Deep Gaussian Scale Mixture Prior for Spectral Compressive Imaging

Tao Huang[1] Weisheng Dong[1]* Xin Yuan[2]* Jinjian Wu[1] Guangming Shi[1]
[1]School of Artificial Intelligence, Xidian University [2]Bell Labs
thuang_666@stu.xidian.edu.cn wsdong@mail.xidian.edu.cn xyuan@bell-labs.com
jinjian.wu@mail.xidian.edu.cn gmshi@xidian.edu.cn

Abstract

In coded aperture snapshot spectral imaging (CASSI) system, the real-world hyperspectral image (HSI) can be reconstructed from the captured compressive image in a snapshot. Model-based HSI reconstruction methods employed hand-crafted priors to solve the reconstruction problem, but most of which achieved limited success due to the poor representation capability of these hand-crafted priors. Deep learning based methods learning the mappings between the compressive images and the HSIs directly achieved much better results. Yet, it is nontrivial to design a powerful deep network heuristically for achieving satisfied results. In this paper, we propose a novel HSI reconstruction method based on the Maximum a Posterior (MAP) estimation framework using learned Gaussian Scale Mixture (GSM) prior. Different from existing GSM models using hand-crafted scale priors (e.g., the Jeffrey's prior), we propose to learn the scale prior through a deep convolutional neural network (DCNN). Furthermore, we also propose to estimate the local means of the GSM models by the DCNN. All the parameters of the MAP estimation algorithm and the DCNN parameters are jointly optimized through end-to-end training. Extensive experimental results on both synthetic and real datasets demonstrate that the proposed method outperforms existing state-of-the-art methods. The code is available at https://see.xidian.edu.cn/faculty/wsdong/Projects/DGSM-SCI.htm.

1. Introduction

Compared with traditional RGB images, hyperspectral images (HSIs) have more spectral bands and can describe the characteristics of material in the imaged scene more accurately. Relying on its rich spectral information, HSIs are beneficial to many computer vision tasks, *e.g.*, object recognition [33], detection [40] and tracking [34]. The conventional imaging systems with single 1D or 2D sensor require a long time to scan the scene, failing to capture dynamic objects. Recently, many coded aperture snapshot spectral imaging (CASSI) systems [8, 22, 23, 35] have been proposed to capture the 3D HSIs at video rate. CASSI utilizes a physical mask and a disperser to modulate different wavelength signals, and mixes all modulated signals to generate a single 2D compressive image. Then a reconstruction algorithm is employed to reconstruct the 3D HSI from the 2D compressive image. As shown in Fig. 1, 28 spectral bands have been reconstructed from a 2D compressive image (measurement) captured by a real CASSI system [22].

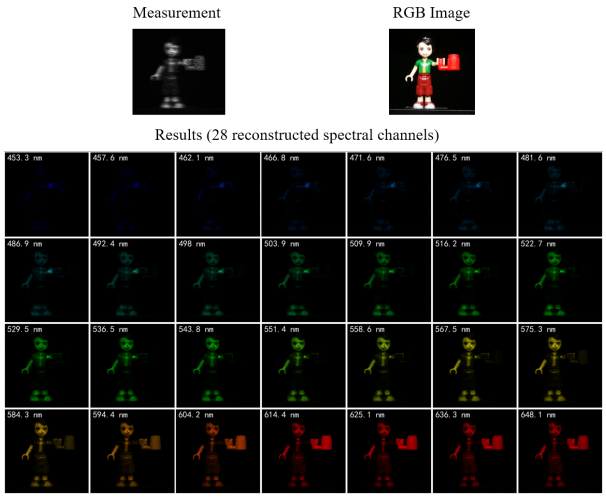

Figure 1. A single shot measurement captured by [22] and 28 reconstructed spectral channels using our proposed method.

Therefore, reconstruction algorithms play a pivot role in CASSI. To solve this ill-posed inverse problem, previous model-based methods adopted hand-crafted priors to regularize the reconstruction process. In GAP-TV [43], the total variation prior was introduced to solve the HSI reconstruction problem. Based on the assumption that HSIs have sparse representations with respective to some dictionaries, sparse-based methods [14, 17, 35] exploited the ℓ_1 sparsity to regularize the solution. Considering that the pixels of

* Corresponding authors.

HSIs have strong long-range dependence, non-local based methods [18, 38, 48] have also been proposed. However, the model-based methods have to tweak parameters manually, resulting in limited reconstruction quality in addition to the slow reconstruction speed. Inspired by the successes of deep convolutional neural networks (DCNNs) for natural image restoration [16, 47], deep learning based HSI reconstruction methods [3, 36, 37] have also been proposed. In [36], an iterative HSI reconstruction algorithm was unfolded into a DCNN, where two sub-networks were used to exploit the spatial-spectral priors. In [37] the nonlocal self-similarity prior has also been incorporated to further improve the results. In addition to the optimization-inspired methods, DCNN-based methods [22, 24, 41] that learned the mapping functions between the 2D measurements and the 3D HSIs directly have also been proposed. λ-net [24] reconstructed the HSIs from the inputs of 2D measurements and the mask through a two-stage DCNN. TSA-Net [22] integrated three spatial-spectral self-attention modules in the backbone U-Net [31] and achieved state-of-the-art results. Although promising HSI reconstruction performance has been achieved, it is non-trivial to design a powerful DCNN heuristically.

Bearing the above concerns in mind, in this paper, we propose an interpretable HSI reconstruction method with learned Gaussian Scale Mixture (GSM) prior. The contributions of this paper are listed as follows.

- Learned GSM models are proposed to exploit the spatial-spectral correlations of HSIs. Unlike the existing GSM models with hand-crafted scale priors (e.g., Jeffrey's prior), we propose to learn the scale prior by a DCNN.
- The local means of the GSM models are estimated as a weighted average of the spatial-spectral neighboring pixels. The spatial-spectral similarity weights are also estimated by the DCNN.
- The HSI reconstruction problem is formulated as a Maximum a Posteriori (MAP) estimation problem with the learned GSM models. All the parameters in the MAP estimator are jointly optimized in an end-to-end manner.
- Extensive experimental results on both synthetic and real datasets show that the proposed method outperforms existing state-of-the-art HSI reconstruction methods.

2. Related Work

Hereby, we briefly review the conventional model-based HSI reconstruction methods, the recently proposed deep learning-based HSI reconstruction methods and the GSM models for signal modeling.

2.1. Conventional model-based HSI reconstruction methods

Reconstructing the 3D HSI from the 2D compressive image is the core of CASSI system and usually with the help of various hand-crafted priors. In [7] gradient projection algorithms were proposed to solve the sparse HSI reconstruction problems. In [17] dictionary learning based sparse regularizers have been employed for HSI reconstruction. In [1, 14, 43] total variation (TV) regularizers have also been adopted to suppress the noise and artifacts. In [18], the nonlocal self-similarity and the low-rank property of HSIs have been exploited, leading to superior HSI reconstruction performance. The major drawbacks of these model-based methods are that they are time-consuming and need to select the parameters manually.

2.2. Deep learning-based HSI reconstruction

Due to the powerful learning ability, deep neural networks treating the HSI reconstruction as a nonlinear mapping problem have achieved much better results than model-based methods. In [41] initial estimates of the HSIs were first obtained by the method of [1] and were further refined by a DCNN. λ-net [24] reconstructed the HSIs through a two-stage procedure, where the HSIs were first initially reconstructed by a Generative Adversarial Network (GAN) with self-attention, followed by a refinement stage for further improvements. In [22], DCNN with spatial-spectral self-attention modules was proposed to exploit the spatial-spectral correlation, leading to state-of-the-art performance. Instead of designing the DCNN heuristically, DCNNs based on unfolding optimization-based HSI reconstruction algorithms have also been proposed [21]. In [36] a HSI reconstruction algorithm with a denoising prior was unfolded into a deep neural network. Since the spatial-spectral prior has not been fully exploited, the method of [36] achieved limited success. To exploit the nonlocal self-similarity of HSIs, the nonlocal sub-network has also been integrated into the deep network proposed in [37], leading to further improvements. The other line of work is to apply deep denoiser into the optimization algorithm, leading to a plug-and-play framework [49].

2.3. GSM models for signal modeling

As a classical probability model, the Gaussian Scale Mixture (GSM) model has been used for various image restoration tasks. In [27] the GSM model was utilized to characterize the distributions of the wavelet coefficients for image denoising. In [5] the GSM model has been proposed to model the sparse codes for simultaneous sparse coding with applications to image restoration. In [26, 32] the GSM models have also been used to model the moving objects of videos for foreground estimation, achieving state-of-the-art performance. In this paper, we propose to character-

ize distributions of the HSIs with the GSM models for HSI reconstruction. Different from existing GSM models with manually selected scale priors, we propose to learn both the scale prior and local means of the GSM models with DCNNs. Through end-to-end training, all the parameters are learned jointly.

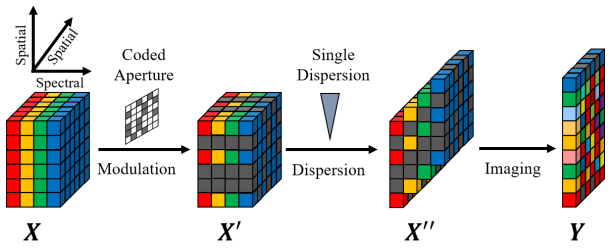

Figure 2. The imaging schematic of CASSI system.

3. The CASSI Observation Model

As shown in Fig. 2, the 3D HSI is encoded into the 2D compressive image by the CASSI system. In the CASSI system, the 3D spectral data cube is first modulated spatially by a coded aperture (i.e., a physical mask). Then, the following dispersive prism disperses each wavelength of the modulated data. A 2D imaging sensor captures the dispersed data and outputs a 2D measurement which mixes the information of all wavelengths.

Let $\mathbf{X} \in \mathbb{R}^{H\times W\times L}$ denote the 3D spectral data cube and $\mathbf{C} \in \mathbb{R}^{H\times W}$ denote the physical mask. The l^{th} wavelength of the modulated image can thus be represented as

$$\mathbf{X}_l^{'} = \mathbf{C} \odot \mathbf{X}_l, \tag{1}$$

where $\mathbf{X}^{'} \in \mathbb{R}^{H\times W\times L}$ is the 3D modulated image and $\odot$ denotes the element-wise product. In CASSI system, the modulated image is dispersed by the dispersive prism. In other words, each channel of the tensor $\mathbf{X}^{'}$ will be shifted spatially and the shifted tensor $\mathbf{X}^{''} \in \mathbb{R}^{H\times(W+L-1)\times L}$ can be written as

$$\mathbf{X}^{''}(r,c,l) = \mathbf{X}^{'}(r,c+d_l,l), \tag{2}$$

where d_l denotes the shifed distance of the l^{th} channel, $1 \le r \le H, 1 \le c \le W$ and $1 \le l \le L$. At last, the 2D imaging sensor captures the shifted image into a 2D measurement (by compressing the spectral domain), as

$$\mathbf{Y} = \sum_{l=1}^{L} \mathbf{X}_l^{''}, \tag{3}$$

where $\mathbf{Y} \in \mathbb{R}^{H\times(W+L-1)}$ represents the 2D measurement. As such, the matrix-vector form of Eq. (3) can be formulated as

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x}, \tag{4}$$

where $\boldsymbol{x} \in \mathbb{R}^N$ and $\boldsymbol{y} \in \mathbb{R}^M$ denote the vectorized form of $\mathbf{X}$ and $\mathbf{Y}$ respectively, $N = HWL$ and $M = H(W + L-1)$, and $\mathbf{A} \in \mathbb{R}^{M\times N}$ denotes the measurement matrix of the CASSI system, implemented by the coded aperture and disperser. Considering the measurement noise $\boldsymbol{n} \in \mathbb{R}^M$, the forward model of CASSI is now

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x} + \boldsymbol{n}. \tag{5}$$

The theoretical performance bounds of CASSI have been derived in [12].

4. The Proposed Method

4.1. GSM models for CASSI

We formulate the HSI reconstruction as a maximum a posteriori (MAP) estimation problem. Given the observed measurement $\boldsymbol{y}$, the desired 3D HSI $\boldsymbol{x}$ can be estimated by maximizing the posterior

$$\log p(\boldsymbol{x}|\boldsymbol{y}) \propto \log p(\boldsymbol{y}|\boldsymbol{x}) + \log p(\boldsymbol{x}), \tag{6}$$

where $p(\boldsymbol{y}|\boldsymbol{x})$ is the likelihood term and $p(\boldsymbol{x})$ is the (to be determined) prior distribution of $\boldsymbol{x}$. The likelihood term is generally modeled with a Gaussian function as

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{||\boldsymbol{y}-\mathbf{A}\boldsymbol{x}||_2^2}{2\sigma^2}\right). \tag{7}$$

For the prior term $p(\boldsymbol{x})$, we propose to characterize each pixel x_i of the HSI with a *nonzero-mean* Gaussian distribution of standard deviation θ_i. With a scale prior $p(\theta_i)$ and the assumption that θ_i and x_i are independent, we can model $\boldsymbol{x}$ with the following GSM model

$$p(\boldsymbol{x}) = \prod_i p(x_i), \quad p(x_i) = \int_0^\infty p(x_i|\theta_i)p(\theta_i)d\theta_i, \tag{8}$$

where $p(x_i|\theta_i)$ is a nonzero-mean Gaussian distribution with variance θ_i^2 and mean u_i, i.e.,

$$p(x_i|\theta_i) = \frac{1}{\sqrt{2\pi}\theta_i} \exp(-\frac{(x_i-u_i)^2}{2\theta_i^2}). \tag{9}$$

With different scale priors, the GSM model can well express many distributions.

Regarding the scale prior $p(\theta_i)$, instead of modeling $p(\theta_i)$ with an exact prior (e.g., the Jeffrey's prior $p(\theta_i) = \frac{1}{\theta_i}$), we introduce a general form as

$$p(\theta_i) \propto \exp(-J(\theta_i)), \tag{10}$$

where the $J(\theta_i)$ is an energy function. Instead of computing an analytical expression of $p(x_i)$ that is often intractable, we propose to jointly estimate $\boldsymbol{x}$ and $\boldsymbol{\theta}$ by replacing $p(\boldsymbol{x})$ with $p(\boldsymbol{x},\boldsymbol{\theta})$ in the MAP estimator. This is

$$\begin{aligned}(\boldsymbol{x},\boldsymbol{\theta}) &= \operatorname{argmax}_{\boldsymbol{x},\boldsymbol{\theta}} \log p(\boldsymbol{y}|\boldsymbol{x}) + \log p(\boldsymbol{x},\boldsymbol{\theta}) \\ &= \operatorname*{argmax}_{\boldsymbol{x},\boldsymbol{\theta}} \log p(\boldsymbol{y}|\boldsymbol{x}) + \log p(\boldsymbol{x}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}).\end{aligned} \tag{11}$$

By substituting the Gaussian likelihood term of Eq. (7) and the prior terms of $p(x_i|\theta_i)$ and $p(\theta_i)$ into the above MAP estimator, we can obtain the following objective function

$$\begin{aligned}(\boldsymbol{x},\boldsymbol{\theta}) &= \underset{\boldsymbol{x},\boldsymbol{\theta}}{\operatorname{argmin}} \, ||\boldsymbol{y}-\mathbf{A}\boldsymbol{x}||_2^2 + \sigma^2 \sum_{i=1}^{N} \frac{1}{\theta_i^2}(x_i-u_i)^2 \\ &\quad + 2\sigma^2 \textstyle\sum_{i=1}^{N} \log\theta_i + 2\sigma^2 J(\boldsymbol{\theta}) \\ &= \underset{\boldsymbol{x},\boldsymbol{\theta}}{\operatorname{argmin}} \, ||\boldsymbol{y}-\mathbf{A}\boldsymbol{x}||_2^2 + \sigma^2 \sum_{i=1}^{N} \frac{1}{\theta_i^2}(x_i-u_i)^2 \\ &\quad + R(\boldsymbol{\theta}), \end{aligned} \tag{12}$$

where $R(\boldsymbol{\theta}) = 2\sigma^2 \sum_{i=1}^{N} \log\theta_i + 2\sigma^2 J(\boldsymbol{\theta})$. Thereby, the HSI reconstruction problem can be solved by alternating optimizing $\boldsymbol{x}$ and $\boldsymbol{\theta}$.

For the $\boldsymbol{x}$-subproblem, with fixed $\boldsymbol{\theta}$, we can solve $\boldsymbol{x}$ by solving

$$\boldsymbol{x} = \operatorname{argmin}_{\boldsymbol{x}} \, ||\boldsymbol{y}-\mathbf{A}\boldsymbol{x}||_2^2 + \textstyle\sum_{i=1}^{N} w_i(x_i-u_i)^2, \tag{13}$$

where $w_i = \frac{\sigma^2}{\theta_i^2}$ and the mean u_i keeps updating with $\boldsymbol{x}$. Inspired by the auto-regressive (AR) model [6], we can calculate the weighted average of the local spatial-spectral neighboring pixels as the estimation of the mean u_i, *i.e.*,

$$u_i = \boldsymbol{k}_i^\top \boldsymbol{x}_i, \tag{14}$$

where $\boldsymbol{k}_i \in \mathbb{R}^{q^3}$ denotes the vectorized 3D filter of size $q \times q \times q$ for x_i and $\boldsymbol{x}_i \in \mathbb{R}^{q^3}$ represents the local spatial-spectral neighboring pixels of x_i. For the 3D filters, some existing methods (e.g., the guided filtering [9, 15], the non-local means methods [2, 4] or the deep learning based method [25]) can be used to estimate the spatially-variant filters.

To solve Eq. (13), we employ gradient descent as

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - 2\delta\{\mathbf{A}^\top(\mathbf{A}\boldsymbol{x}^{(t)}-\boldsymbol{y}) + \boldsymbol{w}^{(t)}(\boldsymbol{x}^{(t)}-\boldsymbol{u}^{(t)})\}, \tag{15}$$

where $\boldsymbol{u}^{(t)} = [u_1^t, \cdots, u_N^t]^\top \in \mathbb{R}^N$, $\boldsymbol{w}^{(t)} = [w_1^t, \cdots, w_N^t]^\top \in \mathbb{R}^N$ and δ is the step size.

The $\boldsymbol{\theta}$-subproblem can be changed to estimate $\boldsymbol{w}$. With fixed $\boldsymbol{x}$, $\boldsymbol{w}$ can be estimated by

$$\boldsymbol{w} = \operatorname{argmin}_{\boldsymbol{w}} \textstyle\sum_{i=1}^{N} w_i(x_i-u_i)^2 + R(\boldsymbol{w}). \tag{16}$$

The solution of $\boldsymbol{w}$ depends on $R(\boldsymbol{w})$ being used. For some priors, a closed-form solution can be achieved [26]; for others, iterative algorithms might be used. However, each of them has their pros and cons. To cope with this challenge, hereby instead of using a manually designed proximal operator, we propose to estimate $\boldsymbol{w}^{(t+1)}$ from $\boldsymbol{x}^{(t+1)}$ using a DCNN as will be described in the next subsection.

4.2. Deep GSM for CASSI

In general, alternating computing $\boldsymbol{x}$ and $\boldsymbol{w}$ requires numerous iterations to converge and it is necessary to impose a hand-crafted prior of $p(\boldsymbol{\theta})$. Moreover, all the algorithm parameters and the 3D filters cannot be jointly optimized. To address these issues, we propose to optimize $\boldsymbol{x}$ and $\boldsymbol{w}$ jointly by a DCNN. For network design purpose, we re-bridge the $\boldsymbol{x}$ and $\boldsymbol{w}$-subproblems via a united framework

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - 2\delta\{\mathbf{A}^\top(\mathbf{A}\boldsymbol{x}^{(t)}-\boldsymbol{y}) + \mathcal{S}(\boldsymbol{x}^{(t)})(\boldsymbol{x}^{(t)}-\boldsymbol{u}^{(t)})\}, \tag{17}$$

where $\mathcal{S}(\cdot)$ represents the function of the DCNN for estimating $\boldsymbol{w}$, *i.e.*, the solution of (16). As shown in Fig. 3(a), we construct the end-to-end network with T stages corresponding to T iterations for iteratively optimizing $\boldsymbol{x}$ and $\boldsymbol{w}$. The proposed network consists of the following main modules.

- The measurement $\boldsymbol{y}$ is split into a 3D data cube of size $H \times W \times L$ to initialize $\boldsymbol{x}$.
- We use two sub-networks to learn the measurement matrix $\mathbf{A}$ and its transposed version $\mathbf{A}^\top$.
- For estimating $\boldsymbol{w}$, we develop a lightweight variant of U-Net and a weight generator to learn the function $\mathcal{S}(\cdot)$.
- Instead of using a manually designed method to learn the 3D filters, we utilize the same lightweight U-Net and a 3D filter generator to generate the spatially-variant filters. According to Eq. (14), we filter the current $\boldsymbol{x}$ by the generated 3D filters for updating the means $\boldsymbol{u}$.

4.3. Network Architecture

Considering that the real system has large spatial size of the mask and measurements (e.g., the mask and measurements of [22] are 660×660 and 660×714), the network training with explicitly constructed $\mathbf{A}$ and $\mathbf{A}^\top$ requires a large amount GPU memory and computational complexity. To address this issue, we propose to learn these two operations with two sub-networks.

The modules for learning the measurement matrix A and $\mathbf{A}^\top$. Learning $\mathbf{A}$ and $\mathbf{A}^\top$ with sub-networks allows one to train them on small patches (e.g., 64×64 or 96×96) that can greatly reduce memory consumption and computational complexity. Furthermore, we can train a sub-network to learn multiple masks such that the trained network can work well on multiple imaging systems. The measurement matrix $\mathbf{A}$ represents a hybrid operator of modulation, *i.e.*, shifting and summation, which can be implemented by two Conv layers and four ResBlocks followed by shifting and

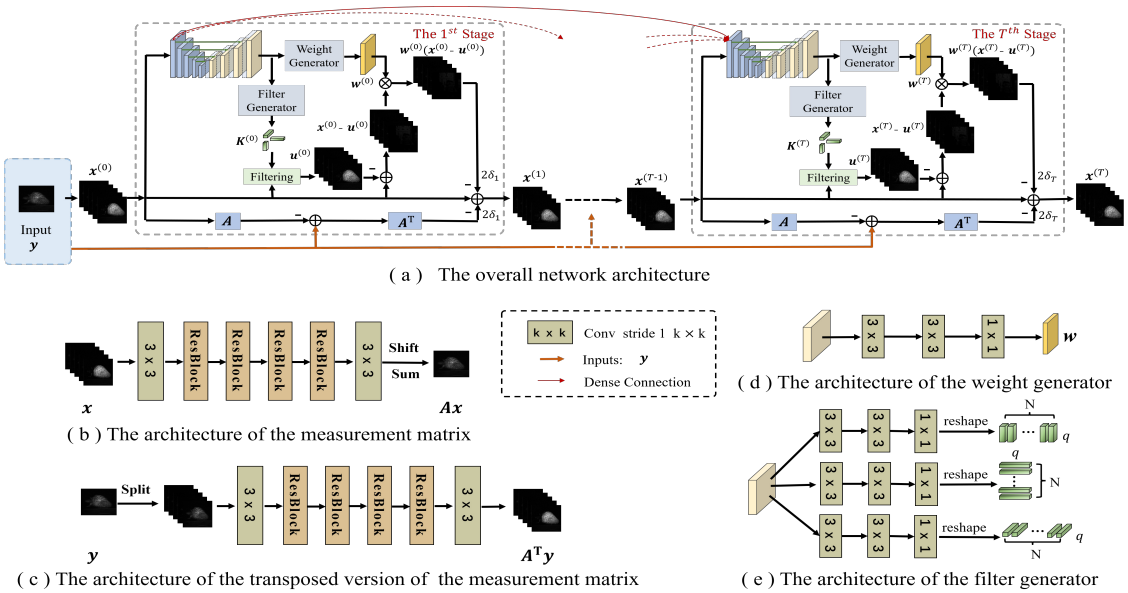

Figure 3. Architecture of the proposed network for hyperspectral image reconstruction. The architectures of (a) the overall network, (b) the measurement matrix, (c) the transposed version of the measurement matrix, (d) the weight generator, and (e) the filter generator.

summation operations. As shown in Fig. 3(b), $\boldsymbol{x}$ is fed into the sub-network to generate modulated feature maps that are further shifted and summed along the spectral dimension to generate the measurements $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}$. Each ResBlock [11] consists of 2 Conv layers with a ReLU nonlinearity function plus a skip connection. Regarding $\mathbf{A}^\top$, as shown in Fig. 3(c), we first slide a $H \times W$ extraction window on the input $\boldsymbol{y}$ of size $H \times (W + L - 1)$ with the slide step one pixel and split the input into L-channel image of size $H \times W$. Then the split sub-images are fed into two Conv layers and four ResBlocks to generate the estimate $\mathbf{A}^\top \boldsymbol{y}$.

The module for estimating the regularization parameters $\boldsymbol{w}$. As shown in the Fig. 3 (a), we propose a lightweight U-Net consisting of five encoding blocks (EBs) and four decoding blocks (DBs) to estimate the weights $\boldsymbol{w}^{(t)}$ from the current estimate $\boldsymbol{x}^{(t)}$. Each EB and DB contains two Conv layers with ReLU nonlinearity function. The average pooling layer with a stride of 2 is inserted between every two neighboring EBs to downsample the feature maps and a bilinear interpolation layer with a scaling factor 2 is adopted ahead of every DB to increase the spatial resolutions of the feature maps. We have noticed that the average pooling works better than max pooling in our problem and the bilinear interpolation plays an important role in DBs. 3×3 Conv filters are used in all the Conv layers. The channel numbers of the output features of the 5 EBs and 4 DBs are set to 32, 64, 64, 128, 128, 128, 64, 64 and 32, respectively. To alleviate the gradient vanishing problem, the feature maps of the first EB are connected to first EB of the U-net of the subsequent stages. The feature maps of the last DB are fed into a weight generator that contains 2 3×3 Conv layers to generate the weights $\boldsymbol{w}$ as shown in Fig. 3 (d). Some weight maps $\boldsymbol{w}$ of two HSIs estimated in the fourth stage are visualized (with normalization) in Fig. 4. From Fig. 4, we can see that $\boldsymbol{w}$ vary spatially and are consistent with the image edges and textures. Aided by this well-learned $\boldsymbol{w}$, the proposed method will pay attentions to the edges and textures.

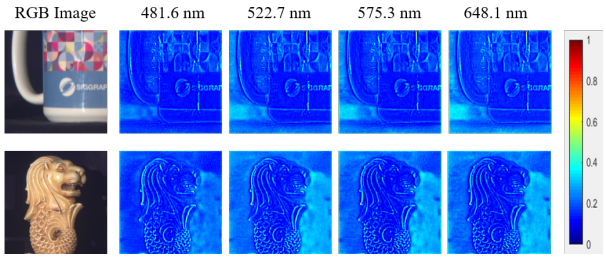

Figure 4. The visualization of the regularization parameters $\boldsymbol{w}$ estimated in the 4-th stage. Left: the corresponding RGB image; right: the $\boldsymbol{w}$ images associated with the four spectral bands.

The module for estimating the local means $\boldsymbol{u}$. We estimate the means of GSM models following Eq. (14). To estimate the spatial-variant 3D filters, we add a filter generator with the input of the feature maps generated by the U-net, as shown in Fig. 3(a). Estimating the spatially adaptive 3D filters has advantages in adapting to local HSI edges and texture structures. However, directly generating these 3D filters will cost a large amount of GPU memory that is unaffordable. To reduce the GPU memory consumption, we propose to factorize each 3D filter into three 1D filters, expressed as

$$\mathbf{K}_i = \boldsymbol{r}_i \otimes \boldsymbol{c}_i \otimes \boldsymbol{s}_i, \tag{18}$$

where $\mathbf{K}_i \in \mathbb{R}^{q\times q\times q}$ denotes the 3D filter, $\boldsymbol{r}_i \in \mathbb{R}^q$, $\boldsymbol{c}_i \in \mathbb{R}^q$ and $\boldsymbol{s}_i \in \mathbb{R}^q$ denote the three 1D filters corresponding to the three dimensions, respectively, and $\otimes$ denotes the tensor product. In this way, filtering the local neighbors $\mathbf{X}_i$ with the 3D filter $\mathbf{K}_i$ can be transformed into convoluting the local neighbors with the three 1D filters along three dimensions in sequence. By factorizing each 3D filter into three 1D filters we can reduce the number of filter coefficients from $N \cdot q^3$ to $3 \cdot N \cdot q$, and thus significantly reduce the GPU memory cost and the computational complexity. As shown in Fig. 3(e), the filter generator contains three branches to learn the 1D filters, respectively. After generating the filters, we can compute the means of GSM models following Eq. (14).

4.4. Network training

We jointly learn the network parameters $\boldsymbol{\Theta}$ through end-to-end training. Except the step size δ, all the network parameters of each stage are shared. All the parameters are optimized by minimizing the following loss function

$$\hat{\boldsymbol{\Theta}} = \text{argmin}_{\boldsymbol{\Theta}} \frac{1}{D}\sum_{d=1}^{D} \|\mathcal{F}(\boldsymbol{y}_d;\boldsymbol{\Theta}) - \boldsymbol{x}_d\|_1, \qquad (19)$$

where D denotes the total number of the training samples, $\mathcal{F}(\boldsymbol{y}_d;\boldsymbol{\Theta})$ represents the output of the proposed network given d^{th} measurement $\boldsymbol{y}_d$ and the network parameters $\boldsymbol{\Theta}$, and $\boldsymbol{x}_d$ is the ground-truth HSI. The ADAM optimizer [13] with setting $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ is exploited to train the proposed network. We set the learning rate as 10^{-4}. The parameters of the convolutional layers are initialized by the Xavier initialization [10]. We implement the proposed method in PyTorch and train the network using a single Nvidia Titan XP GPU. Instead of using the ℓ_2 norm in the loss function, here we use the ℓ_1 norm that has been proved to be better in preserving image edges and textures.

5. Simulation Results

5.1. Experimental Setup

To verify the effectiveness of the proposed HSI reconstruction method for CASSI, we conduct simulations on two public HSI datasets CAVE [42] and KAIST [3]. The CAVE dataset consists of 32 HSIs of spatial size 512×512 with 31 spectral bands. The KAIST dataset has 30 HSIs of spatial size 2704×3376 also with 31 spectral bands. Similar to TSA-Net [22], we employ the *real mask* of size 256×256 for simulation. Following the procedure in TSA-Net [22], the CAVE dataset is used for network training, and 10 scenes of spatial size 256×256 from the KAIST dataset are extracted for testing. To be consistent with the wavelength of the real system [22], we unify the wavelength of the training and testing data by spectral interpolation. Thus, the modified training and testing data have 28 spectral bands ranging from 450nm to 650nm.

During training, to simulate the measurements, we first randomly extract $96 \times 96 \times 28$ patches from the training dataset as training labels (ground truth HSI) and randomly extract 96×96 patches from the *real mask* to generate the modulated data. Then the modulated data is shifted in spatial at an interval of two pixels. The spectral dimension of the shifted data is summed up to generate the 2D measurements of size 96×150 as the network inputs. We use Random flipping and rotation for data argumentation. The peak-signal-to-noise (PSNR) and the structural similarity index (SSIM) [39] are both employed to evaluate the performance of the HSI reconstruction methods.

5.2. Comparison with State-of-the-Art Methods

We compare the proposed HSI reconstruction method with several state-of-the-art methods, including three model-based methods (i.e., TwIST [1], GAP-TV [43] and DeSCI [18]) and four deep learning based methods (i.e., λ-net [24], HSSP [36], DNU [37] and TSA-Net [22]). As the source codes are unavailable, we re-implemented HSSP and DNU by ourselves. For other competing methods, we use the source codes released by their authors. For the sake of fair comparison, all deep learning methods were *re-trained on the same training dataset*. Table 1 shows the reconstruction results of these testing methods on the 10 scenes, where we can see that the deep learning-based methods outperform the model-based methods. The proposed method outperforms other deep learning-based methods by a large margin. Specifically, our method outperforms the second best method TSA-Net by 1.17dB in average PSNR and 0.0227 in average SSIM. Compared with the two deep unfolding methods HSSP and DNU, the improvements by the proposed method over HSSP [36] and DNU [37] are 2.28 dB and 1.89 dB in average, respectively. The HSSP and DNU methods also tried to learn the spatial-spectral correlations of HSIs by two sub-networks without emphasizing image edges and textures. By contrast, we propose to learn the spatial-spectral prior of HSIs by the spatially-adaptive GSM models characterized by the learned local means and variances. The learned GSM models have advantages in adapting to various HSI edges and textures. Fig. 5 plots selected frames and spectral curves of the reconstructed HSIs by the five deep learning-based methods. We can see that the HSIs reconstructed by the proposed method have more edge details and less undesirable visual artifacts than the other methods. The RGB images of the 10 scenes and more visual comparison results are shown in the supplementary material (SM).

5.3. Multiple Mask Results

As mentioned before, our proposed network is robust to mask due to the learning of $\mathbf{A}$ and $\mathbf{A}^\top$. To verify this, we conducted experiments on compound training and testing

Table 1. The PSNR in dB (left entry in each cell) and SSIM (right entry in each cell) results of the test methods on 10 scenes.

Method	TwIST [1]	GAP-TV [43]	DeSCI [18]	λ-net [24]	HSSP [36]	DNU [37]	TSA-Net [22]	Ours
Scene1	25.16, 0.6996	26.82, 0.7544	27.13, 0.7479	30.10, 0.8492	31.48, 0.8577	31.72, 0.8634	32.03, 0.8920	**33.26, 0.9152**
Scene2	23.02, 0.6038	22.89, 0.6103	23.04, 0.6198	28.49, 0.8054	31.09, 0.8422	31.13, 0.8464	31.00, 0.8583	**32.09, 0.8977**
Scene3	21.40, 0.7105	26.31, 0.8024	26.62, 0.8182	27.73, 0.8696	28.96, 0.8231	29.99, 0.8447	32.25, 0.9145	**33.06, 0.9251**
Scene4	30.19, 0.8508	30.65, 0.8522	34.96, 0.8966	37.01, 0.9338	34.56, 0.9018	35.34, 0.9084	39.19, 0.9528	**40.54, 0.9636**
Scene5	21.41, 0.6351	23.64, 0.7033	23.94, 0.7057	26.19, 0.8166	28.53, 0.8084	29.03, 0.8326	**29.39, 0.8835**	28.86, 0.8820
Scene6	20.95, 0.6435	21.85, 0.6625	22.38, 0.6834	28.64, 0.8527	30.83, 0.8766	30.87, 0.8868	31.44, 0.9076	**33.08, 0.9372**
Scene7	22.20, 0.6427	23.76, 0.6881	24.45, 0.7433	26.47, 0.8062	28.71, 0.8236	28.99, 0.8386	30.32, 0.8782	**30.74, 0.8860**
Scene8	21.82, 0.6495	21.98, 0.6547	22.03, 0.6725	26.09, 0.8307	30.09, 0.8811	30.13, 0.8845	29.35, 0.8884	**31.55, 0.9234**
Scene9	22.42, 0.6902	22.63, 0.6815	24.56, 0.7320	27.50, 0.8258	30.43, 0.8676	31.03, 0.8760	30.01, 0.8901	**31.66, 0.9110**
Scene10	22.67, 0.5687	23.10, 0.5839	23.59, 0.5874	27.13, 0.8163	28.78, 0.8416	29.14, 0.8494	29.59, 0.8740	**31.44, 0.9247**
Average	23.12, 0.6694	24.36, 0.6993	25.27, 0.7207	28.53, 0.8406	30.35, 0.8524	30.74, 0.8631	31.46, 0.8939	**32.63, 0.9166**

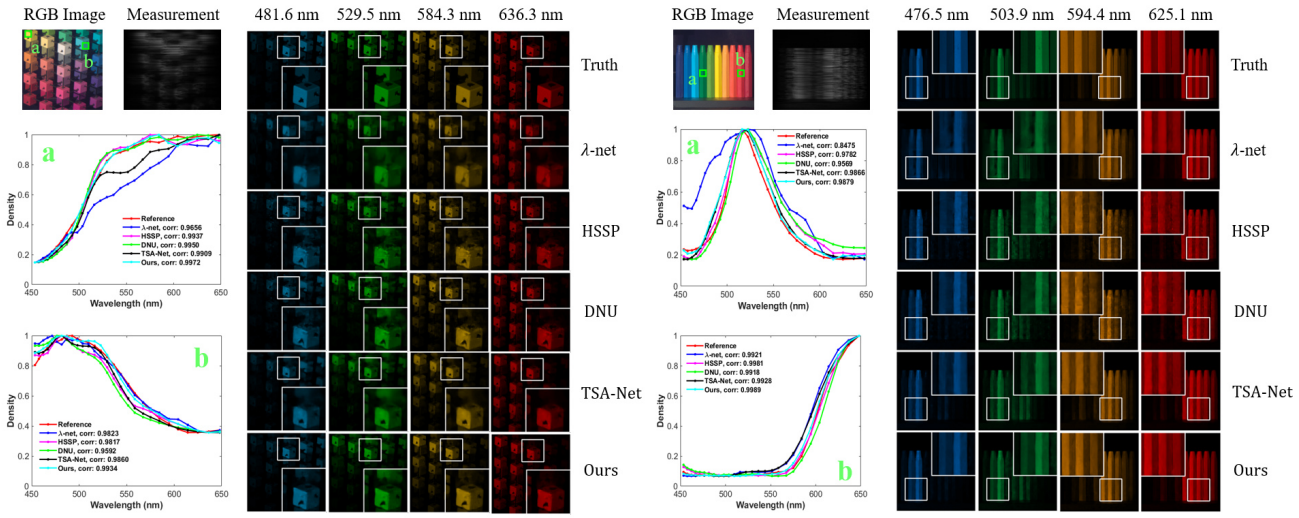

Figure 5. Reconstructed images of *Scene 2* (left) and *Scene 9* (right) with 4 out of 28 spectral channels by the five deep learning-based methods. Two regions in each scene are selected for analysing the spectra of the reconstructed results. Zoom in for better view.

Table 2. The average PSNR (left) and SSIM (right) results with five masks by the competing methods.

Method	DNU [37]	TSA-Net [22]	Ours
mask1	30.29, 0.8588	30.96, 0.8804	**31.38, 0.8979**
mask2	30.46, 0.8516	31.23, 0.8875	**31.73, 0.9034**
mask3	30.80, 0.8663	31.43, 0.8904	**31.81, 0.9055**
mask4	30.65, 0.8610	31.15, 0.8863	**31.58, 0.9038**
mask5	30.74, 0.8631	31.46, 0.8939	**31.70, 0.9018**

datasets that were simulated by applying 5 different masks. The 5 masks of size 256×256 were extracted at the four corners and the center of the real captured mask [22]. We only trained *a single model* by the proposed network on the compound training dataset to deal with multiple masks, whereas we trained *five different models* associated with each mask by the DNU [37] and TSA-Net [22] methods on the datasets generated by the corresponding mask, respectively. Table 2 shows the average PSNR and SSIM results by these testing methods on the 10 scenes. We can see that the proposed method (only trained once on the compound training dataset) still outperforms the other two competing methods that were trained specifically for each mask, verifying the advantages of learning the measurement matrix $\mathbf{A}$ and $\mathbf{A}^\top$.

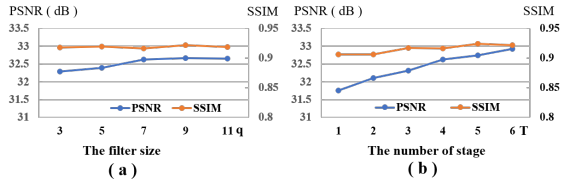

Figure 7. Ablation study on the effects of (a) the filter size; (b) the number of stage.

5.4. Ablation Study

We conduct several ablation studies to verify the impacts of different modules of the proposed network, including the choices of the filter sizes, number of stages and the use of dense connections.

Fig. 7 (a) shows the results with different filter sizes, where we can see that larger filter size can improve the HSI reconstruction quality. The improvement flattens out after $q = 7$ and thus we set $q = 7$ in our implementation. The

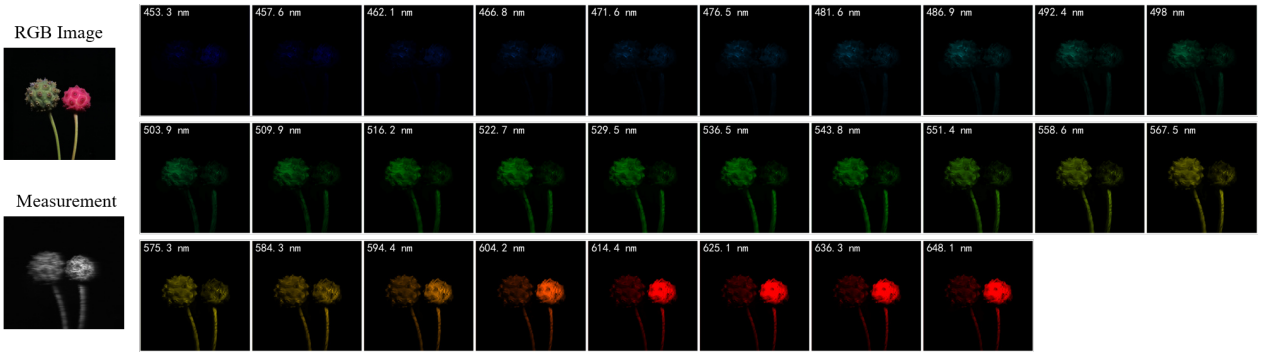

Figure 6. Reconstructed images of the real scene (*Scene 4*) with 28 spectral channels by the proposed method. Zoom in for better view.

results with different number of stages are shown in Fig. 7 (b), from which we observe that increasing the stage number T leads to better performance. We set $T = 4$ in our implementation for achieving a good trade-off between reconstruction performance and computational complexity. We have also conducted a comparison between the proposed network without and with dense connections. The comparison demonstrates that using dense connections can boost PSNR from 30.52dB to 32.63dB and SSIM from 0.8802 to 0.9166.

6. Real Data Results

We now apply the proposed method on the real SD-CASSI system [22] which captures the real scenes with 28 wavelengths ranging from 450nm to 650nm and has 54-pixel dispersion in the column dimension. Thus, the measurements captured by the real system have a spatial size of 660×714. Similar to TSA-Net [22], we re-trained the proposed method on all scenes of CAVE dataset and KAIST dataset. To simulate the real measurements, we injected 11-bit shot noise during training. We compare the proposed method with TwIST [1], GAP-TV [43], DeSCI [18] and TSA-Net [22]. Visual comparison results of the competing methods are shown in Fig. 8. It can be observed that the proposed method can recover more details of the textures and suppress more noise. Fig. 1 and 6 show reconstructed images of two real scenes (*Scene 3* and *Scene 4*) with 28 spectral channels by the proposed method. More visual results are shown in the SM.

7. Conclusions

We have proposed an interpretable hyperspectral image reconstruction method for coded aperture snapshot spectral imaging. Different from existing works, our network is inspired by the Gaussian scale mixture prior. Specifically, the desired hyperspectral images were characterized by the GSM models and then the reconstruction problem was formulated as a MAP estimation problem. Instead of using a manually designed prior, we have proposed to learn the s-

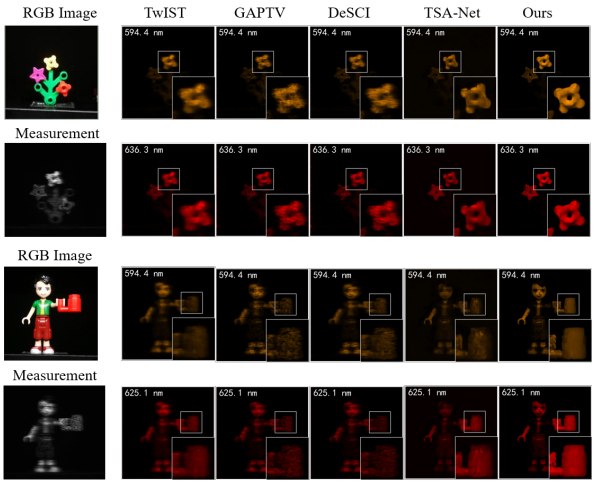

Figure 8. Reconstructed images of two real scenes (*Scene 1* and *Scene 3*) with 2 out of 28 spectral channels by the competing methods. Zoom in for better view.

cale prior of GSM by a DCNN. Furthermore, motivated by the auto-regressive model, the means of the GSM models have been estimated as a weighted average of the spatial-spectral neighboring pixels, and these filter coefficients are estimated by a DCNN as well aiming to learn sufficient spatial-spectral correlations of HSIs. Extensive experimental results on both synthetic and real datasets demonstrate that the proposed method outperforms existing state-of-the-art algorithms.

Our proposed network is not limited to the spectral compressive imaging such as CASSI and similar systems [46, 20], it can also be used in the video snapshot compressive imaging systems [28, 30, 29, 45]. Our work is paving the way of real applications of snapshot compressive imaging [19, 44].

Acknowledgments. This work was supported in part by the National Key R&D Program of China under Grant 2018AAA0101400 and the Natural Science Foundation of China under Grant 61991451, Grant 61632019, Grant 61621005, and Grant 61836008.

References

[1] José M Bioucas-Dias and Mário AT Figueiredo. A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image processing*, 16(12):2992–3004, 2007. 2, 6, 7, 8

[2] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005. 4

[3] Inchang Choi, Daniel S Jeon, Giljoo Nam, Diego Gutierrez, and Min H Kim. High-quality hyperspectral reconstruction using a spectral prior. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017. 2, 6

[4] Pierrick Coupé, Pierre Yger, Sylvain Prima, Pierre Hellier, Charles Kervrann, and Christian Barillot. An optimized blockwise nonlocal means denoising filter for 3-d magnetic resonance images. *IEEE transactions on medical imaging*, 27(4):425–441, 2008. 4

[5] Weisheng Dong, Guangming Shi, Yi Ma, and Xin Li. Image restoration via simultaneous sparse coding: Where structured sparsity meets gaussian scale mixture. *International Journal of Computer Vision*, 114(2-3):217–232, 2015. 2

[6] Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on image processing*, 20(7):1838–1857, 2011. 4

[7] Mário AT Figueiredo, Robert D Nowak, and Stephen J Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of selected topics in signal processing*, 1(4):586–597, 2007. 2

[8] Michael E Gehm, Renu John, David J Brady, Rebecca M Willett, and Timothy J Schulz. Single-shot compressive spectral imaging with a dual-disperser architecture. *Optics express*, 15(21):14013–14027, 2007. 1

[9] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1397–1409, 2012. 4

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 6

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[12] S. Jalali and X. Yuan. Snapshot compressed sensing: Performance bounds and algorithms. *IEEE Transactions on Information Theory*, 65(12):8005–8024, Dec 2019. 3

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[14] David Kittle, Kerkil Choi, Ashwin Wagadarikar, and David J Brady. Multiframe image estimation for coded aperture snapshot spectral imagers. *Applied optics*, 49(36):6824–6833, 2010. 1, 2

[15] Shutao Li, Xudong Kang, and Jianwen Hu. Image fusion with guided filtering. *IEEE Transactions on Image processing*, 22(7):2864–2875, 2013. 4

[16] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 2

[17] Xing Lin, Yebin Liu, Jiamin Wu, and Qionghai Dai. Spatial-spectral encoded compressive hyperspectral imaging. *ACM Transactions on Graphics (TOG)*, 33(6):1–11, 2014. 1, 2

[18] Yang Liu, Xin Yuan, Jinli Suo, David J Brady, and Qionghai Dai. Rank minimization for snapshot compressive imaging. *IEEE transactions on pattern analysis and machine intelligence*, 41(12):2990–3006, 2018. 2, 6, 7, 8

[19] S. Lu, X. Yuan, and W. Shi. Edge compression: An integrated framework for compressive imaging processing on cavs. In *2020 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 125–138, 2020. 8

[20] Xiao Ma, Xin Yuan, Chen Fu, and Gonzalo R. Arce. Led-based compressive spectral temporal imaging system. *Optics Express*, 2021. 8

[21] Ziyi Meng, Shirin Jalali, and Xin Yuan. Gap-net for snapshot compressive imaging. *arXiv: 2012.08364*, December 2020. 2

[22] Ziyi Meng, Jiawei Ma, and Xin Yuan. End-to-end low cost compressive spectral imaging with spatial-spectral self-attention. In *European Conference on Computer Vision*, pages 187–204. Springer, 2020. 1, 2, 4, 6, 7, 8

[23] Ziyi Meng, Mu Qiao, Jiawei Ma, Zhenming Yu, Kun Xu, and Xin Yuan. Snapshot multispectral endomicroscopy. *Opt. Lett.*, 45(14):3897–3900, Jul 2020. 1

[24] Xin Miao, Xin Yuan, Yunchen Pu, and Vassilis Athitsos. lambda-net: Reconstruct hyperspectral images from a snapshot measurement. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4058–4068. IEEE, 2019. 2, 6, 7

[25] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2502–2510, 2018. 4

[26] Qian Ning, Weisheng Dong, Fangfang Wu, Jinjian Wu, Jie Lin, and Guangming Shi. Spatial-temporal gaussian scale mixture modeling for foreground estimation. In *AAAI*, pages 11791–11798, 2020. 2, 4

[27] Javier Portilla, Vasily Strela, Martin J Wainwright, and Eero P Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image processing*, 12(11):1338–1351, 2003. 2

[28] Mu Qiao, Xuan Liu, and Xin Yuan. Snapshot spatial–temporal compressive imaging. *Opt. Lett.*, 45(7):1659–1662, Apr 2020. 8

[29] Mu Qiao, Xuan Liu, and Xin Yuan. Snapshot temporal compressive microscopy using an iterative algorithm with untrained neural networks. *Opt. Lett.*, 2021. 8

[30] Mu Qiao, Ziyi Meng, Jiawei Ma, and Xin Yuan. Deep learning for video compressive sensing. *APL Photonics*, 5(3):030801, 2020. 8

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[32] Guangming Shi, Tao Huang, Weisheng Dong, Jinjian Wu, and Xuemei Xie. Robust foreground estimation via structured gaussian scale mixture modeling. *IEEE Transactions on Image Processing*, 27(10):4810–4824, 2018. 2

[33] Muhammad Uzair, Arif Mahmood, and Ajmal S Mian. Hyperspectral face recognition using 3d-dct and partial least squares. In *BMVC*, volume 1, page 10, 2013. 1

[34] Burak Uzkent, Matthew J Hoffman, and Anthony Vodacek. Real-time vehicle tracking in aerial video using hyperspectral features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 36–44, 2016. 1

[35] Ashwin Wagadarikar, Renu John, Rebecca Willett, and David Brady. Single disperser design for coded aperture snapshot spectral imaging. *Applied optics*, 47(10):B44–B51, 2008. 1

[36] Lizhi Wang, Chen Sun, Ying Fu, Min H Kim, and Hua Huang. Hyperspectral image reconstruction using a deep spatial-spectral prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8032–8041, 2019. 2, 6, 7

[37] Lizhi Wang, Chen Sun, Maoqing Zhang, Ying Fu, and Hua Huang. Dnu: Deep non-local unrolling for computational spectral imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1661–1671, 2020. 2, 6, 7

[38] Lizhi Wang, Zhiwei Xiong, Guangming Shi, Feng Wu, and Wenjun Zeng. Adaptive nonlocal sparse representation for dual-camera compressive hyperspectral imaging. *IEEE transactions on pattern analysis and machine intelligence*, 39(10):2104–2111, 2016. 2

[39] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[40] Weiying Xie, Tao Jiang, Yunsong Li, Xiuping Jia, and Jie Lei. Structure tensor and guided filtering-based algorithm for hyperspectral anomaly detection. *Ieee Transactions on Geoscience and Remote Sensing*, 57(7):4218–4230, 2019. 1

[41] Zhiwei Xiong, Zhan Shi, Huiqun Li, Lizhi Wang, Dong Liu, and Feng Wu. Hscnn: Cnn-based hyperspectral image recovery from spectrally undersampled projections. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 518–525, 2017. 2

[42] Fumihito Yasuma, Tomoo Mitsunaga, Daisuke Iso, and Shree K Nayar. Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum. *IEEE transactions on image processing*, 19(9):2241–2253, 2010. 6

[43] Xin Yuan. Generalized alternating projection based total variation minimization for compressive sensing. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2539–2543. IEEE, 2016. 1, 2, 6, 7, 8

[44] X. Yuan, D. J. Brady, and A. K. Katsaggelos. Snapshot compressive imaging: Theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 38(2):65–88, 2021. 8

[45] Xin Yuan, Yang Liu, Jinli Suo, and Qionghai Dai. Plug-and-play algorithms for large-scale snapshot compressive imaging. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 8

[46] Xin Yuan, Tsung-Han Tsai, Ruoyu Zhu, Patrick Llull, David Brady, and Lawrence Carin. Compressive hyperspectral imaging with side information. *IEEE Journal of Selected Topics in Signal Processing*, 9(6):964–976, September 2015. 8

[47] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 2

[48] Shipeng Zhang, Lizhi Wang, Ying Fu, Xiaoming Zhong, and Hua Huang. Computational hyperspectral imaging based on dimension-discriminative low-rank tensor recovery. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10183–10192, 2019. 2

[49] Siming Zheng, Yang Liu, Ziyi Meng, Mu Qiao, Zhishen Tong, Xiaoyu Yang, Shensheng Han, and Xin Yuan. Deep plug-and-play priors for spectral snapshot compressive imaging. *Photon. Res.*, 9(2):B18–B29, Feb 2021. 2