A Comprehensive Comparison of Multi-Dimensional Image Denoising Methods

Zhaoming Kong, Xiaowei Yang, and Lifang He

Abstract—Filtering multi-dimensional images such as color images, color videos, multispectral images and magnetic resonance images is challenging in terms of both effectiveness and efficiency. Leveraging the nonlocal self-similarity (NLSS) characteristic of images and sparse representation in the transform domain, the block-matching and 3D filtering (BM3D) based methods show powerful denoising performance. Recently, numerous new approaches with different regularization terms, transforms and advanced deep neural network (DNN) architectures are proposed to improve denoising quality. In this paper, we extensively compare over 60 methods on both synthetic and real-world datasets. We also introduce a new color image and video dataset for benchmarking, and our evaluations are performed from four different perspectives including quantitative metrics, visual effects, human ratings and computational cost. Comprehensive experiments demonstrate: (i) the effectiveness and efficiency of the BM3D family for various denoising tasks, (ii) a simple matrix-based algorithm could produce similar results compared with its tensor counterparts, and (iii) several DNN models trained with synthetic Gaussian noise show state-of-the-art performance on real-world color image and video datasets. Despite the progress in recent years, we discuss shortcomings and possible extensions of existing techniques. Datasets and codes for evaluation are made publicly available at https://github.com/ZhaomingKong/Denoising-Comparison.

Index Terms—Multi-dimensional image denoising, nonlocal self-similarity, transform-domain techniques, block-matching filters, deep neural network.

1 INTRODUCTION

Image denoising plays an important role in modern imaging systems, and it has attracted much attention from both academia and industry. Theoretically, image denoising is a special case of inverse problems [1], which aims to estimate the underlying clean image from the noisy observation. In real-world applications, it can be used as a pre-processing step for subsequent tasks, such as visual tracking [2], segmentation [3] and classification [4]. Also, it may serve as the ultimate goal of image quality enhancement to produce more visually pleasant images.

Image denoising enjoys a rich history and the earliest works may date back to the 1950s [5]. The recent surge of denoising methods is mainly credited to the famous block-matching 3D (BM3D) [6] framework, which combines the nonlocal similarity characteristic [7] of natural images and the sparse representation in the transform domain [8]. At an early stage, many related works focus on filtering single-channel grayscale images [9], [10]. Recently, the advancement of imaging systems and technologies has largely enriched the information preserved and presented by multi-dimensional images, which could deliver more faithful representation for real scenes. For example, the popularity of mobile camera phones facilitates the capturing and recording of colorful and dynamic objects. The multispectral imaging (MSI) and hyperspectral imaging (HSI) techniques provide image information in both the spatial and spectral domain. Magnetic resonance imaging (MRI) utilizes a non-invasive imaging technology that produces three dimensional detailed anatomical images often used for disease detection, diagnosis, and treatment monitoring.

The growth of image size and dimension also poses greater demands on denoising. One major challenge of handling higher dimensional images is to effectively exploit the inter-correlated information of multiple channels or spectrums, and also find a balance between noise removal and detail preservation. In the past two decades, the representative BM3D method has been successfully extended to multi-dimensional images in two different ways. The first strategy is to apply a certain decorrelation transform such that in the transformed space each channel or spectral band may be filtered independently by some effective single-channel denoiser. For example, the color-BM3D (CBM3D) method [11] shows that the established opponent or YCbCr color transform of natural RGB images could provide near-optimal decorrelation of the color data [12]. An alternative solution is to model and utilize the channel- or bandwidth correlation by processing the full set of the multi-dimensional image data jointly. To achieve this goal, Maggioni et al. [13] extend BM3D to BM4D by using 3D cubes of voxels, which are then stacked into a 4D group.

Since the birth of BM3D, there is no shortage of denoising methods originating from different disciplines. Interestingly, from the traditional Gaussian denoisers [14] with matrix and tensor representations to recently developed approaches using different deep neural network (DNN) architectures [15], nearly all the newly proposed methods claim to outperform the BM3D family [6], [11], [13], [16], [17]. However, some recent studies [18], [19] come to a different conclusion, and it is observed that many methods are normally verified based on a limited number (often less than three) of datasets, and the parameters of BM3D-based methods may not be fine-tuned with certain noise.

Zhaoming Kong and Lifang He are with the Computer Science and Engineering Department, Lehigh University, PA, 18015 USA.
E-mail: zk219@lehigh.edu
Xiaowei Yang is with the school of Software Engineering, South China University of Technology, Guangdong Province, China.
estimation techniques [20], [21]. With a large number of existing methods [22] and outstanding survey papers [7], [8], [14], [15], [23], [24], [25], [26], [27], there still lacks a thorough comparison for the multi-dimensional image denoising tasks. In this paper, we intend to fill this gap by collecting and comparing different methods. The main contributions of this paper are three-fold:

1) We construct a new real-world indoor-outdoor color image (IOCI) and color video (IOCV) dataset with multiple different cameras. The dataset consist of images and video sequences captured based on both artificial indoor subjects and natural outdoor scenes under different lighting conditions, which are a good complement to current benchmark datasets with various challenging realistic backgrounds.

2) We compare over 60 multi-dimensional image denoising methods, and perform extensive experiments on both synthetic and real-world datasets. We adopt both subjective and objective metrics, and evaluate the denoising quality of compared methods with quantitative results and human visual ratings.

3) We make three interesting observations based on comprehensive experimental results. First the BM3D family still demonstrate very competitive performance for multiple denoising tasks in terms of both effectiveness and efficiency. Second, for traditional denoisers that learn solely on the noisy observation, we argue that a modified singular value decomposition (M-SVD) method is able to produce similar results with many tensor-based approaches. Furthermore, several DNN models trained with synthetic noise demonstrate impressive generalizability to real-world cases. Specifically, a fusion of collaborative and convolutional filtering (FCCF) model [28], and a U-Net [29] and ResNet [30] based denoising network (DRUNet) [31] produce the best results for the color image denoising task. The fast deep video denoising network (FastDVDNet) [32] and a nonlocal video denoising network (VNLNet) [33] show significant improvements for color video denoising compared with the benchmark video-BM4D (VBM4D) [17] implementation.

The rest of this paper is organized as follows. Section 2 introduces background knowledge, and Section 3 gives a brief review on related multi-dimensional image denoising techniques and datasets. Section 4 provides experimental results and discussions. Section 5 concludes the paper.

2 BACKGROUND

2.1 Symbols and Notations

In this paper, we denote multi-dimensional images, we mainly adopt the mathematical notations and preliminaries of tensors from [34]. Vectors and matrices are defined as a lower case letters and second-order tensors which are denoted by boldface main. The Frobenius norm of a tensor \( \| A \|_F = \sqrt{\sum_{i_1=1}^I \cdots \sum_{i_N=1}^I A_{i_1, \ldots, i_N}^2} \). The nuclear norm of a matrix is denoted by \( \| A \|_n \) which is the sum of all singular values of \( A \). The symbol of \( \otimes \) denotes the Kronecker product of matrices.

2.2 Noise modeling

Let us consider a noisy observation \( \mathcal{Y} \) and its underlying clean image \( \mathcal{X} \), a general assumption of noise distribution is additive white Gaussian noise (AGWN) with variance \( \sigma^2 \) represented by \( \mathcal{N}(0, \sigma^2) \), and the degradation process is then given as

\[
\mathcal{Y} = \mathcal{X} + \mathcal{N}
\]

According to [35], noise modeling can be more complex and challenging since noise in real-world images may be multiplicative and signal dependent. Therefore, there are some non i.i.d Gaussian models designed for multi-dimensional images, such as the mixed Gaussian impulse noise of color images [36], [37], sparse random corruptions of video data [38], strip noise removal of MSI [39], and Rician noise reconstruction of MRI [40]. In this paper, our synthetic experiments and analysis are mainly based on the AGWN model because (i) a majority of compared methods could handle Gaussian noise, (ii) certain types of noise could be transformed to Gaussian distribution, and (iii) Romano et al. [41] recently point out that the removal of AGWN from an image is largely a solved problem, which may help explain the effectiveness of the simplified noise modeling in Eq. (1).

2.3 Nonlocal self-similarity

The nonlocal self-similarity (NLSS) prior and patch representation of different multi-dimensional images.

The nonlocal self-similarity (NLSS) feature of natural images is probably the most important prior adopted by many different denoising methods. Briefly, NLSS refers to the fact that a local image patch often has many nonlocal similar patches to it across the image [42]. Usually, the similarity between two patches \( \mathcal{P}_A \) and \( \mathcal{P}_B \) is measured by their Euclidean distance \( d_{AB} = \| \mathcal{P}_A - \mathcal{P}_B \| \). In practice, to save some time, the search for similar patches is restricted to a local window \( \Omega_{SR} \) with pre-defined size. As illustrated in Fig. 1, the patch representation and the rule of similar patch search (SPS) may vary for different types of multi-dimensional images. For example, SPS of color images can be conducted only on the luminance channel to reduce complexity; for video sequences, SPS is performed along both temporal and spatial dimensions; and for MRI and MSI data, a patch could be represented by a 3D cube or a square tube with multiple spectral bands.
3 RELATED WORKS

In this section, we briefly introduce related multi-dimensional denoising methods and datasets, which are summarized in Table 1, Table 2, and Table 3, respectively. The timeline of representative denoising methods and datasets is illustrated in Fig. 2. In our paper, the denoising methods are roughly divided into two categories, namely ‘traditional denoisers’ and ‘DNN methods’, depending on whether neural network architectures are used. More detailed description of related works are available in [7], [8], [14], [15], [23], [24], [25], [26], [27].

3.1 Traditional denoisers

![Diagram](image_url)

Fig. 2. Illustration of the grouping-collaborative filtering-aggregation framework for traditional denoisers.

For traditional denoisers, training and denoising are usually accomplished only with the noisy image. To utilize the NLSS prior, the most popular and successful framework attributed to BM3D [6] mainly follows three consecutive stages: grouping, collaborative filtering and aggregation. The flowchart of this effective three-stage paradigm is illustrated in Fig. 2.

3.1.1 Grouping

For every d-dimensional noisy image patch $p_n$, based on certain patch matching criteria [47], [60], [92], [93], [94], [95], the grouping step stacks $K$ similar (overlapping) patches located within a local window $\Omega_{SR}$ into a $d+1$-dimensional group. For example, consider a 3D patch $p_n \in \mathbb{R}^{H \times W \times N}$, where $H$, $W$, and $N$ represents height, width, and the number of channels or spatial bands, respectively, the 4D group of $K$ patches can be directly represented by a fourth-order tensor $G_n \in \mathbb{R}^{H \times W \times N \times K}$, or a 2D matrix $G_n \in \mathbb{R}^{HW \times K}$ if every patch $p_n$ is reshaped into a long vector $p_n \in \mathbb{R}^{HW \times N}$.

3.1.2 Collaborative filtering

Collaborative filters operate on the noisy patch group $G_n$ to estimate the corresponding underlying clean group $\hat{G}_c$ via

$$\hat{G}_c = \arg \min_{\hat{G}_c} \|G_n - \hat{G}_c\|_F^2 + \rho \cdot \Psi(\hat{G}_c)$$

(2)

or in the matrix form

$$G_c = \arg \min_{G_c} \|G_n - G_c\|_F^2 + \rho \cdot \Psi(G_c)$$

(3)

where $\|G_n - \hat{G}_c\|_F^2$ or $\|G_n - G_c\|_F^2$ indicates the conformity between the clean and noisy groups, and $\Psi(\cdot)$ is a regularization term for certain priors. For example, to model the nonlocal redundancies, the low-rank prior is adopted in [54], [63], [64], [91] with $\Psi(G_c) = \|G_c\|_*$ for matrix and $\Psi(G_c) = \sum_{n=1}^N \|G_{c(n)}\|_*$ for tensor [96]. In addition, the dictionary learning model with over-complete representations [49], [50], [64] is utilized to reconstruct $G_c$ with a dictionary $D$ and sparse coding coefficients $C$ via

$$\hat{C} = \arg \min_C \|G_n - DC\|_F^2 + \lambda \|C\|_1$$

(4)

where $\|\cdot\|_*$ is the regularization term that enforces sparsity constraint on $C$. Once $C$ is computed, the latent clean patch group $G_c$ can be estimated as $G_c = DC$. In [80] and [81], Eq. 4 is extended to tensor for MSI denoising with higher-order SVD (HOSVD) [97], [98] and tensor-SVD (t-SVD) [99], [100] transforms, respectively. A simple and effective method is to model the sparsity with certain thresholding techniques [43], [101] to attenuate the noise. For example, the hard-thresholding technique is adopted by the BM3D family and some state-of-the-art tensor-based methods [19], [77], which attempts to shrink the coefficients $T(G_n)$ in the transform-domain [8] under a threshold $\tau$ via

$$G_1 = \begin{cases} T(G_n), & |T(G_n)| \geq \tau \\ 0, & |T(G_n)| < \tau \end{cases}$$

(5)

where $T$ represents a pre-defined or learned transform. The estimated clean group $\hat{G}_c$ is obtained by inverting the transform via

$$\hat{G}_c = T^{-1}(G_1)$$

(6)

The popularity of SVD-based transforms in traditional denoisers largely results from its invertible orthogonal bases.

3.1.3 Aggregation

To further smooth out noise, the estimated clean patches of $\hat{G}_c$ are averagely written back to their original location. More specifically, at the pixel level, every pixel $\hat{p}_i$ of the denoised image is the (weighted) average of all pixels at the same position of filtered group $G_{cr}$, which can be formulated as

$$\hat{p}_i = \sum_{k \in \hat{G}_c} w_{ik} \hat{p}_{ik}$$

(7)

where $w_{ik}$ and $\hat{p}_{ik}$ denote weight and local pixel of $\hat{G}_c$, respectively.

The major difference among various traditional denoisers mainly derives from the collaborative filtering scheme, which is often decided by the choice of transforms and algebraic representation. Intuitively, reshaping the 4D group $G$ of (2) into the 2D matrix $G$ of (3) may break the internal structure of multi-dimensional images. Therefore, a typical assumption [19], [77], [96], [80], [81], [84], [90], [102] is that tensor representation and decomposition techniques could help preserve more structure information, based on the fact that multi-dimensional images could be naturally represented by multi-dimensional array. However, the conventional and widely-used tensor model may fall into the unbalance trap [103], [104], leading to unsatisfactory image restoration performance. In this paper, we further show that with slight modifications, a simple modified SVD (M-SVD) implementation could produce competitive results for color image denoising compared with tensor-based methods. More details are given in Appendices.

3.2 DNN methods

The most recent development of image denoising is mainly brought by the applications of deep neural networks (DNNs), which demonstrate excellent performance in the latest denoising competition [140] based on the smartphone image denoising dataset (SIDD) [141]. Different from traditional denoisers that use only internal information of
## TABLE 1
Related traditional denoisers for multi-dimensional image data with different noise modeling techniques and applications. ‘CI’: color image, ‘CV’: color video, ‘MSI’: multispectral imaging, ‘HSI’: hyperspectral imaging, ‘MRI’: magnetic resonance imaging.

| Category | Representation | Methods | Noise Modeling | Applications | Key Words |
|----------|----------------|---------|----------------|--------------|-----------|
| **CNN**  |                |         |                |              |           |
| **Tensor** |               |         |                |              |           |
| **Matrix** |               |         |                |              |           |
| **Traditional denoisers** |             |         |                |              |           |
| **Network Architectures** |         |         |                |              |           |
| **DNN methods** |         |         |                |              |           |
| **CNN + GAN** |         |         |                |              |           |
the noisy observation, DNN methods usually adopt the supervised training strategy guided by external priors and datasets to minimize the distance $\mathcal{L}$ between predicted and clean images via

$$\min_{\theta} \sum_{i} \mathcal{L}(\mathcal{F}_\theta(Y_i), X_i) + \rho \cdot \Psi(\mathcal{F}_\theta(Y_i))$$

(8)

where $X_i$ and $Y_i$ are clean/noisy image (patch) pairs, $\mathcal{F}_\theta$ with parameter $\theta$ is a nonlinear function that maps noisy patches onto predicted clean patches, and $\Psi(\cdot)$ represents certain regularizers [109, 142]. Early methods [143, 144] work with the known shift blur function and weighting factors, and Burger et al. [105] show that a plain multilayer perceptron (MLP) network is able to compete with BM3D at certain noise levels. To extract latent features and exploit the NLSS prior, a more commonly used model is the convolutional neural network (CNN) [145], which is suitable for multi-dimensional data processing [146, 147], [148, 149] with flexible size of convolution filters and local receptive fields. Convolution operations are first applied to image denoising in [150], and Fig. 3 illustrates a simple CNN denoising framework with three convolution layers.

\[\text{Noisy} \xrightarrow{\text{Downscaling}} \text{Conv} \xrightarrow{\text{Conv}} \xrightarrow{\text{Conv}} \xrightarrow{\text{Layer 1}} \xrightarrow{\text{Layer 2}} \xrightarrow{\text{Layer 3}} \xrightarrow{\text{Denoised}} \]

Fig. 3. Illustration of a simple CNN denoising framework with three convolution layers.

Despite the effectiveness of DNN methods, they could be two-folded weapons, which enjoy three major advantages and also face the same challenges. First, DNN methods are able to utilize external information to guide the training process, and thus may not be restricted to the theoretical and practical bound of traditional denoisers [154]. However, DNN methods rely heavily on the quality of the training datasets, and also certain prior information such as ISO, shutter speed and camera brands, which are not always available in practice. Second, DNN methods could take advantage of the advanced GPU devices for acceleration and achieve real-time denoising [32], [155] for certain tasks. But the expensive computational resources may not be accessible to ordinary users and researchers. Third, the deep, flexible and sophisticated structure of DNN methods is capable of extracting the latent features underlying the noisy images, but compared with the implementation of CBM3D that only needs to store four small pre-defined transform matrices, the complex networks with millions of parameters may drastically increase the storage cost.

### 3.3 Datasets

In this section, we briefly introduce popular multi-dimensional image datasets used for synthetic and real-world experiments. The information is summarized in Table 3 with more details given in [27], [28], [141], [156], [157], [158], [159], [160], [161], [162], [163], [164], [165]. Some sample images of the datasets are illustrated in Fig. 5. Usually, datasets used for synthetic experiments consist of noise-free images acquired under ideal conditions with sufficient light and careful camera settings, whereas real-world images are inevitably contaminated by noise to various degrees, decided by the environments and equipments.

| Type | Name | Experiments | GT | # Cameras | Image size | # Images |
|------|------|-------------|----|-----------|------------|----------|
| Color image | RedLk | Synthetic | A | / | 768 x 512 x 3 | 24 |
| RENS 156 | Real-world | A | 3 | 364 x 256 x 3 | 120 |
| Nam-CC60 | Real-world | A | 3 | 512 x 512 x 3 | 57 |
| Nam-CC10 | Real-world | A | 3 | 512 x 512 x 3 | 57 |
| Dem 120 | Real-world | A | 3 | 512 x 512 x 3 | 120 |
| DND 18 | Real-world | A | 9 | 1024 x 1024 x 3 | 469 |
| Satt 149 | Real-world | A | / | 128 x 256 x 3 | 4 |
| Color video | DAVIS 107 | Synthetic | A | / | 480 x 640 x 3 x F | 30 |
| CROS 142 | Synthetic | A | 4 | 1920 x 1200 x 3 x F | 41 |
| Our ICVX | Real-world | A | 4 | 512 x 512 x 3 | 50 |
| MAV 105 | Synthetic | A | 1 | 512 x 512 x 3 | 50 |
| EKUL 166 | Synthetic | A | 1 | 1024 x 1024 x 3 | 201 |
| HYD 120 | Real-world | N/A | A | 1360 x 1040 x 3 | 77 |
| MRI | BrainWeb | Synthetic | A | / | 18 x 217 x 181 | 3 |
| OASIS | Synthetic | A | / | 256 x 256 x 128 | 2 |

### 3.3.1 Color image and video datasets

Some recent works [168] target raw image denoising based on the characteristics of photosensors, but the raw data are not always accessible, so we focus on the standard RGB (sRGB) color space. Compared to other types of images, the abundance of real-world color image datasets largely results from the relatively low cost of generating clean images.
Fig. 4. Timeline of representative multi-dimensional image denoising methods and datasets.

Fig. 5. Illustration of multi-dimensional image datasets. From the first row to the fourth row: color image, color video, MSI and MRI data.

reference images from noisy observations. Briefly, a real low-light image noise reduction dataset (RENOIR) [157] is the first attempt to obtain low-noise reference images with low light sensitivity (ISO = 100) and long exposure time. The Darmstadt noise dataset (DND) [18] utilizes a more careful post-processing technique to produce higher quality reference images based on low-ISO images. A more common strategy adopted by a variety of datasets [27], [28], [141], [158], [160] is called ‘image averaging’, which captures the same and unchanged scene for many times and computes their mean value to obtain the corresponding noise-free image. The rationale of this simple strategy is that for each pixel, the random noise is assumed to be larger or smaller than 0, and thus can be greatly suppressed by sampling the same pixel for many times [158]. [160].

Compared to the passion for collecting real-world color image datasets, fewer efforts have been made to produce realistic clean color videos. In this subsection, we briefly introduce the MSI and MRI datasets listed in Table 3. Briefly, the emergence of MSI and MRI reflects the development of new imaging techniques, but compared to color images, collecting MSI and MRI data are of greater difficulty and also more expensive. To the best of our knowledge, there is no publicly available realistic MSI and MRI dataset with noisy/clean pairs.

3.3.2 MSI and MRI datasets

In this subsection, we briefly introduce the MSI and MRI datasets listed in Table 3. Briefly, the emergence of MSI and MRI reflects the development of new imaging techniques, but compared to color images, collecting MSI and MRI data are of greater difficulty and also more expensive. To the best of our knowledge, there is no publicly available realistic MSI and MRI dataset with noisy/clean pairs.

MSI cameras are mostly spectroscopic devices, which need to be carefully calibrated in order to obtain a reliable spectral information [169]. The representative CAVE dataset [161] is constructed with a generalized assorted pixel (GAP) camera, it includes 32 scenes of a wide variety of real-world materials and objects. Each image has the size of 512 × 512 in space and includes full spectral resolution reflectance data ranging from 400 nm to 700 nm at 10 nm steps, which leads to 31 bands. The ICVL dataset [162] is acquired using a Specim PS Kappa DX4 camera, and it includes more than 200 different images collected at 1392 × 1300 spatial resolution, and the data are downsampled to 31 spectral channels from 400nm to 700nm at 10nm increments. The real-world harvard hyperspectral dataset (HHD) dataset [163] includes 75 images of size 1040 × 1392 × 31 collected under daylight, artificial and mixed illumination. The images were captured using a commercial hyperspectral camera (Nuance FX, CRI Inc) with 31 wavelength bands ranging from 420nm to 720nm at 10nm steps.

The MRI technique uses a strong magnetic field and radio waves to create detailed images of the organs and tissues within the body, and it could be regarded as an extension of white light imaging to incorporate better spectral resolution [170]. The acquisition of MRI data is often expensive and time-consuming. For synthetic experiments, the Brainweb dataset contains a set of MRI data volumes produced by an MRI simulator, which are often used as ‘ground-truth’ MR images for different validation purpose. Full 3D data volumes have been simulated using three sequences (T1-, T2-, and proton-density- (PD-) weighted) and a variety of slice thicknesses, noise levels, and levels of intensity non-uniformity. For realistic OASIS brain image datasets, the selected 3D T1-weighted (T1w) data of size 256 × 256 × 128 are acquired by an MP-RAGE volumetric sequence on a Siemens 1.5T Vision scanner, with repetition time (TR) 9.7
msec, echo time (TE) 4.0 msec, flip angle 10 degrees, inversion time (TI) 10 msec, duration time of 200 ms, and voxel resolution of $1 \times 1 \times 1.25$ mm$^3$.

### 3.3.3 The proposed color image and video datasets

![Image](54x547 to 297x601)

(a) The proposed IOCI dataset

![Image](54x625 to 297x672)

(b) The proposed IOCV dataset

Fig. 6. Illustration of the proposed IOCI and IOCV dataset.

In this subsection, we briefly introduce the motivation and details regarding the setup and protocol followed by our indoor-outdoor color image (IOCI) and video (IOCV) datasets, some examples are shown in Fig. 6.

**IOCI.** Many existing datasets are mainly restricted to static indoor scenes where the lighting conditions can be manually controlled to simulate different illumination. But in a lot of real-world cases, photos are taken in outdoor environments where the objects and source of light may be more complicated. In our IOCI dataset, nine different cameras are used to collect images of both indoor and outdoor scenes. Unlike previous works [28], [141], [158] that use pre-defined camera settings such as ISO, shutter speed and aperture, we mostly adopt the settings of the cameras’ ‘auto mode’, which minimizes human interference, and is also suitable for the uncontrollable outdoor environments. Captured images that display obvious misalignment and illumination differences are discarded. Each indoor and outdoor ‘ground-truth’ image is obtained by averaging at least 100 and 30 images of the same scene, respectively.

**IOCV.** The carefully-designed frame-by-frame strategy of the CRVD dataset is able to produce high-quality clean reference videos, but it also requires a lot of human effort, and the manually-created motions may not be continuous. Therefore, we adopt a video-by-video strategy, and instead of manually moving controllable objects, we propose to move cameras automatically. The procedure of generating mean videos as ground-truth is illustrated in Fig. 7. Specifically, we choose light-weight cameras to avoid clear shaking and misalignment, and fix the cameras to a rotatable tripod ball-head placed on top of a professional motorized slider. The slider and the tripod ball-head could be set to move and rotate at different speeds, which simulate the movement of observed objects in more than one directions. Both the slider and cameras are controlled remotely to avoid human interference. Noisy sequences are captured repeatedly for at least 30 times to generate the corresponding ‘ground-truth’ reference video by their mean value.

1Six cameras are used in previous experiments [19]

### 4 Experiments

In this section, we compare the performance of different methods for the denoising task of color images, color videos, MSI and MRI data. All implementations are provided by the authors or downloaded from the authors’ personal websites. The BM3D family normally consist of two stages, where the first stage utilizes hard-thresholding to provide an initial estimate, which is then smoothed with a nonlocal empirical Wiener filter to produce a final filtered image. In our experiments, we refer to these two output stages as ‘BMXDI’ and ‘BMXD2’, respectively. For DNN methods that require GPU devices, we use Google Colab’s free GPU. All other experiments are performed on a personal computer equipped with Core(TM) i5-7500 @ 3.4 GHz and 16GB RAM.

#### 4.1 Results for Real-World Color Image Datasets

##### 4.1.1 Experimental Settings

We compare the performance of over 25 effective methods on four real-world color image datasets, namely CC15, CC60, PolyU and our IOCI. Typically, the decisive parameters for traditional denoisers are input noise level, patch size and the number of local similar patches, etc, whereas the number of layers, learning rate and weight size are essential for DNN methods. It is noticed that many of the compared methods are designed for synthetic datasets, while ideally, in real-world cases, the parameters of all compared methods should be fine tuned to produce the best possible performance, but this could be computationally expensive and also the clean reference images for training and validation are always not available. Therefore, a more practical way is to selectively apply effective pre-trained models. Since most of the DNN implementations offer two to six pre-trained models with different parameter settings for testing, for fair comparison, we tune the input noise level for each Gaussian denoiser from four different values, which could be regarded as four pre-trained models corresponding to ‘low’, ‘medium-low’, ‘medium-high’ and ‘high’ denoising modes. We report the best average values of all compared methods on each dataset. Also, similar to [19], the best result of CBM3D on every image is included to further investigate its effectiveness, and this implementation is termed ‘CBM3D_best’. Peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [177] are employed for objective evaluations. Normally, the higher the PSNR and SSIM values, the better the quality of denoised images.
4.1.2 Objective metric

Detailed average PSNR and SSIM results are given in Table 4. Overall, CBM3D is able to produce very competitive performance on almost every dataset, while DNN methods do not always demonstrate advantages over traditional denoisers, largely due to the lack of training data.

More specifically, for traditional denoisers, three different transform-domain approaches namely CBM3D, the non-local haar (NLH) transform, and color multispectral t-SVD (CMSst-SVD) show similarly good denoising performance on all datasets. Also, the effectiveness of CMSst-SVD and CBM3D1 indicates that a one-step implementation with a small number of local patches and the hard-thresholding technique is able to produce very competitive results. It is noticed that the matrix-based methods such as multi-channel weighted nuclear norm (MCWNNM) and M-SVD present almost the same denoising capacity as tensor-based methods such as hyper-Laplacian regularized unidirectional low-rank tensor (LLRT) and 4DHOVSVD, which indicates that the use of tensor representation may not boost structural information retrieval in realistic cases.

For DNN methods, the dual adversarial network (DANet) utilizes the PolyU dataset for training, on which it shows outstanding denoising results, but it struggles on other datasets when the training data are unavailable. Another important and interesting observation is that DNN models trained with AWGN noise are more competitive and robust than those trained with specific real-world datasets, as can be seen from the excellent performance of FFDNet, FCCF and DRUNet on all datasets, which strongly supports the effectiveness of the Gaussian noise modeling in Eq. [1].

From the perspective of denoising speed reported in Table 2, Self2Self requires thousands of iterations to train its network with the noisy observation, resulting in the highest cost even run on advanced GPU devices. Other more efficient DNN models are able to handle color images of size $512 \times 512 \times 3$ within 0.2 seconds. For traditional denoisers, the computational cost lies mainly in the iterative local patch search and training. For example, M-SVD spends 26 and 42 seconds on grouping and learning local SVD transform, respectively, but it is slightly faster than 4DHOVSVD, because it avoids folding and unfolding operations of tensor data along different modes. Among all the traditional denoisers, the state-of-the-art CBM3D is the most efficient because it does not have to train local transforms, and its grouping step is performed only on the luminance channel.

4.1.3 Visual evaluation

Visual evaluations of compared methods on CC15, PolyU and the proposed IOCI datasets are given in Fig. 8 and Fig. 9.
Fig. [10] respectively. Two commercial softwares Neat Image [NI] (https://ni.neatvideo.com/) and Topaz DeNoiseAI (https://topazlabs.com/) are included in our comparison, and their optimized denoising speed enables users to fine-tune parameters for satisfactory visual effect. From Fig. [8] it can be seen that except for DRUNet and DBF, all other methods produce color artifacts, especially for patch-based approaches such as CBM3D, CMSt-SVD and NLH. Comparison in Fig. [9] indicates that an improvement of objective metric like PSNR does not always guarantee enhancement of visual effect. Except for NI, DeNoiseAI, and NLH, all other methods suffer over-smoothness to certain extents due to improper choice of parameters. From Fig. [8] and Fig. [9] it seems that DeNoiseAI improve the visual quality in terms of noise removal and detail recovery because of its unique sharpening technique. Nevertheless, it also shows some color artifacts, and an interesting example of our IOCV dataset in Fig. [10] shows that the advanced DeNoiseAI software tends to produce unwanted artifacts along the edges, which are barely noticeable in mean image, noisy observation and results of other compared methods.

4.2 Results for Real-World Color Video Dataset

4.2.1 Experimental settings

We evaluate the performance of VBM4D [17], CMSt-SVD [19], VNLNet [33] and FastDVDNet [32] with our IOCV dataset. For VBM4D, to save some time, we only use its first stage implementation named ‘VBM4D1’. The parameters of compared methods are chosen in the same way as section 4.1.1 and similar to [32], the PSNR and SSIM of a sequence are computed as their average values of each frame.

4.2.2 Objective metric

Table 5 lists the average PSNR and SSIM results of compared methods. From Table 5, it can be seen that two traditional denoisers CMSt-SVD and VBM4D show similar performance in terms of both efficiency and effectiveness. For DNN methods, VNLNet produces the best results by integrating the NLSS prior into CNN models, but its computationally expensive patch search process also results in high computational complexity. By contrast, FastDVDNet achieves almost real-time video denoising by getting rid of the time-consuming flow estimation step. Specifically, it takes FastDVDNet less than 0.1 seconds to process a single video frame, which is 8 times faster than VNLNet, and at least 13 times faster than the benchmark traditional denoisers VBM4D and CMSt-SVD. The denoising speed of FastDVDNet is remarkable considering its competitive denoising performance in all cases. Interestingly, comparing the results of color image denoising in Table 4 and video denoising in Table 5, it is noticed that the CNN-based models present more dominating performance in the video denoising task, which may be explained by the fact that videos contain more correlated information among different frames that could be exploited by the CNN architecture.

4.2.3 Visual evaluation

Visual comparison is given in Fig. [11]. It can be clearly observed that the noisy video is severely corrupted. The patch-based traditional denoisers CMSt-SVD and VBM4D1 leave behind medium-to-low-frequency noise, which would lead to video sequences with noticeable flickering, but VBM4D1 looks more visually pleasant than CMSt-SVD because of its built-in de-flickering function. The DNN methods show better results in this case. Specifically, VNLNet gives the best and most impressive visual effects by suppressing noise while recovering true colors and details. FastDVDNet is also able to effectively remove noise but also shows slight over-smooth effects.

4.2.4 Human ratings

Due to the limitations of hardware equipments and environment, the mean videos generated by the motorized slider with our video averaging strategy also inevitably exhibit some noise, flickering, staircase effects, motion blur and misalignment that would undermine the accuracy of objective evaluations. Also, the quality of videos may not be judged frame by frame. Therefore, we further conduct subjective evaluations by collecting human opinions [172]. More specifically, we randomly select 10 videos from our IOCV dataset, and invite 10 volunteers to rate the mean, noisy and denoised sequences. The invited volunteers have very little background knowledge of image or video denoising, and they are not aware of how the presented video sequences are processed. For each of the 10 videos, the volunteers are asked to choose at least 2 best sequences, which then earn 1 point for the corresponding methods. The detailed human rating results are reported in Table 6. First, it is noticed that our mean-video strategy has the highest score on 9 of 10 videos, which justifies the effectiveness of the proposed IOCV dataset for objective evaluation. Furthermore, the human rating results are consistent with our evaluations in Table 5 and Fig. [11] which suggests that DNN methods output higher quality videos than the benchmark traditional denoisers on our IOCV dataset. Also, human eyes prefer VBM4D over CMSt-SVD because CMSt-SVD presents more temporally decorrelated low-frequency noise in flat areas, which will appear as particularly bothersome for the viewer.

To conclude, the CNN-based models FastDVDNet and VNLNet show clear advantages over the representative traditional denoisers. VNLNet demonstrates state-of-the-art

| Dataset | Noisy | CMSt-SVD | VBM4D | FastDVDNet | VNLNet |
|---------|-------|----------|-------|-------------|--------|
| HUAWEI | 38.11 | 40.80 | 41.19 | 41.26 | 41.35 |
| BC | 0.9953 | 0.9834 | 0.9830 | 0.9857 | 0.9868 |
| HUAWEI | 35.58 | 38.71 | 38.76 | 38.03 | 38.79 |
| FC | 0.9413 | 0.9780 | 0.9785 | 0.9776 | 0.9809 |
| OPPO | 32.06 | 33.44 | 33.26 | 33.05 | 33.56 |
| BC | 0.9071 | 0.9508 | 0.9456 | 0.9476 | 0.9544 |
| OPPO | 36.90 | 39.66 | 39.56 | 39.06 | 40.11 |
| FC | 0.9447 | 0.9791 | 0.9785 | 0.9751 | 0.9823 |

Implementation / MEX MEX GPU GPU
Time/Frame (s) / 1.86 1.29 0.08 0.66

TABLE 5

PSNR and SSIM values of compared methods on real-world color video dataset. Average time per frame is also included. The best results are bolded. ‘FC’ and ‘BC’ represent front camera and back camera, respectively.
Fig. 8. Visual evaluations of compared methods (PSNR) on CC15 dataset.

Fig. 9. Visual evaluation of compared methods (PSNR) on PolyU dataset.

Fig. 10. Visual evaluation of compared methods (PSNR) on our IOCI's IPHONE 5S dataset.
4.3 Results for MSI Datasets

4.3.1 Experimental Settings

In this section, we evaluate the performance of various MSI denoising methods on CAVE, ICVL and HHD datasets. For synthetic experiments, due to the high computational cost of some implementations, we mainly use the CAVE dataset for comparison. Similarly, considering the large size of the ICVL dataset, we select the first 20 MSI data and evaluate two representative methods MSt-SVD and QRNN3D. Apart from the classical spatial-based quality indices PSNR and SSIM, we adopt two widely used spectral-based quality indicators for MSIs, namely spectral angle mapper (SAM) and dimensionless global relative error of synthesis (ERGAS). Different from PSNR and SSIM, recovered MSI data with lower SAM and ERGAS are considered of better quality.

4.3.2 Synthetic MSI dataset

We assume that entries in all slices of MSI data are corrupted by zero-mean i.i.d Gaussian noise with different noise levels. Detailed denoising results on CAVE and ICVL dataset are listed in Table 7 and 8 respectively.

Benefiting from the advanced GPU devices, QRNN3D is 60 times faster than the benchmark traditional denoisers MSt-SVD and BM4D. However, like many DNN methods, QRNN3D is also subject to the influence of training datasets. Its models trained with the ICVL dataset can not handle noisy images from the CAVE dataset, as can be seen from the difference of Table 7 and Table 8. Similar to QRNN3D, HSI-SDeCNN also fails to produce competitive performance, because its network models are trained only with certain noise levels and bandwidths. For traditional denoisers, results in Table 7 shows the advantage of tensor-based methods by exploiting spatial and spectral correlation. Specifically, NGMeet, LLRT and LTDL consistently outperform other compared approaches at all noise levels by more than 1db. However, it takes LLRT and LTDL more than 40 minutes to process a single image, because their iterative strategy with a large number of local similar patches significantly increases the computational burden. By comparison, MSt-SVD is more efficient, since it is an one-step approach that utilizes global patch representation.

Fig. 12 compares competitive tensor-based denoisers at high noise level $\sigma = 100$ on the CAVE dataset. In this extreme case, the pre-defined transform bases of BM4D may not take advantage of the correlation among all the spectral bands, while all other compared methods can effectively remove noise and retain details to certain extents. Interestingly, although from Table 7, MSt-SVD is outperformed by NGMeet, LLRT and LTDL, it shows the best detail recovery ability, as can be seen from the white text in Fig. 12. This observation indicates increasing the number of iterations and local similar patches may not help preserve fine details and structure of MSI data at high noise levels.

4.3.3 Real-world HHD Dataset

Inevitably, images of HHD dataset are contaminated by noise to varying degrees under different illumination. Considering the large size of noisy images, we examine the
TABLE 7
Comparison of quantitative results and computational time (minutes) under i.i.d Gaussian noise $\sigma = \{10, 30, 50, 100\}$ on the CAVE dataset.

| Methods | $\sigma = 10$ | $\sigma = 30$ | $\sigma = 50$ | $\sigma = 100$ | Time (m) |
|---------|-------------|-------------|-------------|-------------|--------|
| Noisy | 28.13 | 0.4571 | 238.40 | 0.7199 | 18.59 | 0.1069 | 676.01 | 1.0085 | 14.15 | 0.0475 | 1126.7 | 1.1461 | 8.13 | 0.0136 | 2253.4 | 1.3074 |
| BM4D1 | 42.76 | 0.9716 | 42.19 | 0.1585 | 37.13 | 0.9235 | 82.33 | 0.2910 | 34.55 | 0.8803 | 109.43 | 0.3591 | 31.17 | 0.7763 | 158.83 | 0.5161 |
| BM4D2 | 44.41 | 0.9797 | 34.32 | 0.1048 | 39.07 | 0.9493 | 94.12 | 0.3228 | 36.46 | 0.9171 | 128.15 | 0.5161 | 32.92 | 0.8284 | 128.15 | 0.5161 |
| TDL | 42.77 | 0.9602 | 30.53 | 0.1096 | 40.51 | 0.9408 | 50.05 | 0.1574 | 37.18 | 0.9171 | 120.77 | 0.2576 | 38.48 | 0.9283 | 65.23 | 0.2579 |
| FastHyDe | 42.87 | 0.9763 | 37.13 | 0.9235 | 82.33 | 0.2910 | 34.55 | 0.8803 | 109.43 | 0.3591 | 31.17 | 0.7763 | 158.83 | 0.5161 |
| LLTDL | 45.90 | 0.9859 | 29.06 | 0.0925 | 41.20 | 0.9644 | 49.73 | 0.1301 | 38.88 | 0.9507 | 65.54 | 0.1616 | 34.97 | 0.9072 | 103.65 | 0.2353 |

Average PSNR/SSIM and computational time (minutes) of MSt-SVD and QRNN3D under i.i.d Gaussian noise $\sigma \in \{10, 20, 40\}$ on the first 20 MSI data of ICVL dataset.

| Methods | $\sigma = 10$ | $\sigma = 20$ | $\sigma = 40$ | Time (m) |
|---------|-------------|-------------|-------------|--------|
| MSt-SVD | 46.58/0.9868 | 42.98/0.9746 | 39.32/0.9525 | 10.1 |
| QRNN3D | 47.37/0.9908 | 45.08/0.9842 | 42.38/0.9731 | 0.16 |

TABLE 8

| Methods | $\sigma = 10$ | $\sigma = 20$ | $\sigma = 40$ | Time (m) |
|---------|-------------|-------------|-------------|--------|
| Noisy | 28.13 | 0.4571 | 238.40 | 0.7199 | 18.59 | 0.1069 | 676.01 | 1.0085 | 14.15 | 0.0475 | 1126.7 | 1.1461 | 8.13 | 0.0136 | 2253.4 | 1.3074 |
| BM4D1 | 42.76 | 0.9716 | 42.19 | 0.1585 | 37.13 | 0.9235 | 82.33 | 0.2910 | 34.55 | 0.8803 | 109.43 | 0.3591 | 31.17 | 0.7763 | 158.83 | 0.5161 |
| BM4D2 | 44.41 | 0.9797 | 34.32 | 0.1048 | 39.07 | 0.9493 | 94.12 | 0.3228 | 36.46 | 0.9171 | 128.15 | 0.5161 | 32.92 | 0.8284 | 128.15 | 0.5161 |
| TDL | 42.77 | 0.9602 | 30.53 | 0.1096 | 40.51 | 0.9408 | 50.05 | 0.1574 | 37.18 | 0.9171 | 120.77 | 0.2576 | 38.48 | 0.9283 | 65.23 | 0.2579 |
| FastHyDe | 42.87 | 0.9763 | 37.13 | 0.9235 | 82.33 | 0.2910 | 34.55 | 0.8803 | 109.43 | 0.3591 | 31.17 | 0.7763 | 158.83 | 0.5161 |
| LLTDL | 45.90 | 0.9859 | 29.06 | 0.0925 | 41.20 | 0.9644 | 49.73 | 0.1301 | 38.88 | 0.9507 | 65.54 | 0.1616 | 34.97 | 0.9072 | 103.65 | 0.2353 |

Effectiveness of five efficient benchmarks on handling real-world MSI data. Since there is no clean reference image, we conduct visual evaluations, and for fair comparison, the parameters of compared methods are carefully tuned. The results are visualized in Fig. 13. Without realistic training data, the CNN model of QRNN3D produces obviously blurry image. Interestingly, different from our observations of synthetic AWGN comparison in Fig. 12, no compared methods could consistently outperform BM4D in this realistic case. As can be seen in Fig. 13, all state-of-the-art algorithms struggle to adapt to different local image contents to balance smoothness and sharp details. For patch-based methods, applying a same set of parameters or transforms to multiple spectral bands may result in similar denoise patterns across the whole image. Therefore, apart from the high computational burden, another challenge of filtering MSI data lies in dealing with the spatial and spectral variation of noise levels and distributions.

4.4 Results for MRI datasets
4.4.1 Experimental Settings

Different from the AWGN noise modeling of MSI, MRI data are often corrupted by Rician noise [40]. Specifically, let $X$ be the original noise-free signal, the noise Rician MRI data $Y$ can be obtained via

$$Y = \sqrt{(X + \sigma_1^2) + (\sigma_2^2)}$$

where $\sigma_1, \sigma_2 \sim N(0, 1)$ are i.i.d. random vectors following the standard normal distribution, $\sigma$ is the standard deviation in both real and imaginary channels of the noisy 3D MR images. To handle Rician noise, the technique of forward and inverse variance stabilizing transform (VST) [175] is often adopted by Gaussian denoisers via

$$\hat{X} = \text{VST}^{-1}(\text{denoise}(\text{VST}(Y, \sigma, \sigma_{\text{VST}}), \sigma))$$

where $\text{VST}^{-1}$ denotes the inverse of VST, $\sigma_{\text{VST}}$ is the stabilized standard deviation after VST transform, and $\sigma$ is the standard deviation of the noise in Eq. (9). According to Eq. (10), the noisy Rician data $Y$ is first stabilized by the VST and then filtered by certain Gaussian denoisers using a constant noise level $\sigma_{\text{VST}}$, and then the final estimate is obtained by applying the inverse VST to the output of the denoising result [13].

In our experiments, we use PSNR and SSIM as the objective metrics, and similar to [13], [70], the PSNR value then filtered by certain Gaussian denoisers using a constant

$$\hat{X} = \text{VST}^{-1}(\text{denoise}(\text{VST}(Y, \sigma, \sigma_{\text{VST}}), \sigma))$$

where $\text{VST}^{-1}$ denotes the inverse of VST, $\sigma_{\text{VST}}$ is the stabilized standard deviation after VST transform, and $\sigma$ is the standard deviation of the noise in Eq. (9). According to Eq. (10), the noisy Rician data $Y$ is first stabilized by the VST and then filtered by certain Gaussian denoisers using a constant noise level $\sigma_{\text{VST}}$, and then the final estimate is obtained by applying the inverse VST to the output of the denoising result [13].

In our experiments, we use PSNR and SSIM as the objective metrics, and similar to [13], [70], the PSNR value is often computed on the foreground defined by $X_f = \{x \in X : x > 10 \cdot D/255\}$, where $D$ is the peak of clean data $X$. 

![Fig. 13. Visual evaluation of compared MSI denoising methods on the real-world HHDD dataset.](image-url)
4.4.2 Synthetic Brainweb dataset
The volume data are added with varying levels of stationary Rician noise from 1% to 19% of the maximum intensity with an increase of 2%. In real-world cases, the noise level is usually much lower than 19%, but we are also interested in the denoising capability of compared methods under extreme conditions. Table 5 lists detailed quantitative results, and Fig. 14 compares the denoising performance at high noise levels when $\sigma \geq 11\%$. At lower noise levels, PRI-NLPCA is able to take advantage of the high-quality initial estimate of NLPCA, and achieves the best performance when $\sigma \leq 9\%$. As noise level increases, tensor-based methods demonstrate advantages of extracting latent features, and the iterative low-rank HOSVD (ILR-HOSVD) produces the best results when $9\% \leq \sigma \leq 15\%$. However, the iterative learning strategy is subject to the presence of extremely high noise, while BM4D benefits from its pre-defined transforms and outperforms other methods when $\sigma \geq 17\%$.

![Graphs showing PSNR and SSIM values vs noise level for different methods.](image)

Fig. 14. Average PSNR and SSIM values of representative compared methods with T1w, T2w and PDw data at high noise levels ($\sigma \geq 11\%$).

Visual evaluation on T1w data when $\sigma = 19\%$ is illustrated in Fig. 15 and it shows that BM4D2 is successful in both noise suppression and detail preservation. Compared to NLPCA and ILR-HOSVD, MSt-SVD is less affected by severe noise, as it may benefit from global patch representation with fast Fourier transform (FFT). The state-of-the-art NLM methods, namely PRINLM and PRI-NLPCA exhibit pleasant visual effects in homogeneous areas at the cost of slight over-smoothness along the edges. Furthermore, PRI-NLPCA shows some artifacts on the background due to the influence of noise on learning local PCA transform.

4.4.3 Real-world MRI dataset
To evaluate the performance of compared methods on real-world 3D MR data, we carry out experiments on T1w MR images of the OASIS dataset [165]. The Rician noise levels of two selected T1w data, namely OAS1_0112 and OAS1_0092 are estimated to be 3% and 4.5% of the maximum intensity, respectively [78]. The filtered images of different methods are compared in Fig. 16 and Fig. 17, respectively. Based on our synthetic experiments in Table 9, we learn that in most cases all methods could achieve competitive denoising performance at low noise levels, as can also be seen from the excellent visual effects on realistic OAS1_0112 data in Fig. 16. Thus, from the perspective of real-world MRI denoising, BM4D1 and PRINLM should be preferred due to their low computational cost reported in Table 9. Despite the similar denoising results of compared methods, we also observe that on certain slices of the 3D MR data, NLPCA produces the best results in terms of fine detail preservation, while other methods exhibit over-smoothness to varying degrees, as is illustrated in Fig. 17. This interesting observation is another vivid example to show that the use of tensor representation does not always help preserve more structural information of the underlying clean images.

4.5 Discussion
Our discussion in this section will mainly focus on color image denoising due to the abundance of advanced compared methods and real-world datasets.

4.5.1 The robustness of CBM3D
Recently, there is some criticism [127] of CBM3D that to achieve competitive denoising performance, it may need to traverse different Gaussian noise levels $\sigma$, which could be potentially time-consuming and impractical without ground-truth. Such criticism may be unfair for CBM3D for two reasons. First, compared to nearly all methods listed in Table 4, CBM3D does not involve any training process which could be influenced by the noise distributions of specific datasets. Second, in our experiments, we discover that the pre-defined transforms of CBM3D are not very sensitive to the choice of the input noise level parameter $\sigma$ within a reasonable range. Fig. 18 compares the PSNR and SSIM values of CBM3D with $\sigma \in [10, 30]$ on five datasets. It is noticed that choosing a noise level $\sigma$ between 10 and 20 could yield similarly good performance for CBM3D, making it an exceptionally robust method. Furthermore, from the perspective of blind denoising [68], [121], [176], [177], CBM3D could be viewed as a competitive blind denoiser with input noise level fixed to be $\sigma = 15$.

4.5.2 Understanding the denoising performance
The theoretical bound of compared image denoising methods is hard to obtain, but it is interesting to investigate the denoising capability of compared methods under the challenging practical cases when prior information, such as training datasets and camera settings are unavailable. We use our IOCI’s FUJIFILM X100T camera dataset, and for each scene, we generate three new mean images by averaging 3, 5 and 10 noisy images, and they are named ‘Mean_3’, ‘Mean_5’ and ‘Mean_10’, respectively. They could be regarded as images captured via different continuous shooting modes with ‘high’, ‘medium’ and ‘low’ noise levels. Fig. 19 illustrates the average PSNR differences of six implementations compared with ‘Mean_3’ on FUJI dataset. Interestingly, Fig. 19 shows that state-of-the-art denoising methods only produce marginal improvements compared with Mean_3, which indicates that their denoising performance is similar to obtaining the mean image by averaging 3 consecutive noisy observations.

4.5.3 Denoising with resizing
To boost the denoising performance of traditional patch-based denoisers, many methods [11], [63], [64], [74], [77], [76], [64], [86] propose to iteratively combine the filtered
TABLE 9
Average PSNR/SSIM and computational time (s) of different methods with T1w, T2w, and PDw data corrupted by Rician noise. The standard deviations $\sigma$ of the noise is expressed as percentage relative to the maximum intensity value of the noise-free data [13].

| Noise | Noisy | ONLM | AONLM | ODCT | PRINLM | BM4D1 | BM4D2 | NLPCA | PRI-NLPCA | ILR-HOSVD | 4DHOSVD | MSt-SVD |
|-------|-------|------|-------|------|--------|-------|-------|-------|-----------|-----------|---------|---------|
| 1     | 40.0/0.971 | 41.7/0.991 | 41.9/0.991 | 43.3/0.999 | 43.2/0.992 | 43.4/0.992 | 43.4/0.992 | 44.1/0.994 | 44.3/0.994 | 44.4/0.999 | 44.0/0.999 | 43.9/0.993 | 44.0/0.993 |
| 3     | 30.5/0.822 | 36.4/0.969 | 36.5/0.971 | 36.9/0.975 | 36.6/0.972 | 37.2/0.975 | 37.7/0.977 | 39.6/0.981 | 38.0/0.976 | 37.5/0.976 | 38.0/0.976 | 37.5/0.975 |
| 5     | 26.1/0.676 | 33.3/0.940 | 33.8/0.947 | 34.0/0.955 | 33.7/0.950 | 34.6/0.959 | 33.8/0.940 | 35.3/0.967 | 35.2/0.963 | 34.6/0.959 | 34.6/0.958 |
| 7     | 23.2/0.564 | 31.2/0.911 | 31.9/0.921 | 31.7/0.927 | 32.0/0.934 | 31.8/0.926 | 32.8/0.943 | 31.9/0.912 | 33.4/0.951 | 33.4/0.948 | 32.9/0.941 | 32.8/0.940 |
| 9     | 21.0/0.479 | 29.4/0.880 | 30.5/0.894 | 30.2/0.905 | 30.5/0.911 | 30.3/0.899 | 31.5/0.928 | 31.2/0.908 | 31.8/0.934 | 32.0/0.933 | 31.5/0.922 | 31.3/0.924 |
| 11    | 19.3/0.414 | 27.8/0.847 | 29.3/0.865 | 29.1/0.883 | 29.3/0.888 | 29.2/0.873 | 30.5/0.912 | 30.0/0.883 | 30.7/0.916 | 30.9/0.917 | 30.4/0.903 | 30.2/0.906 |
| 13    | 17.8/0.361 | 26.3/0.811 | 28.3/0.836 | 28.1/0.861 | 28.2/0.864 | 28.1/0.844 | 29.6/0.895 | 29.0/0.855 | 29.6/0.898 | 29.9/0.902 | 29.4/0.883 | 29.3/0.888 |
| 15    | 16.6/0.319 | 24.9/0.771 | 27.4/0.807 | 27.1/0.840 | 27.4/0.841 | 27.3/0.814 | 28.8/0.879 | 28.1/0.826 | 28.8/0.879 | 29.2/0.886 | 28.6/0.863 | 28.5/0.871 |
| 17    | 15.5/0.284 | 23.4/0.729 | 26.5/0.775 | 26.5/0.820 | 26.6/0.817 | 26.5/0.785 | 28.3/0.862 | 27.3/0.798 | 28.0/0.860 | 27.7/0.838 | 27.8/0.842 | 27.7/0.853 |
| 19    | 14.6/0.254 | 22.0/0.684 | 25.7/0.744 | 25.8/0.799 | 25.9/0.797 | 25.7/0.755 | 27.4/0.845 | 26.6/0.772 | 27.3/0.841 | 27.0/0.819 | 27.1/0.822 | 27.1/0.835 |

Implementation / Mex Mex Mex Mex Mex Mex MATLAB MEX MATLAB MATLAB MEX

Time (s) / 139.9 868.1 10.5 18.6 69.9 211.3 638.8 868.9 1896.3 706.9 99.6

(a) Clean (b) Noisy (c) NLPCA (d) PRI-NLPCA (e) MSt-SVD (f) BM4D2

Fig. 15. Visual evaluation of compared methods on synthetic Brainweb T1w data with noise level $\sigma = 19\%$.

(a) MRI (b) NLPCA (c) PRINLM (d) PRI-NLPCA (e) MSt-SVD (f) HOSVDRS (g) BM4D1 (h) BM4D2

Fig. 16. Visual evaluation of compared methods on Real OAS1_0112 T1w data with estimated noise level $\sigma = 3\%$.

(a) MRI (b) NLPCA (c) PRINLM (d) PRI-NLPCA (e) MSt-SVD (f) HOSVDRS (g) BM4D1 (h) BM4D2

Fig. 17. Visual evaluation of compared methods on Real OAS1_0092 T1w data with estimated noise level $\sigma = 4.5\%$.

(a) PSNR (b) SSIM

Fig. 18. PSNR and SSIM change of CBM3D1 with noise level $\sigma \in [10, 30]$ on five real-world color image datasets.

(a) PSNR (b) SSIM

Fig. 19. PSNR and SSIM difference of six implementations compared with ‘Mean _3’ on FUJI dataset.
image with the corresponding noisy observation in different ways [41], [178], [179], [180]. A simple approach is to add back the noisy observation \( \hat{Y} \) to the denoised image \( \hat{X} \) via
\[
\hat{Y} = \lambda \hat{X} + (1 - \lambda) \hat{X}
\]
where \( \lambda \in [0, 1] \) is a relaxation parameter reflecting the importance of \( Y \). The new noisy image \( \hat{Y} \) is then fed into the denoiser for final estimate. The iterative denoising strategies may significantly increase the computational burden, and the results reported in Table 7 and Table 9 show that a simple one-stage implementation such as MSI-SVD and BMXDI could produce competitive performance at moderate noise levels. It is therefore interesting to investigate how to efficiently handle severe noise with traditional denoisers. Recently, Zontak et al. [181] show that in the down-sampled noisy image, patches tend to be noiseless and share similar patterns with underlying clean patches. Therefore, instead of directly filtering the large-size noisy observation, an alternative is to first handle downsized image, and then upscale the denoised image back to its original size with some effective image super-resolution algorithms [182], [183]. This idea is similar to the encoder-decoder scheme of DNN methods. For simplicity we use MATLAB's built-in 'imresize' function and applies it to the efficient CMSt-SVD. This special implementation is termed as 'CMSt-SVD_R'. The visual effects of CMSt-SVD with and without the image resizing strategy are compared in Fig. 20.

As can be seen, the resizing strategy could effectively reduce color artifacts at the cost of slight over-smooth effects, thus it may be more effective and suitable for very noisy images with fewer fine details and textures.

![CMSt-SVD (a) Noisy (RENOIR) (b) CMSt-SVD (c) CMSt-SVD_R (d) Noisy (DND) (e) CMSt-SVD (f) CMSt-SVD_R](image)

Fig. 20. Visual evaluation of CMSt-SVD with and without the image resizing strategy on RENOIR and DND datasets.

5 Conclusion

Recently, multi-dimensional image denoising has become an attractive research topic with potentially important applications. The great success of the BM3D-based methods contributes significantly to the emergence of numerous related denoising methods, varying from traditional Gaussian denoisers to advanced DNN methods. In this paper, new benchmark datasets for color image and video are introduced, and comprehensive comparison of representative denoising methods are conducted with both synthetic and real-world experimental settings. Throughout extensive objective and subjective evaluations, we show: (i) the overall powerful denoising ability of the BM3D family, (ii) the use of tensor representation does not guarantee better results compared to matrix implementations, and (iii) the outstanding generalizability and denoising performance of some DNN methods such as FCCF, DRUNet, FastDVI/Net and VNLNet to real-world color image/video datasets.

Despite the achievements in recent years, the state-of-the-art approaches still suffer from limitations that restrict their application to key real-world scenarios. For example, from Fig. 9 and Fig. 10 it is noticed that many image denoising methods show only limited enhancement and may even degrade the image quality with over-smoothness and annoying artifacts. As image size grows rapidly with the development of new imaging technologies, a challenging problem is to decide if it is worth applying filtering to an image at hand [184], and if denoising is desired, which of the filters listed in Table 1 and Table 2 should be used. Recently, a new trend focuses on how to jointly handle image denoising and other computer vision tasks such as dehazing [185], denoising [186] and classification [187]. It is interesting to collect realistic benchmark datasets to further explore their mutual influence.

Appendix A

The Modified SVD Method

In this section, we introduce the motivation and details of the modified SVD (M-SVD) method briefly in Algorithm 1. For simplicity, our analysis is based on the color image denoising task. Given the local group \( G \in \mathbb{R}^{ps \times ps \times 3 \times K} \) of \( K \) patches, the original SVD algorithm operates directly on the matrix representation \( G(4) \) of \( G \) via
\[
G(4) = USV^T
\]
where \( S \) is singular value matrix, \( U \) and \( V \) are group and patch level transform matrices, respectively. A more effective way to obtain \( U \) is to perform SVD on the opponent color space [11], [19] represented by \( G_{opp} \) with
\[
G_{opp} = G(:,:,1,:) + G(:,:,2,:) + G(:,:,3,:)
\]
then the core matrix \( C \) of \( G(4) \) can be calculated by
\[
C = U^T G(4) V^T
\]
After the truncation of \( C \) and the inverse transform of \[13], the estimated clean group \( G_c = UC_{trunc} \) is obtained.

Appendix B

A Closer Look at the Effectiveness of HOSVD for Denoising

Our real-world experiments of color image and MRI denoising in section 4.3 and 4.4 show that the SVD-based methods could produce very competitive performance compared with HOSVD-based approaches, which are also prone to over-smooth textures. With such observation, we further investigate and discuss the effectiveness of HOSVD for denoising. Our analysis in this section is developed mainly based on the results of [34], [98], [104], [188], [189], [190], with more details available in the supplementary file.
Algorithm 1 M-SVD

Input: Noisy image \( \mathcal{A} \), number of similar patches \( K \).
Output: Estimated clean image \( \mathcal{A} \).

Step 1 (Grouping): For every reference patch of \( \mathcal{A} \), stack \( K \) similar patches in a group \( \mathcal{G} \in \mathbb{R}^{ps \times ps \times 3 \times K} \).

Step 2 (Collaborative filtering):
1. Obtain the group and patch level transform matrices \( U \) and \( V \) by performing SVD on \( G_{op}(p) \) and \( G_{inc}(p) \), respectively.
2. Apply the hard-threshold technique to \( C = U^T G_{inc}(p) V \) via \( C_{trun} = \text{hard-threshold}(C) \).
3. Take the inverse transform to obtain estimated clean group via \( \mathcal{G}_c = U C_{trun} V^T \).

Step 4 (Aggregation): Averagely write back all image patches in \( \mathcal{G}_c \) to their original locations according to Eq. (7).

B.1 The relationship between HOSVD and SVD

The relationship between HOSVD and SVD is established mainly based on the following theorems.

**Theorem 1.** Given an \( N \)-th order tensor \( \mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \), its full HOSVD is
\[
\mathcal{G} = C \times_1 U_1 \times_2 U_2 \times \cdots \times_N U_N \tag{14}
\]
where \( C \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \) represents core tensor. If \( S_i \) is the singular value matrix of \( G(i) \), then
\[
C(i)C^T(i) = S_i S_i^T \tag{15}
\]

**Theorem 2.** Given an \( N \)-th order tensor \( \mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \) and its core tensor \( C \) of the same size, with \( G(i) = U_i S_i V_i^T \) via SVD, then the \( j \)-th column of \( V_i \) can be represented as
\[
V_i(:,j) = U_i^T C(i)(j,:) / \| C(i)(j,:) \| \tag{16}
\]
where \( U_i = U_N \otimes \cdots \otimes U_{i+1} \otimes U_{i-1} \otimes \cdots \otimes U_1 \).

For simplicity, consider the case of third-order tensor group \( \mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times K} \) with \( K \) patches of size \( I_1 \times I_2 \), its full HOSVD is given by
\[
\mathcal{G} = C \times_1 U_1 \times_2 U_2 \times_3 U_3 \tag{17}
\]
The full SVD of the matrix representation \( G(3) \) is
\[
G(3) = U_3 S_3 V_3^T \tag{18}
\]

Theorem 1 indicates that truncating the columns of the group-level transform \( U_3 \) is equivalent to shrinking the rows of \( C(3) \) and also the singular value matrix \( S_3 \) of \( G(3) \). Theorem 2 demonstrates how the patch-level transform \( V_3 \) of \( G(3) \) can be derived from the first and second mode projection matrices \( U_1 \) and \( U_2 \) of \( \mathcal{G} \).

B.2 The risk of truncated HOSVD for denoising

HOSVD-based methods remove noise by truncating the columns of projection matrices via low-rank approximation, or by shrinking the core tensor via the hard-thresholding technique. To demonstrate the over-smooth effects of truncated HOSVD (T-HOSVD), we use a noise-free group \( \mathcal{G} \in \mathbb{R}^{3 \times 3 \times 3} \) consisting of three same patches with \( G(:, :, 1) = G(:, :, 2) = G(:, :, 3) \), and the third-mode unfolding of \( \mathcal{G} \) is
\[
G(3) = \begin{pmatrix}
3 & 1 & 5 & 6 & 4 & 8 & 9 & 2 & 6 \\
3 & 1 & 5 & 6 & 4 & 8 & 9 & 2 & 6 \\
3 & 1 & 5 & 6 & 4 & 8 & 9 & 2 & 6
\end{pmatrix} \tag{19}
\]

where every row of \( G(3) \) corresponds to a vectorized image patch. According to Eq. (17), the core tensor \( C \) is
\[
C(3) = \begin{pmatrix}
-27.98 & 0 & 0 & 0 & -5.38 & 0 & 0 & 0 & 2.07 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \tag{20}
\]

Assume the pre-defined multi-rank \( r_1 = r_2 = 2 \) is applied to truncate \( U_1 \) and \( U_2 \) with \( U = U(:, 1 : 2) \), then after the inverse transform of Eq. (17), the corresponding denoised group \( \mathcal{G}_c \) is
\[
\begin{pmatrix}
3.09 & 1.92 & 4.54 & 5.95 & 3.46 & 8.26 & 9.00 & 2.03 & 5.98 \\
3.09 & 1.92 & 4.54 & 5.95 & 3.46 & 8.26 & 9.00 & 2.03 & 5.98 \\
3.09 & 1.92 & 4.54 & 5.95 & 3.46 & 8.26 & 9.00 & 2.03 & 5.98
\end{pmatrix} \tag{21}
\]

Comparing Eq. (19) and Eq. (21), the loss of information is \( \| G(3) - G_c(3) \|_F^2 = 4.29 \). Also, from (20) it is noticed that the hard-thresholding technique in Eq. (17) will yield the same loss if the threshold parameter \( \tau \) is set to \( 2.07 < \tau < 5.38 \). Fig. 21 illustrates the over-smooth effects of T-HOSVD when applied to a noise-free image with all the patches in a group forced to be the same.

![Fig. 21. Illustration of over-smooth effects produced by truncated HOSVD (T-HOSVD) when all the patches in a group are the same.](image)

Our analysis in this section is based on an extreme and ideal case that all patches in a noise-free group are the same. In real-world applications, the difference among nonlocal similar patches is not negligible, and the use of tensor decomposition techniques may help filter out more redundant information and extract latent features, as is observed in the MRI denoising experiment illustrated in Fig. 15. Also, it is noticed from Fig. 21 that certain structural shapes and textures are preserved. However, despite recent development of the tensor theory [191], the choice of the best multi-rank and threshold parameter \( \tau \) for the tensor truncation strategy still remains a challenge.

ACKNOWLEDGMENT

The authors would like to thank those researchers who make their code, software packages and datasets publicly available. We also appreciate the generosity of our friends and volunteers for sharing their camera devices and participating in the video quality assessment task.
[49] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” IEEE Trans. Image Process., vol. 15, no. 12, pp. 3736–3745, 2006.
[50] J. Mairal, M. Elad, and G. Sapiro, “Sparse representation for color image restoration,” IEEE Trans. Image Process., vol. 17, no. 1, pp. 53–69, 2008.
[51] Y. Fu, A. Lam, I. Sato, and Y. Sato, “Adaptive spatial-spectral dictionary learning for hyperspectral image denoising,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 343–351.
[52] C. Liu and W. T. Freeman, “A high-quality video denoising algorithm based on reliable motion estimation,” in Proc. Eur. Conf. Comput. Vis., 2010, pp. 706–719.
[53] L. Zhuang and J. M. Bioucas-Dias, “Fast hyperspectral image estimation of space-time patches,” J. Math. Imag. Vis., vol. 60, no. 1, pp. 192–203, 2018.
[54] Y. Hou, J. Xu, M. Liu, G. Liu, L. Liu, F. Zhu, and L. Shao, “NLL: A blind pixel-level non-local method for real-world image denoising,” IEEE Trans. Image Process., vol. 29, pp. 5121–5135, 2020.
[55] R. H. Chan, C.-W. Ho, and M. Nikolova, “Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization,” IEEE Trans. Image Process., vol. 14, no. 10, pp. 1479–1495, 2005.
[56] M. Lebrun, M. Colom, and J.-M. Morel, “Multiscale image blind denoising,” IEEE Trans. Image Process., vol. 24, no. 10, pp. 3149–3161, 2015.
[57] L. Zhuang and J. M. Bioucas-Dias, “Fast hyperspectral image denoising and inpainting based on low-rank and sparse representations,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 11, no. 3, pp. 730–742, 2018.
[58] J. V. Manjón, P. Coupé, A. Buades, D. L. Collins, and M. Robles, “New methods for mri denoising based on sparseness and self-similarity,” Med. Image Anal., vol. 16, no. 1, pp. 18–27, 2012.
[59] P. Coupé, P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot, “An optimized blockwise nonlocal means denoising filter for 3-d magnetic resonance images,” IEEE Trans. Med. Imag., vol. 27, no. 4, pp. 475–484, 2008.
[60] J. V. Manjón, P. Coupé, L. Marti-Bonmati, D. L. Collins, and M. Robles, “Adaptive non-local means denoising of mr images with spatially varying noise levels,” J. Magn. Reson. Imag., vol. 31, no. 1, pp. 192–203, 2010.

[61] A. Foi and G. Boracchi, “Foveated nonlocal self-similarity,” Int. J. Comput. Vis., vol. 120, no. 1, pp. 78–110, 2016.
[190] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, “Tensor decompositions for signal processing applications: From two-way to multiway component analysis,” *IEEE Sig. Process. Mag.*, vol. 32, no. 2, pp. 145–163, 2015.

[191] A. Zhang and D. Xia, “Tensor svd: Statistical and computational limits,” *IEEE Trans. Inf. Theory*, vol. 64, no. 11, pp. 7311–7338, 2018.