signal and Image Processing*, vol. 5207. International Society for Optics and Photonics, 2003, pp. 571–583.
- [42] J.-F. Aujol and A. Chambolle, "Dual norms and image decomposition models," *Int. J. Comput. Vision*, vol. 63, no. 1, pp. 85–104, 2005.
- [43] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher, "Structure-texture image decomposition—modeling, algorithms, and parameter selection," *Int. J. Comput. Vision*, vol. 67, no. 1, pp. 111–136, 2006.
- [44] A. Buades, T. M. Le, J.-M. Morel, and L. A. Vese, "Fast cartoon+ texture image filters," *IEEE Trans. Image Proc.*, vol. 19, no. 8, pp. 1978–1986, 2010.
- [45] A. Buades and J. L. Lisani, "Directional filters for color cartoon+ texture image and video decomposition," *J. of Math. Imaging and Vision*, vol. 55, no. 1, pp. 125–135, 2016.
- [46] J. Gilles, "Multiscale texture separation," *Multiscale Model. Simul.*, vol. 10, no. 4, pp. 1409–1427, 2012.
- [47] F. Sur, "A non-local dual-domain approach to cartoon and texture decomposition," *IEEE Trans. Image Proc.*, vol. 28, no. 4, pp. 1882–1894, 2018.
- [48] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," in *Proc. Eur. Conf. Comput. Vision*. Springer, 2014, pp. 815–830.
- [49] F. Zhang, L. Dai, S. Xiang, and X. Zhang, "Segment graph based image filtering: Fast structure-preserving smoothing," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 361–369.
- [50] B. Ham, M. Cho, and J. Ponce, "Robust image filtering using joint static and dynamic guidance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition*, 2015, pp. 4823–4831.
- [51] X. Guo, Y. Li, J. Ma, and H. Ling, "Mutually guided image filtering," *IEEE Trans. Trans. Pattern Anal. Mach. Intell.*, 2018.
- [52] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.
- [53] Z. Shen, "Wavelet frames and image restorations," in *Proc. Int. Congr. Mathematicians*. World Scientific, 2010, pp. 2834–2863.
- [54] T. Goldstein and S. Osher, "The split bregman method for  $l_1$ -regularized problems," *SIAM J. Imaging Sci.*, vol. 2, no. 2, pp. 323–343, 2009.

**Ruotao Xu** received his B.Eng. degree in computer science from South China University of Technology, Guangzhou, China, in 2015, where he is currently a Post-Doctoral Research Fellow with the School of Computer Science and Engineering. His research interests include image processing, sparse coding, and deep learning.

**Yong Xu** received his B.Sc., M.Sc. and Ph.D. degrees in mathematics from Nanjing University, Nanjing, China, in 1993, 1996, and 1999, respectively. He was a Post-Doctoral Research Fellow in computer science with South China University of Technology, Guangzhou, China, from 1999 to 2001, where he is currently a professor with the School of Computer Science and Engineering. His current research interests include computer vision and pattern recognition.

**Yuhui Quan** received his doctoral degree in computer science from South China University of Technology, Guangzhou, China, in 2013. He was a Post-Doctoral Research Fellow in mathematics with the National University of Singapore, Singapore, from 2013 to 2016. He is currently an Associate Professor with the School of Computer Science and Engineering, South China University of Technology. His research interests include computer vision, image processing, deep learning, and sparse representation.