

Joint Convolutional Analysis and Synthesis Sparse Representation for Single Image Layer Separation

Shuhang Gu¹, Deyu Meng², Wangmeng Zuo³, Lei Zhang^{1,*}

¹ The Hong Kong Polytechnic University, ² Xi'an Jiaotong University, ³ Harbin Institute of Technology
{shuhangu@gmail.com, dymeng@mail.xjtu.edu.cn, cswmzuo@gmail.com, cslzhang@comp.polyu.edu.hk}

Abstract

Analysis sparse representation (ASR) and synthesis sparse representation (SSR) are two representative approaches for sparsity-based image modeling. An image is described mainly by the non-zero coefficients in SSR, while is mainly characterized by the indices of zeros in ASR. To exploit the complementary representation mechanisms of ASR and SSR, we integrate the two models and propose a joint convolutional analysis and synthesis (JCAS) sparse representation model. The convolutional implementation is adopted to more effectively exploit the image global information. In JCAS, a single image is decomposed into two layers, one is approximated by ASR to represent image large-scale structures, and the other by SSR to represent image fine-scale textures. The synthesis dictionary is adaptively learned in JCAS to describe the texture patterns for different single image layer separation tasks. We evaluate the proposed JCAS model on a variety of applications, including rain streak removal, high dynamic range image tone mapping, etc. The results show that our JCAS method outperforms state-of-the-arts in these applications in terms of both quantitative measure and visual perception quality.

1. Introduction

In many computer vision and photography applications such as rain streak removal [24], high dynamic range (HDR) image tone mapping [10], reflection removal [23], cartoon-texture decomposition [3], etc., image layer separation is a key step to produce satisfactory results. Given an input image \mathbf{y} , image layer separation aims to decompose the image as the summation of two components:

$$\mathbf{Y} = \mathbf{U} + \mathbf{V}. \quad (1)$$

Since the number of unknowns \mathbf{U} and \mathbf{V} are larger than the input \mathbf{Y} , single image layer separation is a challenging ill-

posed problem. Based on the requirements on the property of separating results, different priors have been suggested to regularize the decomposition results. As a powerful model for image prior modeling, the sparsity prior has been widely adopted in different kinds of layer separation applications [39, 23, 25].

One category of methods utilize analysis sparse representation (ASR) models to characterize the piece-wise smooth layer, and impose the sparsity prior on the filter responses (analysis representation coefficients) over the latent estimation. The Total-Variation (TV) approach [31] and its extensions [39], which are proposed to regularize the gradients of latent estimation, have been successfully used in extracting the piece-wise smooth component from the input image. However, since a sparse analysis coefficient implies that the signal (an image patch in image contexts) should be orthogonal to most of the filters, such a prior modeling mechanism impedes ASR to take benefit from increased redundancy of the analysis dictionary. Despite its success on approximating the major structure of an image, ASR often shows limited capacity in modeling textures or fine-scale details with complex patterns.

Another category of sparse representation models, i.e. synthesis sparse representation (SSR) model, has also been utilized to deal with image layer separation applications. Compared with the ASR model, the SSR approach investigates signals from a synthesis point of view: $\mathbf{x} = \mathbf{D}\alpha_S$. The sparse representation coefficients α_S select a few atoms from the dictionary \mathbf{D} to reconstruct the signal vector \mathbf{x} . Such a mechanism enables SSR taking advantage of an over-complete dictionary to provide flexible prior. Given an appropriate dictionary (learned from training data), signals with specific patterns can be reconstructed with highly sparse coefficients. Recently, several SSR-based methods have been suggested by learning different dictionaries for separating an input image into two layers.

Fig. 1 provides a visual example to illustrate how ASR and SSR represent a target image. An ASR dictionary learning [28] and an SSR dictionary learning [15] methods are used to train dictionaries on the target image, respective-

*This work is supported by HK RGC General Research Fund (PolyU 152124/15E) and National Natural Science Foundation of China (grant no. 61672446).

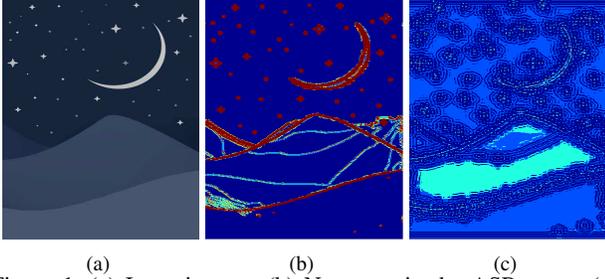


Figure 1. (a) Input image. (b) Nonzeros in the ASR map. (c) Nonzeros in the SSR map. Dark blue pixels indicates coefficients with less nonzeros and red ones indicates coefficients with more than five nonzeros.

ly, and we show the number of non-zeros in their respective coefficient maps. One can see that the ASR coefficients can be very sparse in relatively smooth areas, but it will generate many non-zeros in texture areas. While having sparse non-zeros in texture areas, SSR needs more non-zero coefficients to approximate smooth areas.

In order to take the advantages of both ASR and SSR, we propose a joint convolutional analysis and synthesis (JCAS) sparse representation model for image layer decomposition. More specifically, we propose to use ASR and SSR to approximate the two components of an image, respectively. Since the two models are superior in approximating different components of an image, it is expected to get a piecewise constant layer with image major structures and a texture layer with fine-scale details. A similar idea has been suggested by Starck et al. [34], who utilized a total variation term and a synthesis-based sparse reconstruction term for cartoon-texture decomposition. Comparatively, the proposed JCAS model adopts a convolutional implementation for the SSR part. The convolutional implementation avoids the patch-dividing issue in conventional SSR methods and enables the proposed method to learn only several atoms from the input image itself to model the complex (but highly repetitive) textures.

The contribution of this paper is mainly two-fold. First, we analyze the complementary property of the ASR and SSR models, and integrate ASR and SSR to decompose a single image into two layers with clear physical meanings. The JCAS model is expected to inspire the designing of new sparsity-based methods for the many applications involving image layer separation. Second, without external training data, the proposed JCAS model achieves competitive results on different tasks such as rain streak removal, high dynamic range image tone mapping and contrast enhancement. The results produced by JCAS not only have higher quantitative indexes, but also are with better visual qualities.

2. Related Works

2.1. ASR and SSR for Image Modeling

Given a signal $\mathbf{x} \in \mathfrak{R}^{K \times 1}$, the ASR method adopts an analysis dictionary $\mathbf{P} \in \mathfrak{R}^{M \times K}$ to generate its analysis coefficients $\alpha_A = \mathbf{P}\mathbf{x}$. Such a projective coding strategy allows us to impose the ASR prior on images in a convolution manner. Taking Gaussian image denoising as an example, given the input noisy image \mathbf{Y} , its clean estimation \mathbf{X} can be obtained by solving:

$$\hat{\mathbf{X}} = \arg \min_{\mathbf{X}} \|\mathbf{X} - \mathbf{Y}\|_F^2 + \lambda \sum_i R_i(\mathbf{f}_{A,i} \otimes \mathbf{X}), \quad (2)$$

where $\|\cdot\|_F$ and \otimes represent the Frobenius-norm and the convolution operator, respectively. The penalty function $R_i(\mathbf{f}_{A,i} \otimes \mathbf{X})$ enforces prior knowledge on the latent image by regularizing its analysis coefficients to the filters $\mathbf{f}_{A,i}$.

One of the most notable analysis-based methods is the TV approach [31], which regularizes the ℓ_1 -norm of the gradients of the latent estimation \mathbf{x} . The success of TV model prompted the in-depth studies on ASR models from different aspects, including fast optimization algorithms [5], proper penalty functions [21], and rational analysis dictionaries [33]. Methods [28, 17, 7, 30] have also been proposed to learn analysis dictionaries from training data for better image local structure representation.

The SSR method characterizes signals from a synthesis point of view. Most of previous SSR based methods work on image patches:

$$\hat{\alpha}_S = \arg \min_{\alpha_S} \|\mathbf{y} - \mathbf{D}\alpha_S\|_F^2 + \gamma R(\alpha_S), \text{ s.t. } \mathbf{x} = \mathbf{D}\hat{\alpha}_S, \quad (3)$$

where \mathbf{y} and \mathbf{x} are vectorizations of patches extracted from input and target images, respectively. \mathbf{x} is synthesized as $\mathbf{D}\hat{\alpha}_S$, and the regularization term $R(\alpha_S)$ promotes the sparsity of synthesis coefficients. The success of KSVD [1] triggers the study of SSR on different image restoration and representation tasks [40, 26, 25]. In-depth studies on dictionaries as well as penalty functions [26, 4] have been conducted to pursue better performance. Recently, beyond conventional patch-based implementation of SSR, the convolutional sparse coding method [42] has been proposed to decompose an image with synthesis filters and coefficient maps.

Although both the ASR and SSR models exploit the image sparsity prior, they use different yet complementary representation mechanisms, making them superior in different applications. ASR has achieved a great success in image denoising and blind deblurring applications, while SSR has advantages in providing flexible priors for recovering details in super-resolution applications. Note that SSR has also achieved state-of-the-art results for image denoising, but it needs to cooperate with other priors such as the non-local self-similarity prior [26].

2.2. Single Image Layer Decomposition

Rain streak removal aims to decompose a rainy image into a rain-free background layer and a rain streak layer. Assuming that rain streaks only appear in the high-frequency part of an image, researchers [19, 35] have proposed to decompose the image into low-frequency and high-frequency layers, and separate rain streaks from background details in the high-frequency layer. However, these methods often over-smooth image details and generate blurry background estimation. Recent works have been proposed to directly extract the rain streak layer from the input image, e.g., by using the discriminative sparse coding method [25] and the Gaussian mixture model (GMM) [24]. However, these methods adopt the same type of models to characterize the background part and the rain streak part, and require external data to train different dictionaries or GMM models for the two layers.

HDR tone mapping targets to generate a low dynamic range (LDR) image from an HDR image without loss of significant details. Retinex decomposition model [22] is widely used to separate the HDR image into a base layer and a detail layer. The dynamic range is reduced by compressing the base layer, while detail layer can be enhanced for better visibility. The base/detail layer separation plays a crucial role in the success of tone mapping. One category of methods adopts edge-preserving smoothing method, such as bilateral filter [9] and guided filter [18], to generate a piece-wise smooth base layer from the observation image directly. Another class of methods [10, 14] impose priors on the base layer, and the separation results are achieved by solving an optimization problem.

Others applications Like in tone mapping, in other applications such as cartoon-texture decomposition [27, 39, 3] and contrast or detail enhancement [13], the input image is also separated into a piece-wise smooth (cartoon) component and a detail (texture) component [3]. Many methods adopt different priors for the two layers [27]. The priors on cartoon layer can be set as to the analysis-based TV model and its extensions [3]. However, researchers still lack consensus on the selection of regularization functional form on texture layer. Algorithms have been proposed to model texture layer in the original image space [6] or in certain transformation spaces [2], while there are still some recent works [39] not assigning priors for the texture layer.

3. The Proposed Model

3.1. Joint Convolutional Analysis-Synthesis Model

Single image layer separation is an ill-posed problem, and thus priors of the desired solution are required to provide supplementary information. In the proposed JCAS model, an SSR prior term and an ASR prior term are utilized to regularize the two layers, respectively.

The image separation is achieved by solving the following objective function:

$$\min_{\mathbf{u}, \mathbf{z}} \|\mathbf{Y} - \mathbf{U} - \sum_j^N \mathbf{f}_{S,j} \otimes \mathbf{Z}_j\|_F^2 + \lambda \sum_i^M \|\mathbf{f}_{A,i} \otimes \mathbf{U}\|_1 + \gamma \sum_j^N \|\mathbf{Z}_j\|_1, \quad (4)$$

where $\|\cdot\|_1$ is the ℓ_1 norm, and λ and γ are regularization parameters imposed on the analysis and synthesis prior terms, respectively. Here we model the SSR component \mathbf{V} as $\mathbf{V} = \sum_j^N \mathbf{f}_{S,j} \otimes \mathbf{Z}_j$, where $\mathbf{f}_{S,j}$ is the j -th atom of convolutional synthesis dictionary, \mathbf{Z}_j is its corresponding coefficient map, and “ \otimes ” denotes the convolution operation. Note that we use the convolutional sparse coding for SSR to avoid partitioning the image into patches.

The analysis prior $\|\mathbf{f}_{A,i} \otimes \mathbf{U}\|_1$ is introduced to characterize the ASR component \mathbf{U} by regularizing the sparseness of its filter responses over analysis filters. As discussed in the previous sections, ASR is capable of better modeling the major structures of an image. Thus, the \mathbf{U} layer is expected to represent the background, cartoon and illuminance layers in the rain streak removal, texture-cartoon decomposition and HDR tone mapping applications, respectively. For layer $\mathbf{V} = \sum_j^N \mathbf{f}_{S,j} \otimes \mathbf{Z}_j$, we regularize its synthesis coefficients \mathbf{Z} over the convolutional synthesis dictionary $\mathbf{f}_{S,j}$. Compared with ASR, SSR is a more effective model to reconstruct complex but repetitive textures. Thus, approximating the rain streak components, image textures or fine details with the synthesis layer \mathbf{V} will lead to a lower energy of the objective function.

3.2. Choice of Dictionaries

In (4), an analysis dictionary $\{\mathbf{f}_{A,i}\}_{i=1,\dots,M}$ and a synthesis dictionary $\{\mathbf{f}_{S,j}\}_{j=1,\dots,N}$ are adopted to assign priors for the two image layers, respectively. The two dictionaries should be properly specified to make JCAS a powerful image separation model.

The dictionary plays an important role in deducing an appropriate sparse representation of the input signal [29]. The early studies on sparse representation often utilize mathematical tools to design and fix a class of functions as the dictionary for data representation in a hand-craft manner. During the last decade, in order to achieve a finer adaption to specific instances of the data, dictionary learning methods have been investigated from a data-driven point of view [29]. Compared with hand-crafted dictionaries, the dictionary learned from data is capable of delivering better results due to its adaptability to represent the targeted scenario. However, for those applications (such as texture-cartoon decomposition) where training data are hard to collect to train the desired dictionary, hand-crafted dictionary is still preferred due to its simplicity and efficiency. In this paper, we utilize different strategies to set the two dictionaries for ASR and SSR, based on their different characteristics.

ASR utilizes the analysis dictionary to model the complementary subspace of signals. Each dictionary atom will be compared with the signal (by the inner product). Although this limits the employment of a highly redundant dictionary to provide more flexible prior, it makes ASR be a robust model in capturing image major structures. Even with an extremely simple analysis dictionary (e.g., the gradient operators), ASR can still achieve very competitive results in some applications [31]. Thus, in our method, we adopt the simple gradient operators (1st order and 2nd order) as our analysis dictionary for fast decomposition.

Different from the ASR model, the SSR method selects dictionary atoms to reconstruct the given signal. Having an appropriate dictionary, SSR is able to reconstruct the input signal with very sparse coefficients. However, a hand-crafted dictionary is hard to reconstruct the complex image structures using only a few atoms, and a proper dictionary learning method is required to learn synthesis dictionaries from training data [1]. In this paper, we learn a convolutional synthesis dictionary from the input image itself. Such a strategy not only avoids the requirement of external training data with candidate texture types, but also makes JCAS be able to represent the texture layer with only several atoms. The detailed synthesis dictionary learning method will be introduced in the following section.

3.3. Optimization

As introduced in the previous section, our method learns the synthesis dictionary during the decomposition process. Thus, for the objective function in (4), the synthesis dictionary $\{\mathbf{f}_{S,j}\}_{j=1,\dots,N}$ is a variable to be optimized. We rewrite the convolution in a matrix multiplication form, and add some constraints to ensure the boundness of the synthesis filters. The new objective function for our JCAS model has the following form:

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{f}_{S,j}, \mathbf{z}_j} \|\mathbf{y} - \mathbf{u} - \sum_j \mathbf{F}_{S,j} \mathbf{z}_j\|_2^2 + \lambda \sum_i \|\mathbf{F}_{A,i} \mathbf{u}\|_1 + \gamma \sum_j \|\mathbf{z}_j\|_1, \\ \text{s.t. } \|\mathbf{f}_{S,j}\|_F^2 \leq 1, \end{aligned} \quad (5)$$

where \mathbf{y} , \mathbf{u} and \mathbf{z}_j are the vectorization of image \mathbf{Y} , background layer \mathbf{U} and feature map \mathbf{Z}_j , respectively. $\mathbf{F}_{A,i}$ and $\mathbf{F}_{S,j}$ are the corresponding block circulant with circulant block (BCCB) matrices of filters $\mathbf{f}_{A,i}$ and $\mathbf{f}_{S,j}$, respectively. We update the three variables alternatively, and details of each sub-problem are described as follows.

Updating \mathbf{u} To solve the subproblem with respect to \mathbf{u} , we fix $\{\mathbf{f}_{S,j}\}_{j=1\dots N}$ and $\{\mathbf{z}_j\}_{j=1\dots N}$ and solve the following optimization problem:

$$\min_{\mathbf{u}} \|\mathbf{y} - \mathbf{u} - \sum_j \mathbf{F}_{S,j} \mathbf{z}_j\|_2^2 + \lambda \sum_i \|\mathbf{F}_{A,i} \mathbf{u}\|_1. \quad (6)$$

Denote the fixed variables $\mathbf{y} - \sum_j \mathbf{F}_{S,j} \mathbf{z}_j$ by \mathbf{x} , and introduce a group of auxiliary variables $\{\mathbf{s}_i = \mathbf{F}_{A,i} \mathbf{u}\}_{i=1,\dots,M}$, we

Algorithm 1 JCAS algorithm for image decomposition

Input: Input image \mathbf{Y} , analysis filters $\{\mathbf{f}_{A,i}\}_{i=1,\dots,M}$, regularization parameters λ, γ

- 1: **for** $k=1:K$ **do**
- 2: Update \mathbf{u}^k by (7)
- 3: if $k == 1$, initialize $\{\mathbf{f}_{S,j}\}_{j=1,\dots,N}$ as the PCA basis of the patches in $(\mathbf{y} - \mathbf{u}^1)$
- 4: Update \mathbf{z}_j^k by (8)
- 5: Update synthesis filters $\{\mathbf{f}_{S,j}\}_{j=1,\dots,N}$ by (10)
- 6: **end for**

Output: Decomposition results \mathbf{U} and \mathbf{V}

can readily solve (6) by the ADMM algorithm:

$$\begin{cases} \mathbf{u}^{k+1} &= (\frac{\mu_k}{2} \sum_i \mathbf{F}_{A,i}^T \mathbf{F}_{A,i} + \mathbf{I})^{-1} \\ &\quad (\mathbf{x} + \frac{\mu_k}{2} \sum_i \mathbf{F}_{A,i}^T \mathbf{s}_i + \frac{1}{\mu_k} \sum_i \mathbf{F}_{A,i} \mathbf{L}_i); \\ \mathbf{s}_i^{k+1} &= \mathcal{S}_{\frac{\lambda}{\mu_k}}(\mathbf{F}_{A,i} \mathbf{u}^{k+1} + \frac{1}{\mu_k} \mathbf{L}_i); \\ \mathbf{L}_i^{k+1} &= \mathbf{L}_i^k + \mu_k (\mathbf{F}_{A,i} \mathbf{u}^{k+1} - \mathbf{s}_i); \\ \text{if } \mu_k &< \mu_{max}, \quad \mu_{k+1} = \mu_k * \rho; \end{cases} \quad (7)$$

where \mathbf{L}_i is the Lagrange variable for \mathbf{s}_i , μ_{max} and ρ are the parameters in the algorithm. $\mathcal{S}_{\frac{\lambda}{\mu_k}}(*)$ denotes the soft-thresholding operator with parameter $\frac{\lambda}{\mu_k}$, which is the solution for the ℓ_1 -norm approximation problem. Thanks to the property of BCCB matrix, the closed-form solution in the \mathbf{u} -step in (7) can be efficiently solved in the FFT domain.

Updating \mathbf{z} Fixing \mathbf{u} and the synthesis dictionary \mathbf{f}_S , we solve the following sub-problem to obtain \mathbf{z} :

$$\min_{\mathbf{z}} \|\mathbf{y} - \mathbf{u} - \sum_j \mathbf{F}_{S,j} \mathbf{z}_j\|_2^2 + \gamma \sum_j \|\mathbf{z}_j\|_1. \quad (8)$$

The optimization problem in (8) is a standard convolutional sparse coding problem. We utilize the algorithm in [38] to solve it, which adopts the ADMM scheme and exploits the FFT to improve computation efficiency.

Updating \mathbf{f}_S With the fixed \mathbf{u} and coefficients \mathbf{z} , we need to update the synthesis dictionary. Let $\text{vec}(\mathbf{f}_{S,j} \otimes \mathbf{Z}_j) = \mathbf{F}_{S,j} \mathbf{z}_j = \mathcal{Z} \mathbf{f}_S$, where \mathbf{f}_S is the vectorization of all the filters $\{\mathbf{f}_{S,j}\}_{j=1,\dots,N}$, $\mathcal{Z} = [\mathcal{Z}_1, \dots, \mathcal{Z}_j, \dots, \mathcal{Z}_N]$, and \mathcal{Z}_j is generated by collecting the patches in \mathbf{Z}_j . The objective function can be re-written as the following equivalent form:

$$\min_{\mathbf{f}_S} \|\mathbf{y} - \mathbf{u} - \mathcal{Z} \mathbf{f}_S\|_2^2, \quad \text{s.t. } \|\mathbf{f}_{S,j}\|_F^2 \leq 1. \quad (9)$$

We utilize a proximal gradient descent method to solve (9):

$$\begin{cases} \mathbf{f}_S^{t+0.5} &= \mathbf{f}_S^t - \tau \mathcal{Z}^T (\mathbf{y} - \mathbf{u} - \mathcal{Z} \mathbf{f}_S^t); \\ \mathbf{f}_S^{t+1} &= \text{Prox}_{\|\cdot\| \leq 1}(\mathbf{f}_S^{t+0.5}). \end{cases} \quad (10)$$

In (10), τ is the step length of the gradient descent step, and $\text{Prox}_{\|\cdot\| \leq 1}(*)$ is the ℓ_2 -ball proximal operator, which makes each filter satisfy the constraint $\|\mathbf{f}_{S,j}\|_F^2 \leq 1$.

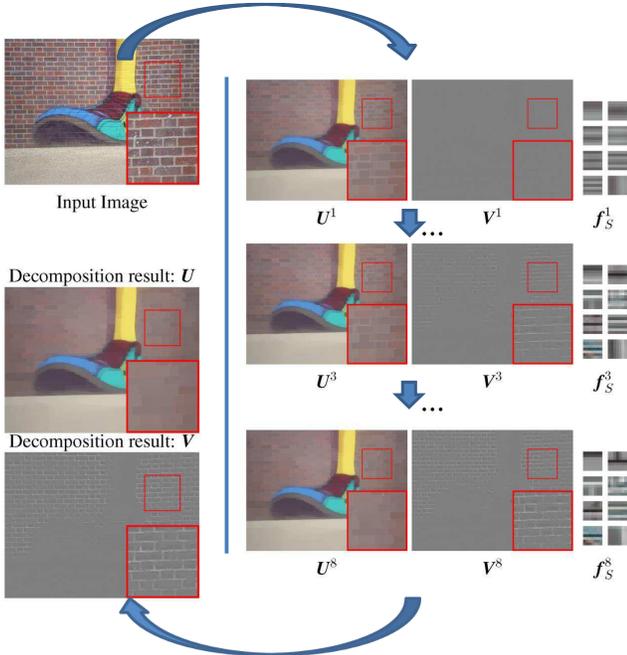


Figure 2. Intermediate results of JCAS for texture-cartoon decomposition. The synthesis dictionary gradually captures the patterns of textures and separates textures from the input image.

The whole procedures of our method are summarized in Algorithm 1. Since all of the three sub-problems involved in our algorithm are convex, each step will not increase the energy of the objective function (5). For our lower bounded objective function (5), the optimization process is guaranteed to converge in terms of energy. We experimentally found that the energy of loss function reduces rapidly. For all the experiments in this paper, we set the maximum number of iterations as 15.

3.4. Discussions

The proposed JCAS model is non-convex. Given the input image and the analysis dictionary, we need to estimate not only the image layers but also the synthesis dictionary. For such a non-convex optimization problem, the initialization and the optimization order of the variables play an important role. In Algorithm 1, we initialize $\{z_{S,j}^0\}_{j=1,\dots,N}$ as an all-zero matrix and solve the \mathbf{u} sub-problem first. The estimation \mathbf{u}^1 provides us a coarse evaluation of the background layer, and the residual image $\mathbf{y} - \mathbf{u}^1$ contains background details as well as repetitive textures. Then, we extract patches in the residual image $\mathbf{y} - \mathbf{u}^1$ and utilize the PCA dictionary to initialize the synthesis dictionary $\{f_{S,j}^1\}_{j=1,\dots,N}$. Having the synthesis dictionary, we are able to get an estimation of the texture layer $\sum_j^N \mathbf{F}_{S,j}^1 z_j^1$ by solving the convolutional sparse coding problem. Due to the sparsity regularization and the constraint on the number of synthesis dictionary atoms, the synthesis approximation $\sum_j^N \mathbf{F}_{S,j} z_j^1$ of the residual image tends to concentrate

on the texture pattern while ignoring the details from background. As a result, the details removed in the previous iteration are still in the residual image $\mathbf{y} - \mathbf{u} - \sum_j^N \mathbf{F}_{S,j} z_j^1$. Such a fact helps us gradually extract the texture layer without over-smoothing the background layer.

Fig. 2 provides some intermediate results of JCAS for texture-cartoon decomposition. In the first iteration, the weak analysis-prior (with the simple gradient operators as the analysis dictionary) provides a coarse estimation of the background. To avoid over-smoothing, a small regularization parameter λ is adopted, and there are still a large amount of textures in \mathbf{u}^1 . Furthermore, with the PCA initialized dictionary, the synthesis component \mathbf{v}^1 is not able to provide a good approximation to the texture. In the following iterations, the synthesis dictionary gradually captures the texture patterns, and $\sum_j^N \mathbf{F}_{S,j} z_j$ extracts the texture layer from the residual $\mathbf{y} - \mathbf{u}$. Since the compact synthesis component focuses only on textures, the image structures in the first iteration of background estimation are still in the residual image. The following iterations will not lose structures but gradually remove textures. As a result, the proposed method is able to remove the repetitive textures (e.g., the brickwork joint) while preserve the illuminance of background layer (e.g., bricks with different colors) unchanged.

4. Experimental Results

In this section, we evaluate the proposed JCAS model on several image decomposition applications, including rain streak removal, tone mapping and others.

4.1. Experiments on Rain Streak Removal

Due to the complex appearance of rain streaks and outdoor background in images, rain streak removal is a challenging image layer separation problem. In the last several years, many models [20, 24, 25, 8, 19] have been proposed to deal with this problem. We compare the proposed method with several rain streak removal algorithms, including frequency domain decomposition method [19], low-rank appearance model (LRA) [8], discriminative sparse coding (DSC) method [25] and layer-prior method (LP) [24]. The code of the LRA algorithm [8] is written by ourselves, while the codes of other competing methods are from the original authors. To validate the effectiveness of joint sparse representation, we also provide the results using a single ASR prior as the baseline.

As introduced in previous sections, to better describe the textures, we learn a synthesis dictionary from the input image during the decomposition process. Specifically, 4 convolutional dictionary atoms of size 7×7 are learned to reconstruct the rain streak layer. Some priors of rain images, including the directional prior and the non-negativeness prior of rain streaks, are further utilized to improve the performance. With the vertical orientation prior [20], we adopt

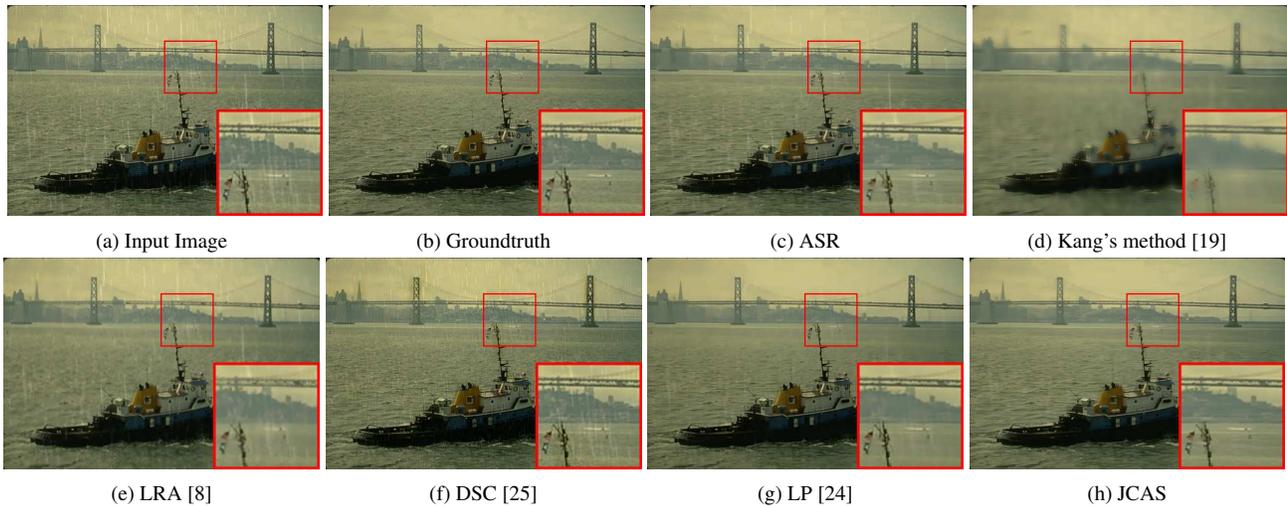


Figure 3. Rain streak removal results by the competing methods.

Table 1. Rain streak removal results (SSIM) of the competing methods on 14 images.

	ASR	Kang's [19]	LRA [8]	DSC [25]	LP [24]	JCAS
1	0.5317	0.5371	0.5683	0.5163	0.5639	0.5769
2	0.6534	0.7294	0.7496	0.5926	0.7677	0.7594
3	0.8099	0.6797	0.7903	0.7895	0.8610	0.8841
4	0.8791	0.7489	0.8437	0.8541	0.9135	0.9359
5	0.7879	0.8077	0.7600	0.7190	0.9154	0.8827
6	0.9425	0.7199	0.8210	0.9413	0.9173	0.9533
7	0.9067	0.5669	0.8624	0.8867	0.8728	0.9079
8	0.9298	0.7124	0.8847	0.9206	0.9435	0.9370
9	0.9410	0.7948	0.9079	0.9274	0.9420	0.9606
10	0.8178	0.7399	0.7955	0.7814	0.8804	0.9088
11	0.8946	0.7027	0.8566	0.8867	0.8979	0.9394
12	0.8254	0.6988	0.7928	0.7701	0.8642	0.8953
13	0.8453	0.5782	0.8248	0.8250	0.8394	0.8959
14	0.8209	0.7303	0.7836	0.7736	0.9076	0.9168
Ave.	0.8275	0.6962	0.8029	0.7559	0.8656	0.8826

the horizontal gradient filters $[-1, 1]$ and $[-1, 0, 1]$ as the analysis dictionary for rain removal application. We incorporate the non-negativeness prior by adding positive constraints on both the synthesis coefficients and dictionary in (5). This prior will not introduce further computation burden. By simply changing the proximal steps in the z and f_S subproblems to their non-negative version, we can get the non-negative estimation.

We perform rain streak removal experiments on 14 synthetic rainy images. The first two images are collected from [19] and the other 12 images are provided by [24]. For each algorithm, the parameters are set the same on all the 14 images. We set the parameters λ and γ in our JCAS model as 0.005 and 0.02. The same parameter $\lambda = 0.005$ is utilized for the baseline ASR method. For the other competing methods, we carefully tuned their parameters for their best performance on the dataset.

We follow the experimental setting in [24] and compare different results in term of structure similarity (SSIM) indexes [37]. The SSIM results are listed in Table 1, where the best results are highlighted in bold. The proposed J-

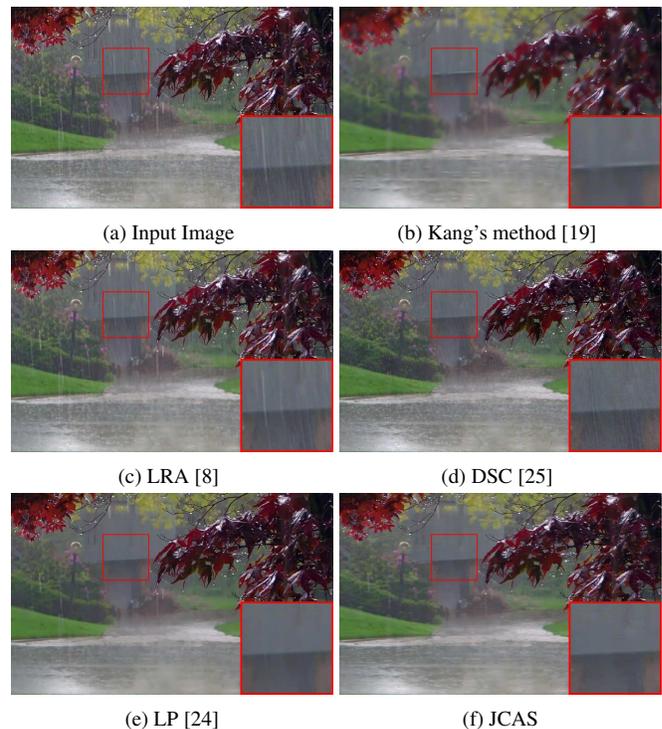


Figure 4. Visual comparison of the competing rain streak removal algorithms on a real rainy image.

CAS algorithm achieves the best results on 11 out of the 14 testing images, and the second best on the others. Furthermore, the higher SSIM index of JCAS over ASR validates the effectiveness of joint ASR and SSR approximation. By extracting repetitive textures from the input image, the synthesis model helps the analysis model to better characterize the latent background.

Fig. 3 shows the rain removal results on a synthetic image. It can be observed that the Kang's method [19] produces an over-smoothed estimation which loses many de-

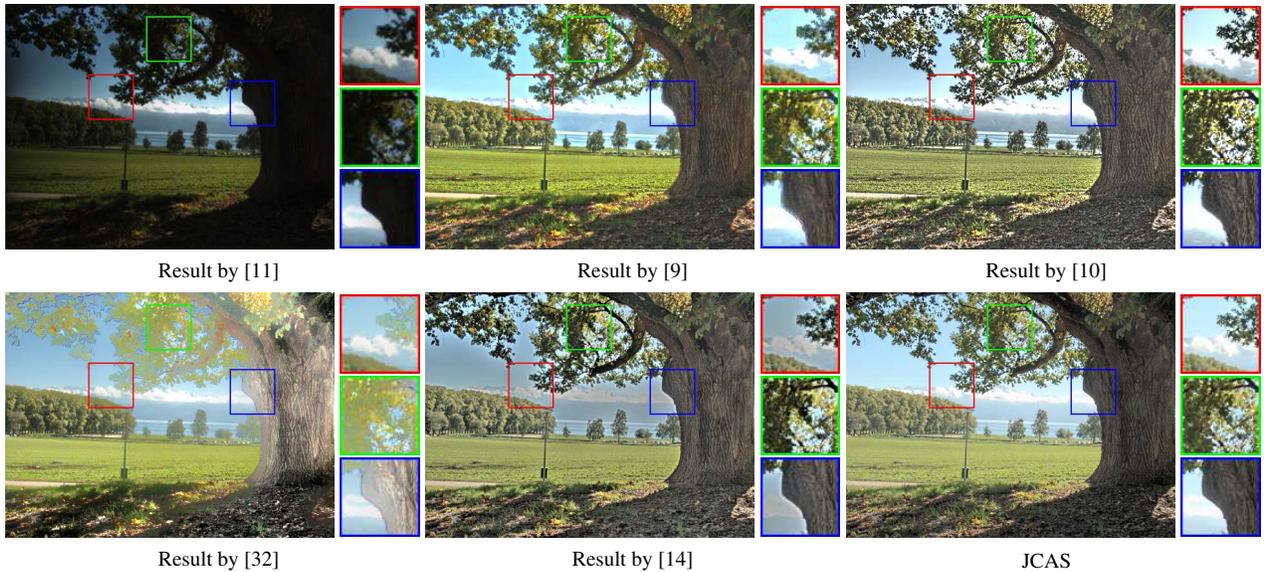


Figure 5. Tone mapping results by the competing methods.

Table 2. Quantitative results (TMQI value [41]) of the compared methods on HDR tone mapping experiments.

	[11]	[9]	[10]	[32]	[14]	JCAS
1	0.7207	0.9477	0.8629	0.8636	0.9288	0.9484
2	0.7714	0.8829	0.7788	0.8050	0.8564	0.8908
3	0.7330	0.9358	0.8897	0.9095	0.9021	0.9541
4	0.7570	0.9480	0.9053	0.8979	0.9343	0.9658
5	0.4792	0.6994	0.8253	0.9490	0.9204	0.8441
6	0.6251	0.8708	0.7973	0.9097	0.8849	0.8950
7	0.8247	0.9855	0.7378	0.9261	0.8880	0.8709
8	0.7542	0.9010	0.8146	0.8254	0.8666	0.9197
9	0.6877	0.9478	0.9558	0.9581	0.8970	0.9731
10	0.8031	0.9717	0.7869	0.9151	0.9314	0.9239
11	0.6480	0.8910	0.8773	0.9567	0.9404	0.9873
12	0.6519	0.8763	0.8521	0.9231	0.9561	0.9621
13	0.3873	0.7127	0.7715	0.9368	0.9219	0.9231
14	0.7555	0.9583	0.9037	0.9203	0.8849	0.8760
15	0.7542	0.9010	0.8200	0.8231	0.8667	0.9177
Ave.	0.6902	0.8953	0.8386	0.9013	0.9053	0.9235

tails in the background. Other competing methods preserve most of the details in the background but remains some streak residuals. In contrast, the proposed JCAS yields a cleaner background estimation with less rain streak residuals. In Fig. 4, we show the results on a real rainy image. The highlight windows clearly show the advantages of the proposed algorithm. It removes more rain streaks and keeps details better in the background layer. More visual comparisons are provided in the supplementary file.

4.2. Experiments on HDR Tone Mapping

Tone mapping aims to reproduce a low dynamic range (LDR) image from its HDR counterpart for display in LDR devices. The details and colors of the HDR image should be preserved in the LDR image. To apply our JCAS algorithm

to the HDR tone mapping problem, we first transform the HDR image into the logarithmic domain, and then adopt our JCAS model to decompose its illuminance component into a base layer U and a detail layer V . The base layer is compressed with a scale factor 0.4, and then added back to the detail layer. A color restoration [11] step is used to reproduce chrominance information.

We compare our method with state-of-the-art tone mapping methods on 15 HDR images provided in [41]. The competing methods include filtering based methods [11, 9, 10] and recently proposed optimization-based methods [32, 14]. The source codes of competing methods are obtained from the original authors. The parameters for each algorithm are set the same on all the 15 images. For our JCAS model, we adopt the first-order and second-order gradients as the analysis dictionary, and train 8 filters as the synthesis dictionary. The regularization parameter λ and γ are fixed as $1E-3$ and $1E-4$, respectively. The tone-mapped image quality index (TMQI) [41] is used to compare different methods quantitatively. From Table 2, one can see that JCAS achieves the highest TMQI values for 9 out of the 15 images, and its average TMQI value is much higher than other competing methods.

Fig. 5 shows some visual examples of the tone mapping results. Methods [11], [9], [10] and [32] lose significant details in the dark or highlight areas. The result of [14] preserves most of the details in the HDR image; however, some halo artifacts appear in the strong edge area (highlighted in the blue window) and the color in the leaves area is not natural. Compared with other methods, our method is able to generate high quality tone mapping results with more details and less artifacts. More visual comparisons are provided

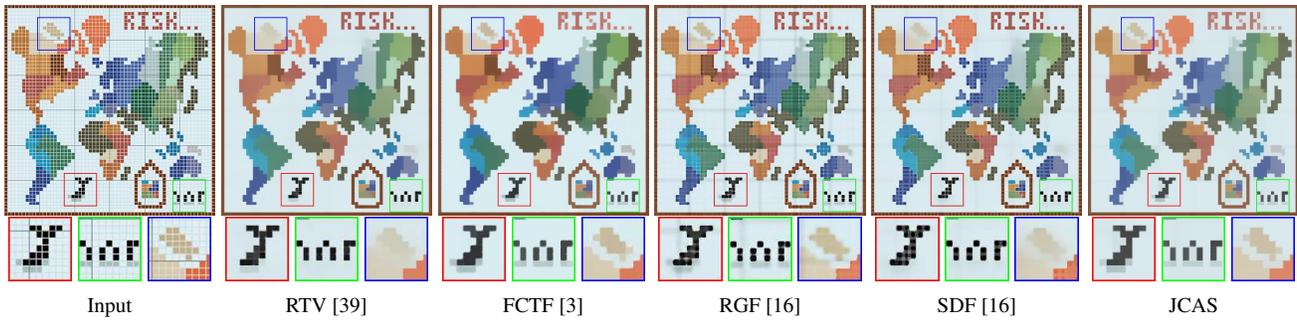


Figure 6. The texture removal results by the competing methods on the *Map* image.

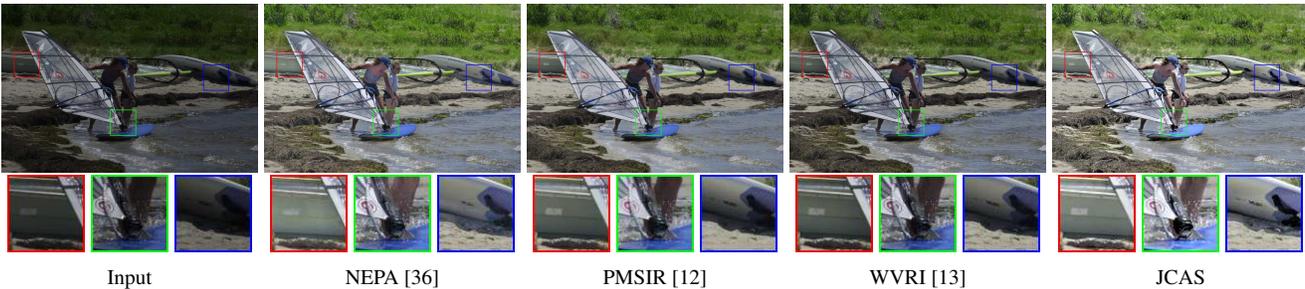


Figure 7. The contrast enhancement results by the competing methods.

in the supplementary file.

4.3. Experiments on Other Applications

The JCAS model can also be used for other applications. Here we test it on a multiplicative decomposition problem, i.e., contrast enhancement, and an additive decomposition problem, i.e., texture-cartoon decomposition. The setting of synthesis and analysis dictionaries for the two applications are exactly the same as that in the HDR tone mapping application. Since there are no widely accepted quantitative measures for these two applications, we only provide visual examples for qualitative evaluation.

For texture-cartoon decomposition, we compare JCAS with relative total variation (RTV) [39], fast cartoon+texture filtering (FCTF) [3] and the recently proposed rolling guidance filter [43] and static and dynamic guidance filtering (SDF) [16] method. The codes of these methods are provided by their authors. We have tried our best to tune their parameters on the testing images for their possibly best performance. Fig. 6 shows an example of texture removal results by different methods. It is easy to see that RTV [39], RGF [43] and SDF [16] fail to remove the white grids in dark area and produce blurry background in low-contrast areas. FCTF [3] and JCAS successfully remove the high-contrast textures. However, the illuminance in the pink island area is deviated more in the result of FCTF [3].

Fig. 7 compares the proposed JCAS model with state-of-the-art contrast enhancement algorithms, including NEPA [36], PMSIR [12] and WVRI [13]. The codes are from the

original authors and we utilize the default parameters. We can see that JCAS generates higher quality enhancement result with more details and more faithful colors. More visual comparisons can be found in the supplementary file.

5. Conclusion

In this study we integrated the ASR and SSR models into a joint convolutional sparse representation (JCAS) framework to deal with the single image layer separation problem. The analysis-component was used to approximate image large-scale structures, while the synthesis-component was used to represent image fine-scale textures. The complementary property of ASR and SSR makes the proposed JCAS be able to effectively extract the image texture layer without over-smoothing the background layer, and it can be used to flexibly model different types of image structures. Our experimental results on rain streak removal, texture-cartoon decomposition and HDR tone mapping validated the generality and effectiveness of the proposed model. The proposed JCAS model is expected to inspire more future investigations on the behaviors of analysis-based and synthesis-based prior modeling methods and to be extended to a wider range of application tasks.

References

- [1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE TSP*, 2006.

- [2] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher. Structure-texture image decomposition modeling, algorithms, and parameter selection. *IJCV*, 2006.
- [3] A. Buades, T. M. Le, J.-M. Morel, and L. A. Vese. Fast cartoon+ texture image filters. *IEEE TIP*, 2010.
- [4] E. J. Candes, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier analysis and applications*, 2008.
- [5] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 2004.
- [6] T. F. Chan and S. Esedoglu. Aspects of total variation regularized ℓ_1 function approximation. *SIAM Journal on Applied Mathematics*, 2005.
- [7] Y. Chen, R. Ranftl, and T. Pock. Insights into analysis operator learning: From patch-based sparse models to higher order mrfs. *IEEE TIP*, 2014.
- [8] Y.-L. Chen and C.-T. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *CVPR*, 2013.
- [9] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In *ACM TOG*, 2002.
- [10] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. In *ACM TOG*, 2008.
- [11] R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range compression. In *ACM TOG*, 2002.
- [12] X. Fu, Y. Liao, D. Zeng, Y. Huang, X.-P. Zhang, and X. Ding. A probabilistic method for image enhancement with simultaneous illumination and reflectance estimation. *IEEE TIP*, 2015.
- [13] X. Fu, D. Zeng, Y. Huang, X.-P. Zhang, and X. Ding. A weighted variational model for simultaneous reflectance and illumination estimation. In *CVPR*, 2016.
- [14] B. Gu, W. Li, M. Zhu, and M. Wang. Local edge-preserving multiscale decomposition for high dynamic range image tone mapping. *IEEE TIP*, 2013.
- [15] S. Gu, W. Zuo, Q. Xie, D. Meng, X. Feng, and L. Zhang. Convolutional sparse coding for image super-resolution. In *ICCV*, 2015.
- [16] B. Ham, M. Cho, and J. Ponce. Robust image filtering using joint static and dynamic guidance. In *CVPR*, 2015.
- [17] S. Hawe, M. Kleinstueber, and K. Diepold. Analysis operator learning and its application to image reconstruction. *IEEE TIP*, 2013.
- [18] K. He, J. Sun, and X. Tang. Guided image filtering. In *ECCV*, 2010.
- [19] L.-W. Kang, C.-W. Lin, and Y.-H. Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE TIP*, 2012.
- [20] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim. Single-image deraining using an adaptive nonlocal means filter. In *IEEE ICIP*, 2013.
- [21] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *NIPS*, 2009.
- [22] E. H. Land and J. J. McCann. Lightness and retinex theory. *JOSA*, 1971.
- [23] Y. Li and M. S. Brown. Single image layer separation using relative smoothness. In *CVPR*, 2014.
- [24] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *CVPR*, 2016.
- [25] Y. Luo, Y. Xu, and H. Ji. Removing rain from a single image via discriminative sparse coding. In *ICCV*, 2015.
- [26] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *ICCV*, 2009.
- [27] Y. Meyer. *Oscillating patterns in image processing and nonlinear evolution equations: the fifteenth Dean Jacqueline B. Lewis memorial lectures*. American Mathematical Soc., 2001.
- [28] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *CVPR*, 2005.
- [29] R. Rubinstein, A. M. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *Proceedings of IEEE*, 2010.
- [30] R. Rubinstein, T. Peleg, and M. Elad. Analysis k-svd: a dictionary-learning algorithm for the analysis sparse model. *IEEE TSP*, 2013.
- [31] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 1992.
- [32] Q. Shan, J. Jia, and M. S. Brown. Globally optimized linear windowed tone mapping. *IEEE transactions on visualization and computer graphics*, 2010.
- [33] J.-L. Starck, E. J. Candès, and D. L. Donoho. The curvelet transform for image denoising. *IEEE TIP*, 2002.
- [34] J.-L. Starck, M. Elad, and D. L. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *IEEE TIP*, 2005.
- [35] S.-H. Sun, S.-P. Fan, and Y.-C. F. Wang. Exploiting image structural similarity for single image rain removal. In *IEEE ICIP*, 2014.
- [36] S. Wang, J. Zheng, H.-M. Hu, and B. Li. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE TIP*, 2013.
- [37] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004.
- [38] B. Wohlberg. Efficient convolutional sparse coding. In *IEEE ICASSP*, 2014.
- [39] L. Xu, Q. Yan, Y. Xia, and J. Jia. Structure extraction from texture via relative total variation. *ACM TOG*, 2012.
- [40] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE TIP*, 2010.
- [41] H. Yeganeh and Z. Wang. Objective quality assessment of tone-mapped images. *IEEE TIP*, 2013.
- [42] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *CVPR*, 2010.
- [43] Q. Zhang, X. Shen, L. Xu, and J. Jia. Rolling guidance filter. In *ECCV*, 2014.