Coarse to Fine Two-Stage Approach to Robust Tensor Completion of Visual Data

Yicong He and George K. Atia, *Senior Member, IEEE*

***Abstract*—Tensor completion is the problem of estimating the missing values of high-order data from partially observed entries. Data corruption due to prevailing outliers poses major challenges to traditional tensor completion algorithms, which catalyzed the development of robust algorithms that alleviate the effect of outliers. However, existing robust methods largely presume that the corruption is sparse, which may not hold in practice. In this article, we develop a two-stage robust tensor completion approach to deal with tensor completion of visual data with a large amount of gross corruption. A novel coarse-to-fine framework is proposed which uses a global coarse completion result to guide a local patch refinement process. To efficiently mitigate the effect of a large number of outliers on tensor recovery, we develop a new M-estimator-based robust tensor ring recovery method which can adaptively identify the outliers and alleviate their negative effect in the optimization. The experimental results demonstrate the superior performance of the proposed approach over state-of-the-art robust algorithms for tensor completion.**

***Index Terms*—Half-quadratic (HQ), robust method, tensor completion.**

I. Introduction

PREDICTING missing information from partially observed data is an emerging topic in modern data science due to unprecedented growth in data volume and dimensionality [1]. In multiway data analysis, where the data can be represented as a high-order tensor, this problem can be formulated in the lens of tensor completion with the goal of recovering the missing entries of a partially observed tensor. While the tensor completion problem is ill-posed without further model assumptions, actual formulations exploit the low-rank structure intrinsic to much of the real-world data. To date, numerous tensor completion algorithms have been proposed based on different tensor decomposition models [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], and have been successfully applied to a wide range of problems in computer vision [12], [13], pattern recognition [14], [15], and signal processing [16], [17].

In real applications, data may be corrupted by outliers due to human error or signal interference, making some of the observed data unreliable [18], [19]. Traditional tensor completion algorithms are largely based on a second-order measure of the error residuals, thus their performance degrades in the presence of outliers. In recent years, many works have focused on robust tensor completion and proposed several algorithms that were shown to outperform traditional completion algorithms in the presence of sparse outliers [20], [21], [22], [23], [24], [25]. Despite their robust performance with outlier-corrupted data compared with traditional methods, the usefulness of these algorithms is limited to settings with a small fraction of outliers. When the number of outliers increases, there are primarily two interrelated difficulties. First, the large number of outliers could overwhelm their underlying outlier rejection mechanism, leading to the severe performance degradation. For example, when the corruption is nonsparse, the ℓ_1-norm, which is at the heart of ℓ_1-norm-based robust tensor completion methods, falls short of accurately capturing the error residual. Second, the percentage of data entries to be relied on for completion of the missing entries decreases accordingly. For example, if 50% of the entries of a given tensor is observed, of which 60% is perturbed with outliers, then only 20% of the entries of the whole tensor is correctly observed. The reduced amount of reliable information for completion renders the tensor completion task more challenging, necessitating different means of completion.

To deal with tensor completion in the presence of a large number of outliers, we develop a novel two-stage coarse-to-fine framework for robust tensor completion. At the global coarse stage, a robust tensor completion algorithm is applied to the entire tensor to get a coarse completion result and identify a large number of outliers. At the local refinement stage, for each patch of the given tensor, a novel patch jitter procedure is proposed and used to construct a patch tensor using neighboring patches. Subsequently, robust tensor recovery incorporating the global coarse completion information is performed on the patch tensor, resulting in refined patch tensor recovery. In sharp contrast to existing nonlocal patch-based tensor completion algorithms [26], [27], [28], the proposed patch-based method does not perform block-matching, which greatly saves the computational cost and also avoids biased matching caused by outliers.

Further, to improve both the robustness and completion/refinement performance, we propose a new robust tensor ring (TR) recovery algorithm utilizing an M-estimator as the error measure. TR rank model has shown the desirable

Manuscript received 27 March 2022; revised 8 July 2022; accepted 7 August 2022. This work was supported in part by the National Science Foundation (NSF) CAREER under Award CCF-1552497 and Award CCF-2106339. This article was recommended by Associate Editor Q. Liu. *(Corresponding author: Yicong He.)*

Yicong He is with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816 USA (e-mail: yicong.he@ucf.edu).

George K. Atia is with the Department of Electrical and Computer Engineering and the Department of Computer Science, University of Central Florida, Orlando, FL 32816 USA (e-mail: george.atia@ucf.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2022.3198932.

Digital Object Identifier 10.1109/TCYB.2022.3198932

performance in many tensor completion tasks owing to its flexibility [8], [29]. M-estimators rooted in robust statistics are generalizations of maximum likelihood (ML) estimators for which the objective function is a sample average [30]. The selection of a proper loss function for M-estimators can greatly enhance robustness against large outliers. In order to handle the complex objective resulting from the use of an M-estimator, we leverage a half-quadratic (HQ) [31] minimization approach whereby the problem is reformulated as a reweighted TR completion program. Then, based on a TR unfolding scheme [9], [10], we develop a robust TR completion algorithm utilizing truncated singular value decomposition (SVD) to capture the low-rank structure. The proposed robust algorithms are efficient and have a simple structure owing to the use of an HQ-based method and a TR unfolding scheme, and can be applied to both the global tensor completion and local patch refinement stages. Further, the convergence of the proposed algorithm is analyzed. The following summarizes the main contributions of this article.

1) We propose a novel two-stage coarse-to-fine framework for robust tensor completion of visual data. First, we perform global coarse completion. Then, local patch refinement is applied to patch tensors created using patch jitter, where prior information from the global coarse completion result is incorporated.
2) We propose a new M-estimator-based TR recovery method for both global tensor completion and local patch refinement. An HQ approach is introduced to transform the nonconvex optimization problem to a reweighted tensor completion problem. Then, a new algorithm is developed based on a TR unfolding scheme and truncated SVD, and its convergence is analyzed.
3) We perform experiments on real data for image and video completion, demonstrating the superior performance of the proposed algorithm compared with existing robust tensor completion algorithms.

This article is organized as follows. In Section II, we present the related work in matrix and tensor completion. In Section III, we briefly introduce our notation and provide some preliminary background on TR decomposition and completion. In Section IV, we present our two-stage coarse-to-fine tensor completion framework, along with the formulation of the objective function for each stage. In Section V, we propose our new HQ-based robust TR recovery algorithm. Experimental results are presented in Section VI to demonstrate the completion performance. Finally, the conclusion is given in Section VII.

II. Related Work

Low-Rank Matrix and Tensor Completion: Matrix or tensor completion aim to fill the missing entries of a partially observed matrix or tensor data. The key idea underlying the ability to estimate their unknown entries is the low-rank property inherent in many machine learning problems [32], [33], [34], [35], which captures the redundancy and correlation within a matrix or tensor [36].

Different from the matrix domain where the rank is uniquely defined, the rank of a tensor has various definitions corresponding to different tensor factorization (decomposition) models, such as CANDECOMP/PARAFAC (CP) [37], Tucker [38], tensor SVD (t-SVD) [39], TR [29], and tensor train (TT) [40]. Based on different tensor decomposition models, a large number of tensor completion algorithms was developed [2], [3], [4], [5], [6], [7], [8], [9], [10], [11] and achieve the desirable performance in noiseless environments and Gaussian noise with small variance. However, when the data are contaminated with large outliers, the performance of these traditional algorithms is unsatisfactory in general. This spurred further research on robust matrix and tensor completion, which is the main focus of our work.

Robust Matrix and Tensor Completion: In robust matrix and tensor completion, the goal is to recover the low-rank matrix or tensor from corrupted partial observations. Following the method of robust principal component analysis (RPCA) [41], [42], a matrix or tensor can be completed by decomposing it into the sum of low-rank and sparse components. The low-rank component represents the actual noise-free low-rank matrix or tensor and the sparse component models the sparse outliers.

The ℓ_1-norm is widely utilized to constrain the sparse component [32], [43], and a number of ℓ_1-norm-based robust completion algorithms has been proposed for different decomposition (factorization) models, such as matrix factorization [44], Tucker [20], TR [22], and t-SVD [23]. Other algorithms impose more flexible ℓ_p-norm and $\ell_{p,\epsilon}$-norm constraints on the sparse component instead of the ℓ_1-norm [24], [25].

The success of the aforementioned existing robust methods is largely dependent on the assumption that the outliers are sparse—otherwise, their performance may greatly degrade. In this article, we develop a new two-stage framework with a new robust tensor completion algorithm to improve the performance under heavy noise and data corruption.

Patch-Based Matrix and Tensor Completion: To further improve the completion performance on visual data, the patch-based method has been recently introduced to matrix and tensor completion. Similar to patch-based image and video denoising methods [45], [46], these methods apply block-matching 3-D (BM3D) [45] or BM4D [46] to find similar patches across the spatial domain. Then, the completion is applied to matrices or tensors constructed from similar patches. A variety of patch-based completion algorithms incorporating different matrix and tensor completion methods has been developed, such as [27], [28], [47]. Also, in [48], a tensor completion algorithm is proposed utilizing local and nonlocal patch completion as the regularization terms of the global tensor completion.

Similar to the traditional completion algorithms, current patch-based methods also suffer from the performance degradation in the presence of outliers. One idea would be to replace the completion methods in a path-based framework with robust ones. However, directly applying block-matching to robust tensor completion tasks may result in inaccurate matching results due to missing entries and outliers, which in turn affects the completion performance. In this article, we propose an efficient method called "patch jitter" to directly bypass the block-matching procedure. Further, the global completion result is incorporated into local patch refinement to further improve the performance.

III. Preliminary

Notation: In this article, we adopt tensor notation similar to [9], [10]. Uppercase calligraphic letters are used to denote tensors (e.g., $\mathcal{X}$), uppercase boldface letters for matrices (e.g., $\mathbf{X}$), lowercase boldface letters for vectors (e.g., $\mathbf{x}$) and lowercase letters for scalars (e.g., x). An N-order tensor is defined as $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, where $I_i, i = 1, \ldots, N$ is the dimension of the ith way of the tensor. $\mathcal{X}_{i_1 \ldots i_N}$ denotes the $(i_1, i_2, \ldots, i_N)$-th entry of tensor $\mathcal{X}$, and $\mathbf{X}_{i,j}$ denotes the (i, j)th entry of matrix $\mathbf{X}$. The Frobenius norm of a tensor is defined as $\|\mathcal{X}\|_F = \sqrt{\sum_{i_1 \ldots i_N} |\mathcal{X}_{i_1 \ldots i_N}|^2}$. The product $\mathcal{A} \circ \mathcal{B}$ denotes the Hadamard (element-wise) product of tensors $\mathcal{A}$ and $\mathcal{B}$. For a scalar n, $[n] := \{1, \ldots, n\}$.

A. Tensor Ring Model

We briefly review the definition of TR decomposition.

Definition 1 ([29, Sec. 2] TR Decomposition): Given TR rank $[r_1, \ldots, r_N]$, in TR decomposition a high-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is represented as a sequence of circularly contracted 3-order core tensors $\mathcal{U}_k \in \mathbb{R}^{r_k \times I_k \times r_{k+1}}, k = 1, \ldots, N$, with $r_{N+1} = r_1$. Specifically, the element-wise relation of tensor $\mathcal{X}$ and its TR core tensors $\{\mathcal{U}_k\}_{k=1}^N$ is defined as

$$\mathcal{X}_{i_1 \ldots i_N} = \mathrm{Tr}\left(\prod_{k=1}^{N} \mathcal{U}_k(:, i_k, :)\right)$$

where $\mathcal{U}_k(:, i_k, :) \in \mathbb{R}^{r_k \times r_{k+1}}$ is the i_k-th slice matrix of $\mathcal{U}_k$ along mode-2, and $\mathrm{Tr}(\cdot)$ is the matrix trace operator.

Based on the definition above, Huang *et al.* [9] and Yu *et al.* [10] proposed a new circular TR unfolding scheme, defined below.

Definition 2 ([9, Sec. 2.2] Tensor Ring Unfolding): Given an N-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, its TR unfolding $\mathbf{X}_{\langle k,d \rangle} \in \mathbb{R}^{\prod_{i=k}^{k+d-1} I_i \times \prod_{j=k+d}^{k+N-1} I_j}$, $k, d \in \{1, \ldots, N\}$, is a matrix whose entries are defined through the relation $(\mathbf{X}_{\langle k,d \rangle})_{s,t} = \mathcal{X}_{i_1 \ldots i_N}$ with

$$s = 1 + \sum_{c=k}^{k+d-1} (i_c - 1) \prod_{j=k}^{c-1} I_j, \quad t = 1 + \sum_{c=k+d}^{k+N-1} (i_c - 1) \prod_{j=k+d}^{c-1} I_j$$

where $I_{k+N} = I_k, i_{k+N} = i_k$ for $1 \leq k \leq N$. In practice, $\mathbf{X}_{\langle k,d \rangle}$ can be generated by first permuting $\mathcal{X}$ with order $[k, \ldots, N, 1, \ldots, k-1]$, then performing unfolding along the first d modes.

Theorem 1 [10, Sec. 2]: Assume $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is Nth-order tensor with TR rank $[r_1, r_2, \ldots, r_N]$, then for each unfolding matrix $\mathbf{X}_{\langle k,d \rangle}$

$$\mathrm{rank}(\mathbf{X}_{\langle k,d \rangle}) \leq r_k r_{k+d} \tag{1}$$

with $r_{i+N} = r_i, i = 1, \ldots, N$, where $\mathrm{rank}(\mathbf{X})$ denotes the rank of matrix $\mathbf{X}$.

B. Tensor Ring Completion

Given an N-order tensor $\mathcal{M} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, and index set $\Omega \subseteq [I_1] \times \cdots \times [I_N]$, TR completion is the problem of filling in the missing entries of tensor $\mathcal{M}$ using the observed entries indexed by set Ω and the low-rank property. This problem can be formulated as

$$\min_{\mathcal{X}} \mathrm{rank}_{\mathrm{tr}}(\mathcal{X}), \text{ s.t. } \mathcal{P} \circ \mathcal{X} = \mathcal{P} \circ \mathcal{M} \tag{2}$$

where the mask tensor $\mathcal{P} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is set as

$$\mathcal{P}_{i_1 \ldots i_N} = \begin{cases} 1, & (i_1, \ldots, i_N) \in \Omega \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

and $\mathrm{rank}_{\mathrm{tr}}(\mathcal{X})$ denotes the TR rank of tensor $\mathcal{X}$. According to (1), one can further solve the TR completion task using the following optimization problem [9], [22]:

$$\min_{\mathcal{X}} \sum_{k=1}^{N} \beta_k \, \mathrm{rank}(\mathbf{X}_{\langle k,d \rangle}), \text{ s.t. } \mathcal{P} \circ \mathcal{X} = \mathcal{P} \circ \mathcal{M} \tag{4}$$

where $\{\beta_k\}_{k=1}^N$ are weight parameters.

IV. Two-Stage Coarse-to-Fine Robust Tensor Completion Framework

Our goal is to perform robust tensor completion of visual data with a large number of outliers. To this end, we develop a two-stage coarse-to-fine tensor completion framework, illustrated in Fig. 1. Given a noisy, partially observed image tensor $\mathcal{M} \in \mathbb{R}^{I_1 \times I_2 \times n}$ (n is 1 and 3 for gray and color images, respectively), in the first (global) stage, a robust TR completion algorithm is applied to the entire tensor, yielding a coarse completion result. In the second (local) stage, we first divide the tensor into overlapping patches of size $m \times m \times n$ with overlap pixels o. Then, with the guidance of the global completion result, we perform local patch-based robust TR refinement on each patch tensor constructed using a patch jitter. The final completion result is obtained by aggregating the refined local patches. In the following, we describe each stage. The details of the robust recovery algorithms are discussed in Section V.

A. Global Robust Tensor Completion With M-Estimator and Tensor Ring Rank

In robust tensor completion, the predominant measure of error is the ℓ_1-norm of the error residual [20], [22], [23]. The ℓ_1-norm-based tensor completion algorithms aim to solve

$$\min_{\mathcal{X}} \mathrm{rank}_t(\mathcal{X}) + \lambda \|\mathcal{E}\|_1, \text{ s.t. } \mathcal{P} \circ (\mathcal{X} + \mathcal{E}) = \mathcal{P} \circ \mathcal{M} \tag{5}$$

where $\mathrm{rank}_t(\mathcal{X})$ denotes the rank of tensor $\mathcal{X}$, which varies depending on different definitions of the tensor rank. For the optimization problem (5), it is always assumed that the error term $\mathcal{E}$ is sparse, that is, there are only a few outliers. In the case, where a large number of observed entries is perturbed by outliers, $\mathcal{E}$ is no longer sparse in general, and a solution to (5) is unreliable.

The M-estimator has been widely used in robust statistics due to its robustness to outliers. Given a tensor $\mathcal{X}$, its M-estimator $F(\mathcal{X})$ can be formed as a sum of functions

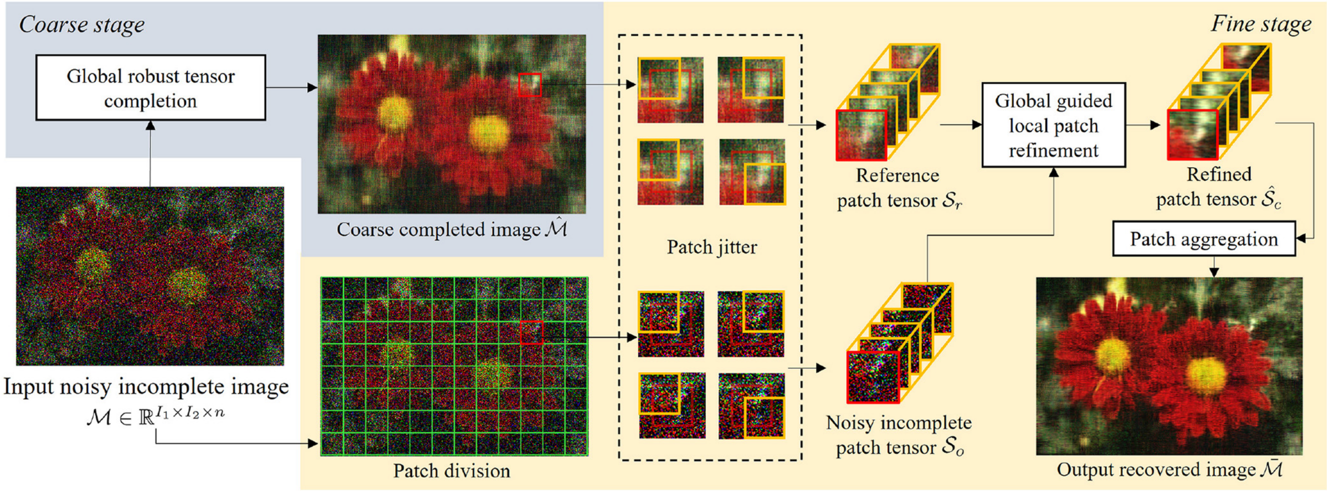

Coarse stage
Global robust tensor completion
Coarse completed image $\hat{\mathcal{M}}$
Input noisy incomplete image
$\mathcal{M} \in \mathbb{R}^{I_1 \times I_2 \times n}$
Patch division
Patch jitter
Reference patch tensor $\mathcal{S}_r$
Noisy incomplete patch tensor $\mathcal{S}_o$
Global guided local patch refinement
Fine stage
Refined patch tensor $\hat{\mathcal{S}}_c$
Patch aggregation
Output recovered image $\bar{\mathcal{M}}$

Fig. 1. Proposed two-stage coarse-to-fine robust tensor completion framework for visual data. In the coarse stage (blue region), the robust tensor completion algorithm is applied to the entire image. In the fine stage (yellow region), for each patch in the divided image, local patch refinement guided by the global completion result is applied to a corresponding patch tensor obtained from the patch jitter.

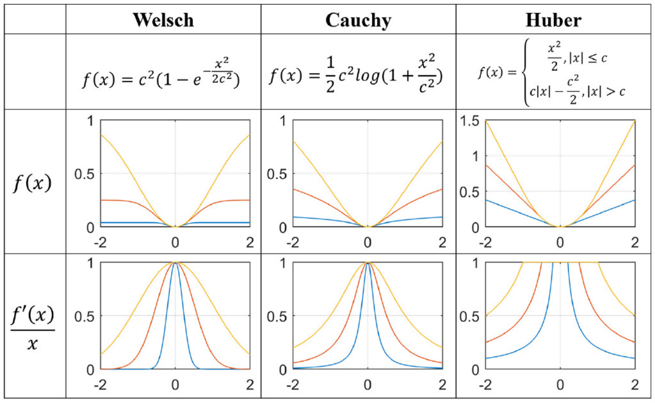


Fig. 2. Illustration of loss functions of M-estimators (top) and corresponding weight function (bottom) with different shape parameter c (Blue: $c = 0.2$, red: $c = 0.5$, yellow: $c = 1$).

of the data, that is, $F(\mathcal{X}) = \sum_{i_1 \ldots i_N} f(\mathcal{X}_{i_1 \ldots i_N})$, where $f(.)$ is a function with certain properties [30]. Compared with the ℓ_1-norm, the M-estimator is differentiable at 0, and is more flexible with different choices of a shape parameter (see Fig. 2). In this work, we introduce an M-estimator with adaptive parameter selection to the robust TR completion task.

By introducing the M-estimator in (4), we obtain the unconstrained M-estimator-based robust TR completion optimization problem

$$\min_{\mathcal{X}} \sum_{k=1}^{N} \beta_k \operatorname{rank}(\mathbf{X}_{\langle k,d \rangle}) + \lambda \sum_{i_1 \ldots i_N} \mathcal{P}_{i_1 \ldots i_N} f(\mathcal{E}_{i_1 \ldots i_N}) \tag{6}$$

where $\mathcal{E}_{i_1 \ldots i_N} = \mathcal{X}_{i_1 \ldots i_N} - \mathcal{M}_{i_1 \ldots i_N}$.

In our work, we use three functions for M-estimators: Huber function, Welsch function [49], and Cauchy function shown in Fig. 2. The Welsch and Cauchy functions yield a type of redescending M-estimators, which also satisfy that $\lim_{x \to +\infty} f'(x) = 0$. In [50], a redescending M-estimator is introduced for low-Tucker-rank tensor completion, solved using a block coordinate descent method. However, the Tucker-rank-based method is not applicable to TR-based methods due to the difference in the rank model. Also, its performance is limited by the low convergence speed of the gradient-based method. In the next section, we will develop a more general and efficient solution using an HQ method for the TR model.

The global completion can identify most of the reliable observed entries, that is, the clean unperturbed observed entries and the observed entries with small perturbation. However, the global completion performance may still be limited due to insufficient reliable information for completion or disturbance by a small number of unrecognized noisy entries. On the other hand, patch-based methods can yield the better completion performance than global ones by performing completion on similar patches [27], [28]. To further improve the completion performance, we propose a new refinement process on local patches which incorporates both global and local information. In the following parts, we will introduce our proposed local-based method.

B. Local Patch Tensor Construction Using Patch Jitter

Patch-based methods have been widely used in visual data processing [45], [46]. In tensor completion, a patch tensor is created using block-matching across the spatial domain [26], [27], [28]. Existing block-matching methods presume that the data entries are not perturbed by outliers, such that similar patches can be accurately matched. Similar patches are then stacked to a patch tensor, on which completion can be applied. However, in our setting, the distance between patches will be biased due to the existence of outliers, which would deteriorate the results of block-matching.

By contrast, instead of using block-matching to find similar patches, in this work, we directly apply patch jitter on each patch to generate a patch tensor. Specifically, given a patch of size $m \times m \times n$, we generate its neighboring patches with jitter length l, that is, the $(2l + 1)^2$ number of patches in a window of size $(2l + 1) \times (2l + 1)$ centered at the original patch. Note

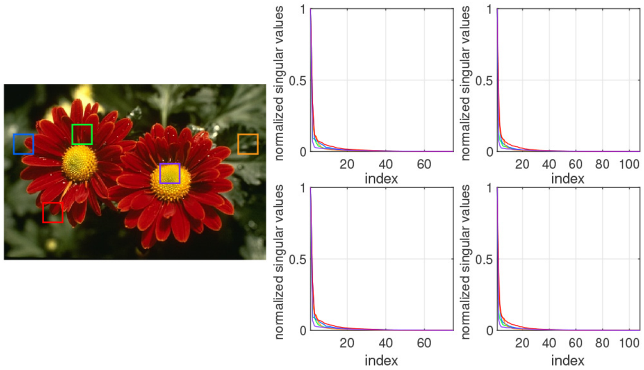


Fig. 3. Left: image "flower" from Berkeley Segmentation Dataset [51], rectangles: five selected patches with size 36 × 36 × 3. Right: normalized singular values of TR unfolding matrices of the five patch tensors obtained using patch jitter with $l = 2$. Top-left: $k = 1, d = 2$, top-right: $k = 2, d = 2$, bottom-left: $k = 3, d = 2$, and bottom-right: $k = 4, d = 2$.

that the original patch is also included in the neighboring patch set. Then, the $(2l+1)^2$ patches are stacked in a patch tensor $\mathcal{S} \in \mathbb{R}^{m\times m\times n\times(2l+1)^2}$. To match the patches at the corners and boundaries of the frames, the tensor $\mathcal{M}$ is first padded by mirroring l pixels at all boundaries and corners, resulting in a padded tensor $\mathcal{M}_p$ of size $(I_1+2l)\times(I_2+2l)\times n$.

We briefly give insight into the patch jitter procedure. An example of a patch jitter with a fully observed image without outliers is shown in Fig. 3. We pick five patches at different locations (marked by green rectangles). For each patch, the patch tensor is created using patch jitter with $l = 2$. The normalized singular values (with respect to maximum singular value) of TR unfolding matrices of the patch tensors are shown in the right of Fig. 3. As can be seen, with a small offset around the original patch, the obtained patch tensor can be well approximated by a low TR rank tensor. Therefore, for a partially observed image, the patch tensor generated using a patch jitter can be well completed using a TR completion algorithm. Moreover, compared with traditional patch-based methods, the jitter operation does not require block-matching, thereby avoiding incorrect matching due to outliers and greatly reducing the computational cost.

C. Global Completion Guided Local Patch Tensor Refinement

After constructing patch tensors using the patch jitter, we apply a local patch refinement process to each patch tensor. We utilize the global coarse completion result to help identify the outliers in the patch tensor, as well as give a good initialization to the missing entries of the patch tensor.

Assume we have obtained the (coarsely) completed tensor (denoted $\hat{\mathcal{M}}$) from the tensor $\mathcal{M}$ using (6). Given a patch tensor $\mathcal{S}_o \in \mathbb{R}^{m\times m\times n\times(2l+1)^2}$ from $\mathcal{M}_p$, we extract the patches from the same locations in the padded completed tensor $\hat{\mathcal{M}}_p$ and stack them to a reference patch tensor $\mathcal{S}_r$. Then, the missing entries in $\mathcal{S}_o$ are filled with corresponding entries in $\mathcal{S}_r$, resulting in a combined tensor $\mathcal{S}_c$ with entries

$$(\mathcal{S}_c)_{i_1\ldots i_N} = \begin{cases} (\mathcal{S}_o)_{i_1\ldots i_N}, & (i_1,\ldots,i_N)\in\Omega_s \\ (\mathcal{S}_r)_{i_1\ldots i_N}, & \text{otherwise} \end{cases} \tag{7}$$

where Ω_s denotes the observation index set of $\mathcal{S}_o$.

In order to represent the different confidence levels of each entry, we use a soft weighting strategy [52] in which we assign different weights to each entry of the combined patch tensor $\mathcal{S}_c$. In particular, defining a weight tensor $\mathcal{W}$ with the same size as $\mathcal{S}_c$, each element of $\mathcal{W}$ is obtained as

$$\mathcal{W}_{i_1\ldots i_N} = \begin{cases} \exp\left(-\dfrac{\left((\mathcal{S}_c)_{i_1\ldots i_N}-(\mathcal{S}_r)_{i_1\ldots i_N}\right)^2}{2\sigma^2}\right), & (i_1,\ldots,i_N)\in\Omega_s \\ v, & \text{otherwise} \end{cases} \tag{8}$$

where σ is a parameter controlling the similarity. Specifically, for entries of $\mathcal{S}_c$ that are originally observed (i.e., $(i_1,\ldots,i_N)\in\Omega_s$), the weight is assigned in terms of its distance from the corresponding entry in the reference patch tensor $\mathcal{S}_r$. For the entries of $\mathcal{S}_c$ that were filled from $\mathcal{S}_r$, the weights are all set to some value v.

Inspired by adaptive parameter selection for the M-estimator which will be presented in Section V-B, we adaptively determine $\sigma_{\min}$ and $v_{\max}$ using

$$\begin{aligned} \sigma &= \max\left\{\eta_\sigma\left(\max\{(\mathbf{d}_{\Omega_s})_{(0.25)},(\mathbf{d}_{\Omega_s})_{(0.75)}\}\right)^{-1},\sigma_{\min}\right\} \\ v &= \min\left\{\eta_v\max\{(\mathbf{d}_{\Omega_s})_{(0.25)},(\mathbf{d}_{\Omega_s})_{(0.75)}\},v_{\max}\right\} \end{aligned} \tag{9}$$

where $\mathbf{d}_{\Omega_s}\in\mathbb{R}^{|\Omega_s|\times 1}$ denotes the vector composed of entries $\mathcal{D}_{i_1\ldots i_N} = (\mathcal{S}_c)_{i_1\ldots i_N} - (\mathcal{S}_r)_{i_1\ldots i_N}$, $(i_1,\ldots,i_N)\in\Omega_s$, $\mathbf{y}_{(q)}$ denotes the qth quantile of $\mathbf{y}$, η_σ and η_v are free parameters to be chosen, $\sigma_{\min}$ is a lower bound on σ, and $v_{\max}$ is an upper bound on v.

We can readily formulate the local patch refinement as a weighted robust tensor recovery problem

$$\min_{\mathcal{X}}\sum_{k=1}^{N}\beta_k\operatorname{rank}(\mathbf{X}_{\langle k,d\rangle}) + \lambda\sum_{i_1\ldots i_N}\mathcal{W}_{i_1\ldots i_N}f\left((\mathcal{E}_c)_{i_1\ldots i_N}\right) \tag{10}$$

where $(\mathcal{E}_c)_{i_1\ldots i_N} = \mathcal{X}_{i_1\ldots i_N} - (\mathcal{S}_c)_{i_1\ldots i_N}$.

Note that (10) can be obtained by replacing the binary indicator tensor $\mathcal{P}$ in (6) with a weight tensor $\mathcal{W}$ with entries from [0, 1]. Therefore, (6) can be viewed as a special case of (10) with binary weights. In Section V, we propose a half quadratic-based algorithm that can solve both (6) and (10).

The completion and refinement processes of the proposed framework are summarized in Algorithm 1. We remark that the framework can also be extended to video data, where an additional temporal dimension is added. In this case, for a video with f frames, the patch will be of size $m\times m\times n\times f$ and the corresponding patch tensor is a 5th-order tensor of size $m\times m\times n\times f\times(2l+1)^2$.

V. Half-Quadratic Approach to Weighted Robust Tensor Recovery

In this section, we aim to solve the following optimization problem, which combines both (6) and (10):

$$\min_{\mathcal{X}}\Phi(\mathcal{X}) + \lambda\sum_{i_1\ldots i_N}\mathcal{W}_{i_1\ldots i_N}f(\mathcal{E}_{i_1\ldots i_N}) \tag{11}$$

where $\Phi(\mathcal{X}) = \sum_{k=1}^{N}\beta_k\operatorname{rank}(\mathbf{X}_{\langle k,d\rangle})$, and the entries of $\mathcal{W}_{i_1\ldots i_N}$ are in the range [0, 1] with $\mathcal{W}_{i_1\ldots i_N} = 0$,

Algorithm 1 Coarse-to-Fine Robust Tensor Ring Completion (C2FRTRC)

```
Input: Partially observed image tensor $\mathcal{M} \in \mathbb{R}^{I_1 \times I_2 \times n}$, patch generation parameters $m$, $o$, patch jitter parameter $l$, parameters $\eta_\sigma$, $\eta_v$, $\sigma_{\min}$ and $v_{\max}$.
Complete $\mathcal{M}$ using (6) and obtain the completed tensor $\hat{\mathcal{M}}$.
Pad the tensors $\mathcal{M}$ and $\hat{\mathcal{M}}$ to $\mathcal{M}_p$ and $\hat{\mathcal{M}}_p$.
Divide $\mathcal{M}_p$ into patches according to size $m$ and overlap pixels $o$ and create patch set $\mathcal{I}$.
for each patch in $\mathcal{I}$ do
    Construct patch tensor $\mathcal{S}_o$ using $(2l+1)^2$ neighbor patches around the location in $\mathcal{M}_p$.
    Construct reference patch tensor $\mathcal{S}_r$ using $(2l+1)^2$ neighbor patches around the location in $\hat{\mathcal{M}}_p$.
    Construct combined tensor $\mathcal{S}_c$ using $\mathcal{S}_o$ and $\mathcal{S}_r$, and obtain refined patch tensor $\hat{\mathcal{S}}_c$ by solving (10).
end for
Obtain refined completed image tensor $\bar{\mathcal{M}}$ by aggregating patches from all patch tensors $\hat{\mathcal{S}}_c$ according to $\mathcal{I}$ and removing padded border pixels.
Output: Completed image tensor $\bar{\mathcal{M}}$.
```

$(i_1, \ldots, i_N) \notin \Omega$. We develop an HQ-based approach to efficiently solve (11). We also propose an adaptive parameter selection strategy and discuss the property of the HQ-based solution.

A. Half-Quadratic Approach to the Non-Convex Program

We present an HQ method to solve the M-estimator-based optimization problem in (11). HQ methods have been broadly applied in nonquadratic optimization [53]. Instead of directly optimizing a complex nonquadratic objective, HQ transforms the nonquadratic loss function to an HQ one. Specifically, there exists a strictly convex and decreasing dual function $\varphi(.)$ such that minimizing the loss function $f(t)$ with respect to (w.r.t.) t is equivalent to minimizing an augmented cost function in an enlarged parameter space $\{t, q\}$, that is, [53], [54]

$$\min_t f(t) = \min_{t,q} \frac{1}{2} q t^2 + \varphi(q). \tag{12}$$

Therefore, by substituting (12) in the M-estimator, the minimization of function $\sum_{i_1 \ldots i_N} \mathcal{W}_{i_1 \ldots i_N} f(\mathcal{E}_{i_1 \ldots i_N})$ becomes

$$\begin{aligned} &\min_{\mathcal{X}} \sum_{i_1 \ldots i_N} \mathcal{W}_{i_1 \ldots i_N} f(\mathcal{E}_{i_1 \ldots i_N}) \\ &\quad = \min_{\mathcal{X}, \mathcal{Q}} \sum_{i_1 \ldots i_N} \left(\frac{1}{2} \mathcal{W}_{i_1 \ldots i_N} \mathcal{Q}_{i_1 \ldots i_N} \mathcal{E}^2_{i_1 \ldots i_N} + \mathcal{W}_{i_1 \ldots i_N} \varphi(\mathcal{Q}_{i_1 \ldots i_N}) \right). \end{aligned} \tag{13}$$

Hence, (10) is equivalent to the following problem:

$$\min_{\mathcal{X}, \mathcal{Q}} \Phi(\mathcal{X}) + \frac{\lambda}{2} \left\| \sqrt{\mathcal{W}} \circ \sqrt{\mathcal{Q}} \circ (\mathcal{X} - \mathcal{M}) \right\|_F^2 + \lambda \Psi_{\mathcal{W}}(\mathcal{Q}) \tag{14}$$

where $\Psi_{\mathcal{W}}(\mathcal{Q}) = \sum_{i_1 \ldots i_N} \mathcal{W}_{i_1 \ldots i_N} \varphi(\mathcal{Q}_{i_1 \ldots i_N})$.

The problem above is a reweighted TR completion problem, and one could use alternating minimization to solve it. Specifically, by fixing tensor $\mathcal{X}$, tensor $\mathcal{Q}$ can be found by solving (14) with fixed residual $\mathcal{E}$. According to [54, Th. 1], the optimal solution q^* in the right hand side (RHS) of (12) can be obtained as $q^* = ([f'(t)]/t)$. Thus, we obtain each entry $\mathcal{Q}_{i_1 \ldots i_N}$ as

$$\mathcal{Q}_{i_1 \ldots i_N} = \frac{f'(\mathcal{E}_{i_1 \ldots i_N})}{\mathcal{E}_{i_1 \ldots i_N}}. \tag{15}$$

Subsequently, given a fixed $\mathcal{Q}$, (14) becomes the double-weighted tensor completion problem

$$\min_{\mathcal{X}} \Phi(\mathcal{X}) + \frac{\lambda}{2} \left\| \sqrt{\mathcal{W}} \circ \sqrt{\mathcal{Q}} \circ (\mathcal{X} - \mathcal{M}) \right\|_F^2. \tag{16}$$

The weighting tensor $\mathcal{Q}$ assigns different weights to each observed entry based on the error residual tensor $\mathcal{E}$. Fig. 2 depicts the weights in terms of the error, x, for different loss functions. One can observe that given a proper shape parameter c, a large error may lead to a small weight, so that the error statistics will not be unduly affected by large outliers. Specifically, when $c \to +\infty$, all entries of $\mathcal{Q}$ will be equal to 1 and (14) reduces to a traditional second-order statistics-based completion method. In this case, the algorithm cannot alleviate the effect of outliers since all error residuals are treated equally.

B. Adaptive Parameter Selection for M-Estimator

Most M-estimators, such as Huber, Cauchy, and Welsch have a parameter c to control the shape of the loss. Per the previous discussion, the weights $\mathcal{Q}$ based on the error residual play an important role in recognizing the outliers. As Fig. 2 shows a relatively smaller c can better reduce the effect of outliers. However, in practice convergence will be slower if c is set to a small fixed value. Therefore, to improve both efficiency and accuracy, we use an adaptive kernel width selection method for the M-estimator. Specifically, the shape parameter is determined by

$$c = \max\left\{\eta_c \max\{(\mathbf{e}_\Omega)_{(0.25)}, (\mathbf{e}_\Omega)_{(0.75)}\}, c_{\min}\right\} \tag{17}$$

where $\mathbf{e}_\Omega \in \mathbb{R}^{|\Omega| \times 1}$ denotes the vector composed of entries $\mathcal{E}_{i_1 \ldots i_N}, i_1 \ldots i_N \in \Omega$. The parameter η_c controls the range of outliers, and $c_{\min}$ is a lower bound on c. Using the adaptive method above, c is set to a relatively large value in the beginning to speed up convergence. As the convergence speed reduces, c is decreased correspondingly and the effect of the outliers is gradually reduced.

C. Truncated SVD-Based Algorithm

To solve (14), we define the indicator function for $\mathbf{X}_{\langle k,d \rangle}, k = 1, \ldots, N$ as [50]

$$\delta(\mathbf{X}_{\langle k,d \rangle}) = \begin{cases} 0, & \text{if } \operatorname{rank}(\mathbf{X}_{\langle k,d \rangle}) \le r_{kd} \\ +\infty, & \text{otherwise} \end{cases} \tag{18}$$

where $r_{kd} = r_k r_{k+d}$. Thus, the minimization is expressed as

$$\min_{\mathcal{X}, \mathcal{Q}} \sum_{k=1}^{N} \beta_k \delta\big(\mathbf{X}_{\langle k,d \rangle}\big) + \frac{\lambda}{2} \left\| \sqrt{\mathcal{W}} \circ \sqrt{\mathcal{Q}} \circ (\mathcal{M} - \mathcal{X}) \right\|_F^2 + \lambda \Psi_{\mathcal{W}}(\mathcal{Q}). \tag{19}$$

We devise an alternating direction method of multipliers (ADMMs) method to solve (19). In particular, we introduce the dual variables $\{\mathcal{Z}^{(k)}\}_{k=1}^{N}$ and rewrite (19) as

$$\min_{\mathcal{X},\mathcal{Q},\mathcal{Z}^{(k)}} \sum_{k=1}^{N} \beta_k \delta\Big(\mathbf{Z}^{(k)}_{\langle k,d\rangle}\Big) + \frac{\lambda}{2}\Big\|\sqrt{\mathcal{W}}\circ\sqrt{\mathcal{Q}}\circ(\mathcal{M}-\mathcal{X})\Big\|_F^2 + \lambda\Psi_{\mathcal{W}}(\mathcal{Q}) \quad \text{s.t.}\, \mathcal{Z}^{(k)}=\mathcal{X},\ k=1,\ldots,N. \tag{20}$$

The augmented Lagrangian function can be written as

$$\begin{aligned}&\mathscr{L}_{\mu}(\mathcal{X},\mathcal{Q},\mathcal{Z}^{(1)},\ldots,\mathcal{Z}^{(N)},\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(N)})\\ &= \sum_{k=1}^{N}\bigg(\beta_k\delta(\mathbf{Z}^{(k)}_{\langle k,d\rangle}) + \langle\mathcal{G}^{(k)},\mathcal{Z}^{(k)}-\mathcal{X}\rangle + \frac{\mu}{2}\Big\|\mathcal{Z}^{(k)}-\mathcal{X}\Big\|_F^2\bigg)\\ &\quad+\frac{\lambda}{2}\Big\|\sqrt{\mathcal{W}}\circ\sqrt{\mathcal{Q}}\circ(\mathcal{M}-\mathcal{X})\Big\|_F^2+\lambda\Psi_{\mathcal{W}}(\mathcal{Q})\end{aligned} \tag{21}$$

where $\{\mathcal{G}^{(k)}\}_{k=1}^{N}$ is the dual variables and μ is the step size. One can alternatively update each variable while fixing the others.

1) *Update c and $\mathcal{Q}$:* First, c is estimated using (17). Then, for each element $\mathcal{Q}_{i_1\ldots i_N}$, the optimal solution can be directly obtained using (15).

2) *Update $\mathcal{Z}^{(k)}$:* For each $\mathcal{Z}^{(k)}, k=1,\ldots,N$, the optimal solution can be obtained by solving

$$\mathcal{Z}^{(k)} = \arg\min_{\mathcal{Z}} \left(\left\|\mathcal{Z}-\left(\mathcal{X}-\frac{1}{\mu}\mathcal{G}^{(k)}\right)\right\|_F^2\right) \quad \text{s.t. } \operatorname{rank}(\mathbf{Z}_{\langle k,d\rangle}) \le r_{kd}. \tag{22}$$

This is a low-rank approximation problem which has an optimal solution [55]

$$\mathcal{Z}^{(k)} = \text{fold}_{\langle\text{k,d}\rangle}\left(\Pi_{r_{kd}}\left(\mathbf{X}_{\langle k,d\rangle}-\frac{1}{\mu}\mathbf{G}^{(k)}_{\langle k,d\rangle}\right)\right) \tag{23}$$

where $\Pi_r(.)$ is the truncated SVD (or hard thresholding) operator with rank r, and $\text{fold}_{\langle k,d\rangle}(.)$ is the reverse operation of TR unfolding.

3) *Update $\mathcal{X}$:* Tensor $\mathcal{X}$ can be obtained as

$$\mathcal{X} = \arg\min_{\mathcal{X}}\left(\frac{\lambda}{\mu}\Big\|\sqrt{\mathcal{W}}\circ\sqrt{\mathcal{Q}}\circ(\mathcal{M}-\mathcal{X})\Big\|_F^2 + \sum_{k=1}^{N}\Big\|\mathcal{X}-(\mathcal{Z}^{(k)}+\frac{1}{\mu}\mathcal{G}^{(k)})\Big\|_F^2\right). \tag{24}$$

By taking the derivative of $\mathcal{X}$ and setting it to be the zero tensor, we obtain the optimal solution

$$\mathcal{X} = \mathcal{L} + \frac{\lambda\mathcal{W}\circ\mathcal{Q}}{\lambda\mathcal{W}\circ\mathcal{Q}+\mu N}\circ(\mathcal{M}-\mathcal{L}) \tag{25}$$

where the division of tensors is computed element-wise and $\mathcal{L}=(1/N)\sum_{k=1}^{N}(\mathcal{Z}^{(k)}+(1/\mu)\mathcal{G}^{(k)})$.

4) *Update $\mathcal{G}^{(k)}$:* For each k, $\mathcal{G}^{(k)}$ can be updated as

$$\mathcal{G}^{(k)} = \mathcal{G}^{(k)} + \mu(\mathcal{Z}^{(k)}-\mathcal{X}). \tag{26}$$

We name this algorithm HQ-based weighted TR recovery (HQWTRR), and its pseudocode is summarized in Algorithm 2. For global coarse completion, $\{\mathcal{G}^{(k),0}\}_{k=1}^{N}$ and $\mathcal{X}^1$ are initialized as zero tensors, and $\mathcal{W}$ is set according to (3), that is, entries $(i_1,\ldots,i_N)\in\Omega$ are set to 1 and 0 otherwise. While in the global-completion-guided local patch refinement, for each patch $\mathcal{S}$, all entries of $\{\mathcal{G}^{(k),0}\}_{k=1}^{N}$ and $\mathcal{X}^1$ are initialized as the average value of entries of the corresponding reference patch $\mathcal{S}_r$, and $\mathcal{M}$ is a fully observed tensor with $\mathcal{W}$ obtained from (8).

Algorithm 2 HQWTRR for Weighted Robust Tensor Recovery

```
Input: Partially observed $\mathcal{M}$ with observation set $\Omega$, $\mathcal{W}$, $d$, $\mu$, $\alpha$,
    $\{r_k\}_{k=1}^{N}$, $\lambda$, $\eta_c$ and $\epsilon$
initial tensors $\{\mathcal{G}^{(k),0}\}_{k=1}^{N}$, $\mathcal{X}^1$, set $\mathcal{W}_{i_1\ldots i_N}=0$ for
    $(i_1,\ldots,i_N)\notin\Omega$, $t=1$
repeat
    estimate $c^{t+1}$ using (17).
    compute $\mathcal{Q}^{t+1}$ using (15).
    compute $\mathcal{Z}^{(k),t+1}$ for $k=1,\ldots,N$ using (23).
    compute $\mathcal{X}^{t+1}$ using (25).
    compute $\mathcal{G}^{(k),t+1}$ for $k=1,\ldots,N$ using (26).
    update $\mu^{t+1}=\alpha\mu^{t}$
    $t=t+1$
until $|\|\mathcal{X}^{t-1}-\mathcal{X}^{t-2}\|_F/\|\mathcal{X}^{t-2}\|_F - \|\mathcal{X}^{t}-\mathcal{X}^{t-1}\|_F/$
    $\|\mathcal{X}^{t-1}\|_F| < \epsilon$
Output: $\mathcal{M}=\mathcal{X}^{t}$.
```

D. Relation to Prior Tensor Ring Completion Algorithms

To better understand the relationship between the proposed algorithm and existing ℓ_2-norm-based TR completion algorithms, we first rewrite (25) element-wise as

$$\mathcal{X}_{i_1\ldots i_N} = \Theta_{i_1\ldots i_N}\mathcal{M}_{i_1\ldots i_N} + (1-\Theta_{i_1\ldots i_N})\mathcal{L}_{i_1\ldots i_N} \tag{27}$$

with $\Theta=([\lambda\mathcal{W}\circ\mathcal{Q}]/[\lambda\mathcal{W}\circ\mathcal{Q}+\mu N])$. When the regularization parameter λ is set to a sufficiently large value compared with μN, (27) reduces to

$$\mathcal{X} = \begin{cases}\mathcal{M}_{i_1\ldots i_N}, & (i_1,\ldots,i_N)\in\Omega\\ \mathcal{L}_{i_1\ldots i_N}, & (i_1,\ldots,i_N)\notin\Omega.\end{cases} \tag{28}$$

Again, by replacing hard thresholding using $\{r_{kd}\}_{k,d=1}^{N}$ in (22) with a soft thresholding method, Algorithm 2 reduces to the traditional TR nuclear norm minimization (TRNNM) method [10] solving

$$\min_{\mathcal{X}}\sum_{k=1}^{N}\beta_k\left\|\mathbf{X}_{\langle k,d\rangle}\right\|_* + \lambda\|\mathcal{W}\circ(\mathcal{X}-\mathcal{M})\|_F^2\,. \tag{29}$$

Thus, TRNNM can be seen as a special case of HQWTRR.

When the regularization parameter λ is properly chosen, the elements of $\mathcal{Q}$ will assign different weights to different values of the error residuals. It can be observed from Fig. 2, that a large error residual $\mathcal{E}_{i_1\ldots i_N}$ caused by an outlier may result in a small $\mathcal{Q}_{i_1\ldots i_N}$ (consequently a small $\Theta_{i_1\ldots i_N}$). In this case, the values of the entries with large error residuals will be dominated by the predicted value $\mathcal{Q}$ rather than $\mathcal{M}$. If the error residual is large enough, θ will be zero so the corresponding entry will be set to the corresponding entry in $\mathcal{Q}$, which amounts to treating it as a missing entry. In general, by assigning different weights to observed entries, the proposed algorithm can automatically identify the outliers.

E. Convergence Analysis

The following theorem characterizes the convergence of the proposed algorithm. For simplicity, we define $\mathcal{Z}_a = \{\mathcal{Z}^{(k)}\}_{k=1}^N$ and $\mathcal{G}_a = \{\mathcal{G}^{(k)}\}_{k=1}^N$.

Theorem 2 (HQWTRR Convergence): Let $\{(\mathcal{X}^t, \mathcal{Q}^t, \mathcal{Z}_a^t, \mathcal{G}_a^t)\}$ be a sequence generated by Algorithm 2 using the M-estimators defined in Fig. 2. If $\|\mathcal{X}^{t+1} - \mathcal{X}^t\|_F^2 < \infty$ for any $t \geq 1$, and $\{\mathcal{G}^{(k),t}\}$ converges to some constant tensor $\mathcal{C}$ for all $k = 1, \ldots, N$, then, $\{\mathcal{X}^t\}$ will converge for an M-estimator parameter c decreasing to 0.

The proof is deferred to the Appendix. In the theorem, a sequence $\{c^t\}$ with $\lim_{t\to\infty} c^t = 0$ is sufficient to ensure convergence of HQWTRR. In practice, adaptive parameter selection using (17) yields a sequence $\{c^t\}$ that approaches a small $c_{\min}$, albeit not monotonically decreasing. Still, it yields the desirable performance as shown in the experimental results. Since HQWTRR is a nonconvex optimization problem due to the use of a truncated SVD, the convergence analysis of ADMM is very challenging in general without additional assumptions [56]. Hence, similar to [57], [58], the assumption of the convergence of $\{\mathcal{G}^{(k),t}\}$ is used in Theorem 2. In practice, HQWTRR using Algorithm 2 works very well without this assumption, which is verified in Section VI.

F. Complexity Analysis

We first analyze the complexity of the global coarse completion step. Given an N-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, for simplicity we assume the TR rank $r_1 = \cdots = r_N = r$ and the tensor size $I_1 = \cdots = I_N = I$. Then, the time complexity of updating $\{\mathcal{Z}^{(k)}\}_{k=1}^N$ in (23) using truncated SVD is $\mathcal{O}(I^N r^2 N)$. The update of $\mathcal{Q}$ incurs a complexity of $\mathcal{O}(I^N)$, and the complexity of updating $\{\mathcal{G}^k\}_{k=1}^N$ or $\mathcal{X}$ is $\mathcal{O}(I^N N)$. Therefore the total time complexity of the coarse stage is $\mathcal{O}(I^N(r^2+2)N)$.

For the local patch refinement step, assume the total number of patch tensors is T_p. The dimension of the tensor is N_p, the length of each dimension is I_p, and the elements of the TR rank are all set to r_p. Then, the total time complexity of the fine stage is $\mathcal{O}(I_p^{N_p}(r_p^2+2)T_p N_p)$. We should remark that tensor completion is independent for each patch tensor, hence parallel computation can be applied to further improve the computational efficiency.

VI. Experimental Results

We conduct experiments using both image and video data to verify the performance of the proposed algorithm. We compare with existing tensor completion algorithms using different tensor rank models, including ℓ_1-regularized sum of nuclear norm (ℓ_1-SNN)[1] [20], soft thresholding using Welsch loss (W-ST) [50], tensor nuclear norm (TNN) [6], ℓ_1-regularized TNN (ℓ_1-TNN) [23], transformed nuclear norm-based total variation (TNTV)[2] [59], ℓ_p-regularized TT completion (ℓ_p-TTC)[3] [24], TR nuclear norm (TRNN) [10], ℓ_1 regularized TRNN (ℓ_1-TRNN)[4] [22], and $\ell_{p,\epsilon}$-regularized TR completion ($\ell_{p,\epsilon}$-TRC) [25]. All these algorithms are robust tensor completion algorithms except TRNN and TNN. For the proposed algorithm, we use C2FRTRC to designate the two-stage algorithm described in Algorithm 1, which uses HQWTRR for both global completion and local refinement. For comparison, we also include the results of global completion alone (without the local refinement) obtained at the coarse stage using HQWTRR, and local patch-only completion results without the global completion prior (i.e., setting $v_{\max} = 0$ and $\sigma_{\min} = +\infty$). To distinguish these two single-stage methods from the two-stage C2FRTRC, in the experiments we name the global-only completion procedure HQ TR Completion (HQTRC), and the local patch-only completion algorithm local patch-based robust TR completion (LPRTRC). For HQTRC, LPRTRC, and C2FRTRC, we use the Cauchy loss function as the default.

Two visual data quality metrics are used, including peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). For each experiment, the average PSNR/SSIM values are obtained over 20 Monte Carlo runs with different missing entries and noise realizations. For the proposed C2FRTRC framework in Algorithm 1, the patch size m is set to 36 and 20 for image and video data, respectively. The pixel overlap o is set to $o = \lceil m/5 \rceil$, and the jitter parameter l is set to 2. For HQTRC in Algorithm 2, we set $\mu = 10^{-4}$, $\lambda = 2\mu N$, $\alpha = 1.1$, $d = \lceil N/2 \rceil$, and $\epsilon = 10^{-3}$. For adaptive selection of σ, v, and c in (9) and (17), the parameters are set to $\eta_\sigma = 0.02$, $\eta_v = \eta_c = 4$, $\sigma_{\min} = 0.3$, $v_{\max} = 0.2$, $c_{\min} = 0.15$. For rank selection, we set al. the elements of the rank to be the same, that is, $r_1 = \cdots = r_N = r$. Then, inspired by [50], the parameter r is determined as $(0.04pI_1I_2)^{1/4}$ and $(0.25\ \mathrm{pm}^2)^{1/4} f^{1/6}$ for global tensor completion and local patch refinement, respectively, where p is the observation rate and f is the number of frames. For ℓ_p-TTC and $\ell_{p,\epsilon}$-TRC, p is set to 1. For the other algorithms, the parameters are adjusted so as to achieve the best performance. Further, the parameters are fixed during each simulation. All algorithms are implemented using MATLAB r2021a on a standard 16-GB memory PC with a 2.6-GHz CPU. The algorithms are run without any acceleration from (e.g., parallel computation).

A. Color Image Inpainting

In this section, we verify the robust completion performance on an image inpainting task using the proposed framework, in comparison to other existing TR completion algorithms. Image inpainting takes advantage of the fact that most natural images can be well approximated with their low-rank components, such that filling missing parts of an incomplete image can be regarded as a tensor completion problem.

Test images of size 320 × 480 × 3 are selected from the Berkeley segmentation dataset [51]. For each image, the pixel value is first normalized to $[0, 1]$. Then, pI_1I_2n pixels are selected uniformly at random and set as observed pixels, and the observed pixels are further perturbed with

[1] https://tonyzqin.wordpress.com/research

[2] https://github.com/xjzhang008/TNTV

[3] https://github.com/LI-X-P/CodeofRobustTensorCompletion

[4] https://github.com/HuyanHuang/Robust-Low-rank-Tensor-Ring-Completion

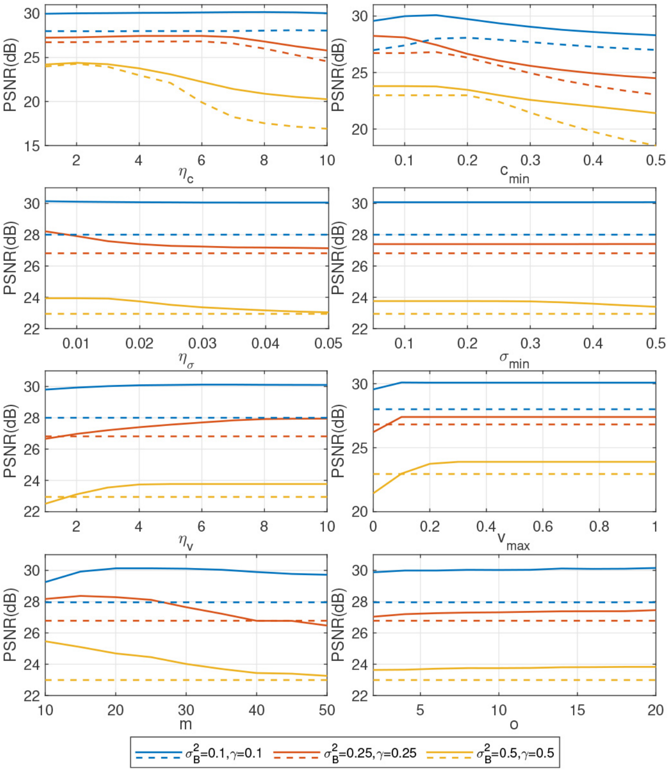


Fig. 4. Curves of average PSNR versus different parameters. Solid lines: C2FRTRC, dotted lines: HQTRC.

i.i.d. additive noise generated from a given distribution. The image inpainting task is then formulated as a $320 \times 480 \times 3$ robust tensor completion problem with an observation rate p. For TRNN, which favors high-order tensors for the better performance [10], we reshape the tensor to a 9-order tensor of size $4 \times 4 \times 4 \times 5 \times 4 \times 4 \times 5 \times 6 \times 3$. For the proposed HQTRC, we reshape the tensor to the same size used for TRNN, while for local patch tensor refinement with HQWTRR, the tensor size is not changed.

1) Ablation Experiment and Parameter Sensitivity Analysis: We first carry out an ablation experiment and parameter sensitivity analysis on the proposed coarse-to-fine framework and robust TR algorithm. The experiment is carried out on the flower image (see Fig. 6). The observed pixels are perturbed with additive noise generated from the standard two-component Gaussian mixture model (GMM) with probability density function $(1-\gamma)N(0,\sigma_A^2)+\gamma N(0,\sigma_B^2)$. $N(0,\sigma_A^2)$ represents the general Gaussian noise disturbance with variance σ_A^2, and $N(0,\sigma_B^2)$ with a large variance σ_B^2 captures the outliers. The variable γ controls the occurrence probability of outliers. Unless specified otherwise, the observation rate is set to $p = 0.5$, and the GMM noise parameters $\sigma_A^2 = 0.001, \sigma_B^2 = 0.25, \gamma = 0.5$.

First, we investigate the parameter sensitivity of the adaptive strategy in (9) and (17), along with the patch size m and pixel overlap o. Fig. 4 depicts the average PSNR of the recovered image versus η_c, η_v, η_σ, $v_{\max}$, $\sigma_{\min}$, and $c_{\min}$ under different outlier noise variance σ_B^2 using HQTRC and C2FRTRC. As shown, C2FRTRC outperforms HQTRC over a wide range of parameters. Specifically, for C2FRTRC, a small value of η_c, η_σ, $c_{\min}$, $\sigma_{\min}$, and m can result in a relatively higher PSNR, while a larger value of η_v, $v_{\max}$, and o can yield a higher PSNR. One can also observe that when η_c and $c_{\min}$ increase to relatively large values, the algorithm cannot properly alleviate the effect of the outliers and the performance degrades.

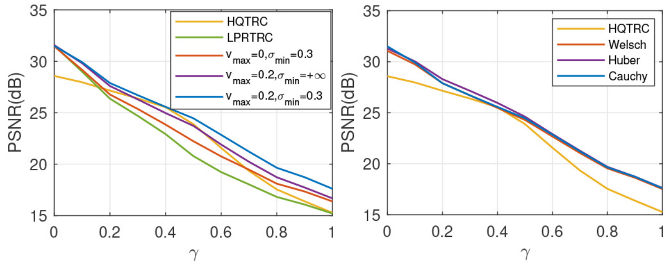


Fig. 5. Left: average PSNR versus outlier occurrence probability γ under different parameters $v_{\max}$ and $\sigma_{\min}$. Right: average PSNR versus p for different M-estimators.

Second, we evaluate the performance of the weighted strategy on local tensor refinement using the global completion result. Apart from LPRTRC which corresponds to the setting $v_{\max} = 0$ and $\sigma_{\min} = +\infty$, we also test the results of completion with nonsoft weight ($v_{\max} = 0.2$, $\sigma_{\min} = +\infty$) and without missing entry filling from the global completion result ($v_{\max} = 0$, $\sigma_{\min} = 0.3$). The average PSNR for different outlier occurrence probability γ is shown in Fig. 5 (left). One can observe that compared with local-only LPRTRC, the completion performance is greatly improved by incorporating the global completion information. The best performance of C2FRTRC verifies that both filling the missing entries with the result of global completion and assigning weights using the soft weighting strategy can improve the completion performance.

Third, we test the performance using different M-estimators. The parameter settings for the Welsch loss function are the same as the Cauchy loss function. For the Huber estimator, the parameters η_c and $c_{\min}$ are set to 2 and 0.05, respectively. The curves of average PSNR with different M-estimators are shown in Fig. 5 (right). As shown, the M-estimators yield similar performance, showing the flexibility of the proposed robust method with different selections of M-estimators.

To better illustrate the performance improvement with the proposed framework, we show a visual example of the recovered image using different weight parameters and M-estimators in Fig. 6. As can be seen, compared with global tensor completion using HQTRC, the proposed coarse-to-fine framework can improve the texture details, especially in the heavy outlier scenario (2nd and 3rd rows). Further, the global information can also help local patch tensor refinement for accurate estimation of the missing pixels.

2) Performance Comparison With Other Algorithms: In this part, we compare with existing tensor completion algorithms for different noise environments. We use four images (shown in Fig. 7) and add different noise to each image. Specifically, for image flower, all observed pixels are perturbed with Gaussian noise with zero mean and variance σ_G^2. For image cruise, GMM noise with outlier occurrence probability

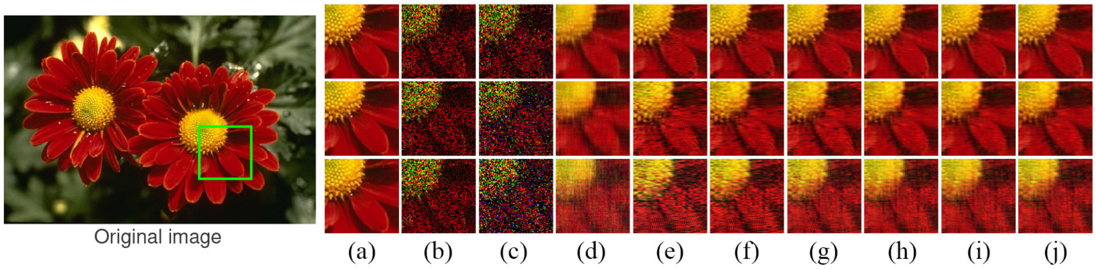


Fig. 6. Example of the recovered images (partially enlarged) using different parameters. From top to bottom row: $c = 0.2, 0.5, 0.8$. (a) Original partially enlarged image. (b) Noiseless images with missing entries. (c) Noisy image with missing entries (final observed image). (d) and (e) Recovered images from HQTRC and LPRTRC. (f) and (g) Recovered images from C2FRTRC for different pairs $\nu_{\max}$ and $\sigma_{\min}$: $\{\nu_{\max} = 0, \sigma_{\min} = 0.3\}$ and $\{\nu_{\max} = 0.2, \sigma_{\min} = +\infty\}$. (h)–(j) Recovered images from C2FRTRC using different M-estimators: Huber, Welsch, and Cauchy.

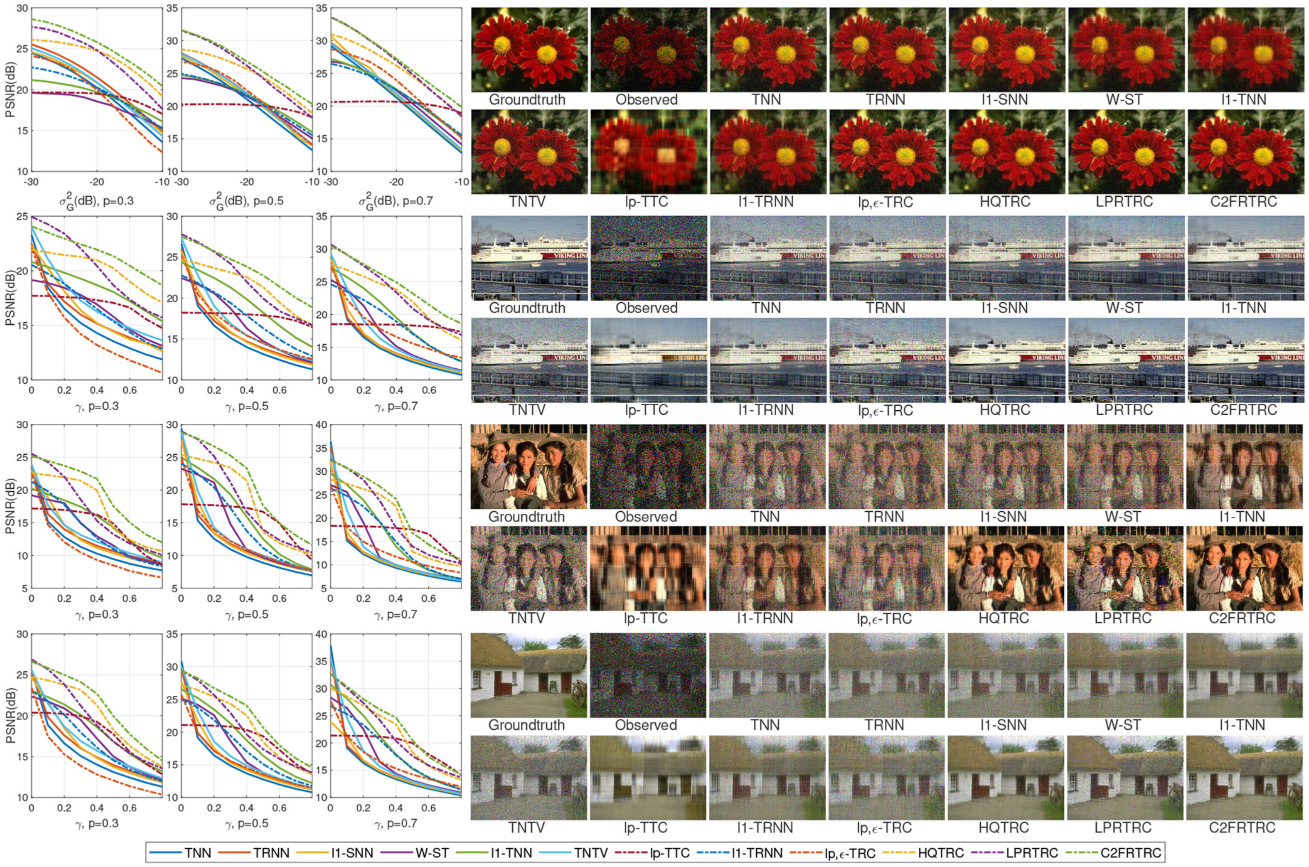


Fig. 7. From top to bottom: "flower", "cruise", "girls" and "house". For each image, (left) Average PSNR versus noise parameter γ/σ_G^2 for different observation rates $p = 0.3, 0.5, 0.7$. (Right) example of the recovered images ($p = 0.5, \gamma = 0.4/\sigma_G^2 = -20$ dB). Best viewed in ×2 sized color pdf file.

γ is added to the observed pixels. For image girls, $\gamma \times 100\%$ of the observed pixels is perturbed with salt and pepper noise. While for image house, $\gamma \times 100\%$ of the observed pixels is replaced with random values in the range [0, 1].

We investigate the performance on the four images for different settings of the noise parameter γ and observation rate p. The average PSNR for different algorithms is shown in Fig. 7 (left), and an example of the recovered images is shown in Fig. 7 (right). It can be observed that the proposed C2FRTRC obtains the overall best performance for different noisy environments. Specifically, LPRTRC can achieve the similar performance to C2FRTRC in Gaussian noise and non-Gaussian noise with small number of outliers ($\gamma \leq 0.2$). While in heavy noise, C2FRTRC is guided by the global prior which can further enhance the performance, resulting in the better performance than LPRTRC. One should also notice that when $p = 0.7$ and $\gamma = 0$ (i.e., noise-free case), TNTV, TRNN, and TNN may outperform the proposed method. However, these algorithms suffer from the severe performance degradation with a small number of outliers (i.e., $\gamma = 0.1$).

B. Video Completion

In this part, we compare the performance of the proposed method with existing robust tensor completion algorithms in a video completion task. The completion performance is evaluated using four color video fragments from the YUV dataset.[5]

[5] http://trace.eas.asu.edu/yuv/

TABLE I
COMPLETION PERFORMANCE COMPARISON FOR DIFFERENT ALGORITHMS ON FOUR VIDEO SEQUENCES WITH DIFFERENT MISSING PATTERNS

Video	Noise setting	Metric	TNN	TRNN	ℓ_1-SNN	W-ST	ℓ_1-TNN	TNTV	ℓ_p–TTC	ℓ_1-TRNN	$\ell_{p,\epsilon}$-TRC	HQTRC	LPRTRC	C2FRTRC
Tempete	γ 0.3	PSNR	11.45	11.56	11.53	12.91	22.03	17.38	17.60	18.50	20.57	22.50	$\underline{25.93}$	**26.49**
		SSIM	0.3122	0.3177	0.3148	0.3371	0.7218	0.5338	0.5193	0.5839	0.6417	0.8088	$\underline{0.9115}$	**0.9194**
	0.5	PSNR	9.29	9.35	9.44	10.60	14.60	12.26	17.64	12.42	20.41	22.59	$\underline{24.62}$	**25.33**
		SSIM	0.2215	0.2265	0.2252	0.2403	0.4050	0.3193	0.5179	0.3227	0.6267	0.7797	$\underline{0.8559}$	**0.8818**
	0.7	PSNR	7.78	7.96	7.93	8.98	10.94	9.66	17.76	9.55	20.13	$\underline{20.88}$	19.02	**22.19**
		SSIM	0.1632	0.1661	0.1634	0.1627	0.2450	0.2030	0.5144	0.2013	0.6163	$\underline{0.6440}$	0.5636	**0.7082**
Stefan	γ 0.3	PSNR	11.63	12.14	11.71	8.95	18.87	13.51	17.18	12.72	17.89	19.82	$\underline{22.32}$	**22.48**
		SSIM	0.2079	0.2245	0.2223	0.1449	0.6037	0.2949	0.5378	0.2472	0.5061	0.6819	**0.8117**	$\underline{0.8025}$
	0.5	PSNR	9.51	10.11	9.55	7.59	11.50	10.28	$\underline{17.13}$	10.03	16.73	16.11	15.55	**19.95**
		SSIM	0.1112	0.1254	0.1091	0.0630	0.1818	0.1391	$\underline{0.5375}$	0.1224	0.4090	0.3309	0.3273	**0.6672**
	0.7	PSNR	8.01	8.72	8.24	6.71	8.94	8.65	$\underline{16.89}$	8.56	15.10	14.57	13.70	**17.34**
		SSIM	0.0570	0.0615	0.0608	0.0509	0.0718	0.0667	**0.5251**	0.0610	0.1917	0.1543	0.1620	$\underline{0.4598}$
Foreman	γ 0.3	PSNR	11.40	11.55	11.51	11.11	21.96	18.79	18.21	19.13	22.45	25.58	$\underline{29.52}$	**30.71**
		SSIM	0.1955	0.2137	0.2128	0.1985	0.5887	0.4758	0.6704	0.5022	0.7208	0.8781	$\underline{0.9270}$	**0.9275**
	0.5	PSNR	9.24	9.39	8.94	9.35	14.61	12.54	18.26	12.46	22.08	25.37	$\underline{27.67}$	**29.47**
		SSIM	0.1354	0.1495	0.1457	0.1378	0.2815	0.2267	0.6674	0.2274	0.7021	0.8433	$\underline{0.8750}$	**0.9117**
	0.7	PSNR	7.85	7.93	7.87	7.95	10.99	9.69	18.37	9.50	21.85	$\underline{22.06}$	19.51	**23.99**
		SSIM	0.1007	0.1136	0.1138	0.1023	0.1597	0.1432	0.6593	0.1397	$\underline{0.6927}$	0.6496	0.5103	**0.7356**
Bus	σ_G^2 (dB) -20	PSNR	20.09	20.17	19.95	19.53	21.40	21.07	16.54	20.51	19.34	21.66	**25.20**	$\underline{24.06}$
		SSIM	0.5108	0.5333	0.5253	0.5049	0.5446	0.5597	0.3154	0.5400	0.4045	0.6226	**0.7411**	$\underline{0.7221}$
	-15	PSNR	15.42	15.89	15.66	13.78	17.36	17.05	16.63	16.34	19.22	21.09	**22.59**	$\underline{22.03}$
		SSIM	0.3109	0.3343	0.3265	0.2693	0.3514	0.3674	0.2928	0.3425	0.3862	0.5511	**0.5884**	$\underline{0.5878}$
	-10	PSNR	10.46	10.70	10.67	10.65	12.29	11.67	16.61	11.28	19.05	$\underline{20.01}$	18.74	**20.08**
		SSIM	0.1662	0.1734	0.1710	0.1560	0.1970	0.1938	0.2720	0.1838	0.3649	**0.4384**	0.3919	$\underline{0.4350}$

Some frames of the original videos are shown in Fig. 8. For each video, a sequence of 30 frames is selected, and each frame is resized to 144 × 180 to obtain a tensor of size 144 × 180 × 3 × 30. Similar to the previous section, a tensor with noisy and missing (partially observed) entries is generated by selecting a fraction of pixels as observed pixels and then adding i.i.d. noise from a given distribution to the observed pixels. For TRNN and HQTRC, the observed tensor is reshaped to an 11-order tensor of size $3 \times 3 \times 4 \times 4 \times 3 \times 3 \times 4 \times 5 \times 3 \times 5 \times 6$. For ℓ_1-SNN, W-ST ℓ_1-TNN and TNTV, we reshape the tensor to a 3-order tensor of size 144 × 180 × 90.

We apply different types of missing patterns and noise distributions to each video fragment. In particular, for video "tempete", a fixed sentence is masked on all frames so that the video contains a "watermark", and $\gamma \times 100\%$ of the rows in each frame is perturbed with outliers generated from a Gaussian distribution with zero mean and variance 0.25. For video "Stefan", 60% of the rows is randomly and uniformly selected as the observed rows, and the observed data is perturbed with salt and pepper noise with probability $\gamma \times 100\%$. For video "foreman", a watermark moving from the top-left to the bottom right of the video is added as the missing pattern, then GMM noise with $\sigma_A^2 = 0.001$, $\sigma_B^2 = 0.25$ and outlier occurrence probability γ is added to the observed pixels. Finally, for video "bus", we use a time-variant missing pattern to simulate the effect of the raindrop, and a Gaussian noise with zero mean and variance σ_G^2 is added to the observed data. Representatives of the observed noisy frames are shown in Fig. 8.

Table I shows the average PSNR and SSIM for different algorithms on four video fragments in different noise environments. The SSIM of a video is computed as the average SSIM across frames. As shown, C2FRTRC achieves the overall best performance, and LPRTRC yields the second-best performance. Specifically, C2FRTRC achieves the highest PSNR/SSIM in most cases. In Gaussian noise environments with $\sigma_G^2 \leq -15$ dB, LPRTRC outperforms C2FRTRC. Fig. 8 shows an example of the recovered frames from the four fragments in heavy noise environments. As can be seen, only the proposed C2FRTRC successfully recovers the frames of all videos. Further, similar to image completion, C2FRTRC yields the best visual results having the most clean and detailed texture.

Fig. 9 reports the average running time of all algorithms on the four videos in Table I. It can be seen that the coarse completion HQTRC incurs a time cost similar to ℓ_1-based algorithms ℓ_1-SNN, ℓ_1-TNN, and ℓ-TRNN. For C2FRTRC, the time cost is higher due to the extra refinement step for local patch tensors. As refinement for each patch tensor is independent, the time cost of C2FRTRC could be reduced using parallel computation. Although the local fine stage incurs the additional time cost, the performance improvement is significant.

VII. CONCLUSION

We proposed a novel two-stage coarse-to-fine tensor completion framework for the robust visual data completion. A global coarse completion stage is first performed, whereby most of the outliers are identified. Then, guided by the result of global completion, a local patch refinement process is applied by performing robust tensor recovery incorporating both local and global information. Further, a new M-estimator-based TR

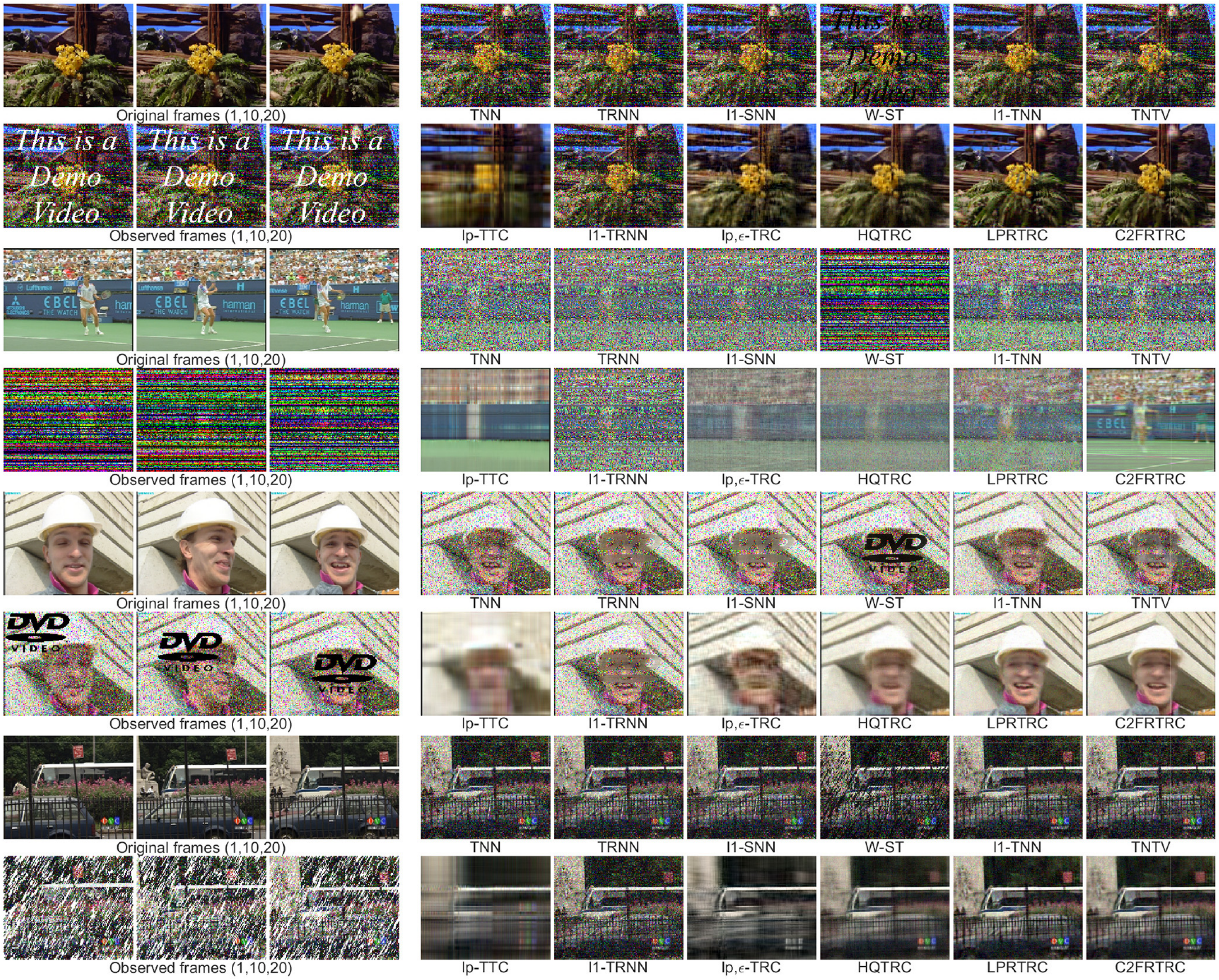


Fig. 8. From top to bottom: video tempete, stefan, foreman, and bus. For each video: (left) the original frames (frame number: 1, 10, 20) and corresponding observed noisy frames with missing pixels ($\gamma = 0.5/\sigma_G^2 = -15$ dB) and (right) recovered 20th frame using different algorithms. Best viewed in ×2 sized color pdf file.

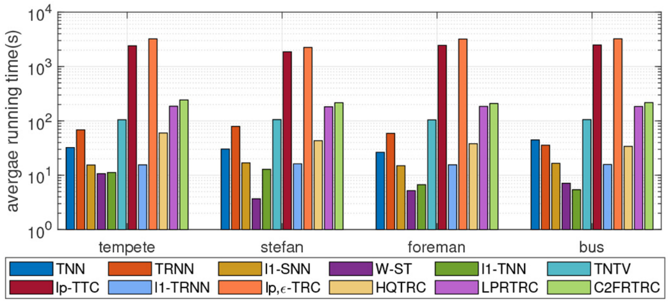


Fig. 9. Average running times of four videos corresponding to Table I.

recovery algorithm using HQ approach is proposed, which can accurately complete and recover the tensor in the presence of a large number of outliers. Numerical experiments on image and video completion in various noise environments demonstrate the advantage of incorporating global coarse completion with local patch refinement. The results also demonstrate that the proposed method can outperform existing state-of-the-art robust tensor completion algorithms, especially in heavy noise settings.

Appendix
Proof of Theorem 2

From (25) and (26), we have

$$\begin{aligned}\mathcal{L}^{t+1} &= \frac{1}{N}\sum_{k=1}^{N}\left(\mathcal{Z}^{(k),t+1} + \frac{1}{\mu}\mathcal{G}^{(k),t}\right)\\ &= \frac{1}{N}\sum_{k=1}^{N}\left(\mathcal{Z}^{(k),t+1} + \frac{1}{\mu}\left(\mathcal{G}^{(k),t+1} - \mu\left(\mathcal{Z}^{(k),t+1} - \mathcal{X}^{t+1}\right)\right)\right)\\ &= \mathcal{X}^{t+1} + \frac{1}{\mu N}\sum_{k=1}^{N}\mathcal{G}^{(k),t+1}. \end{aligned} \tag{30}$$

Substituting (30) into (25) we get

$$\mathcal{G}_s^{t+1} = \lambda\mathcal{V}^{t+1} \circ \left(\mathcal{X}^{t+1} - \mathcal{M}\right) \tag{31}$$

where $\mathcal{G}_s^{t+1} = \sum_{k=1}^{N} \mathcal{G}^{(k),t+1}$ and $\mathcal{V}^{t+1} = \mathcal{W} \circ \mathcal{Q}^{t+1}$. Since $\mathcal{W}_{i_1\dots i_N} = 0$ for $(i_1, \dots, i_N) \notin \Omega$, we get that $(\mathcal{G}_s^{t+1})_{i_1\dots i_N} = 0$ for $(i_1, \dots, i_N) \notin \Omega$. Next, we will show that $\lim_{c\to 0}(\mathcal{G}_s^{t+1})_{i_1,\dots,i_N} = 0$ for $(i_1, \dots, i_N) \in \Omega$.

Here, we use the Welsch function as an example. By applying the Welsch function to the second term on the RHS of (25), we have that for all $(i_1 \dots i_N) \in \Omega$

$$\begin{aligned}\mathcal{Q}_{i_1\dots i_N}^{t+1}&(\mathcal{M}_{i_1\dots i_N} - \mathcal{L}_{i_1\dots i_N}^{t+1})\\ &= \exp\left(-\frac{(\mathcal{M}_{i_1\dots i_N} - \mathcal{X}_{i_1\dots i_N}^{t})^2}{2c^2}\right)\\ &\quad\times\left(\mathcal{M}_{i_1\dots i_N} - \mathcal{X}_{i_1\dots i_N}^{t+1} - \frac{1}{\mu N}(\mathcal{G}_s^{t+1})_{i_1\dots i_N}\right). \qquad (32)\end{aligned}$$

Based on the assumption that $\|\mathcal{X}^{t+1} - \mathcal{X}^{t}\|_F^2 < \infty$, we get that $|\mathcal{X}_{i_1\dots i_N}^{t+1} - \mathcal{X}_{i_1\dots i_N}^{t}| \le P$, where P is some finite value. Thus, we have

$$a \le \mathcal{Q}_{i_1\dots i_N}^{t+1}(\mathcal{M}_{i_1\dots i_N} - \mathcal{L}_{i_1\dots i_N}^{t+1}) \le b \qquad (33)$$

where

$$a = \exp\left(-\frac{\left(\mathcal{E}_{i_1\dots i_N}^{t}\right)^2}{2c^2}\right)\left(\mathcal{E}_{i_1\dots i_N}^{t} - \frac{1}{\mu N}\left(\mathcal{G}_s^{t+1}\right)_{i_1\dots i_N} - P\right)$$

$$b = \exp\left(-\frac{\left(\mathcal{E}_{i_1\dots i_N}^{t}\right)^2}{2c^2}\right)\left(\mathcal{E}_{i_1\dots i_N}^{t} - \frac{1}{\mu N}\left(\mathcal{G}_s^{t+1}\right)_{i_1\dots i_N} + P\right)$$

where $\mathcal{E}_{i_1\dots i_N}^{t} = \mathcal{M}_{i_1\dots i_N} - \mathcal{X}_{i_1\dots i_N}^{t}$. For the Welsch function $f(x) = c^2(1 - \exp(-x^2/(2c^2)))$, $x \in \mathbb{R}$, $f'(x) = x\,\exp(-x^2/[2c^2]) \in [-ce^{-0.5}, ce^{-0.5}]$, hence a and b are bounded for any values of $\mathcal{X}_{i_1\dots i_N}^{t}$. It can be also observed that both a and b are 0 when $c \to 0$ and $\mathcal{E}_{i_1\dots i_N}^{t} \neq 0$. Therefore, for $\mathcal{E}_{i_1\dots i_N}^{t} \neq 0$, from (33) we have that $\lim_{c\to 0} \mathcal{Q}_{i_1\dots i_N}^{t+1}(\mathcal{M}_{i_1\dots i_N} - \mathcal{L}_{i_1\dots i_N}^{t+1}) = 0$. Then, using (31) one can further obtain that $\lim_{c\to 0}(\mathcal{G}_s^{t+1})_{i_1\dots i_N} = 0$.

The key point of the above analysis is the boundedness of $f'(x)$. Since $f'(x)$ is also bounded for the Cauchy and Huber functions, a similar result can be derived.

Combining the results above, we conclude that $\lim_{c\to 0}(\mathcal{G}_s^{t+1})_{i_1\dots i_N} = 0$ for $\{(i_1, \dots, i_N) : \mathcal{E}_{i_1\dots i_N}^{t} \neq 0\}$. Further, since $\{\mathcal{G}^{(k),t}\}$ converges to $\mathcal{C}$, we get that $\lim_{t\to\infty}\lim_{c\to 0}(\mathcal{G}^{(k),t})_{i_1,\dots,i_N} = 0$ for $\{(i_1, \dots, i_N) : \limsup_{t\to\infty}\mathcal{E}_{i_1\dots i_N}^{t} \neq 0\}$. Moreover, for the indices $(i_1, \dots, i_N)$ for which $\lim_{t\to\infty}\mathcal{E}_{i_1\dots i_N}^{t} = 0$, we have $\lim_{t\to\infty}\mathcal{X}_{i_1\dots i_N}^{t} = \mathcal{M}_{i_1\dots i_N}$, and from (31) one can obtain $\lim_{t\to\infty}(\mathcal{G}^{(k),t})_{i_1,\dots,i_N} = 0$. Therefore, we get that $\lim_{t\to\infty}\lim_{c\to 0}\mathcal{G}^{(k),t} = 0$.

We also conclude from (26) that $\{\mathcal{Z}^{(k),t} - \mathcal{X}^{t}\}$ converges to 0 for all $k = 1, \dots, N$. Therefore, from (23) we have

$$\begin{aligned}\lim_{t\to\infty}\lim_{c\to 0}\mathbf{X}_{\langle k,d\rangle}^{t+1} &= \lim_{t\to\infty}\lim_{c\to 0}\Pi_{r_{kd}}\left(\mathbf{X}_{\langle k,d\rangle}^{t} - \frac{1}{\mu}\mathbf{G}_{\langle k,d\rangle}^{(k),t}\right)\\ &= \lim_{t\to\infty}\lim_{c\to 0}\Pi_{r_{kd}}\left(\mathbf{X}_{\langle k,d\rangle}^{t}\right). \qquad (34)\end{aligned}$$

From the property of the truncated SVD, we have that, $\lim_{t\to\infty}\lim_{c\to 0}\mathcal{X}^{t+1} = \lim_{t\to\infty}\lim_{c\to 0}\mathcal{X}^{t}$. Therefore, we conclude that as c decreases to 0, $\{\mathcal{X}^{t}\}$ converges.

References

[1] Q. Song, H. Ge, J. Caverlee, and X. Hu, "Tensor completion algorithms in big data analytics," *ACM Trans. Knowl. Disc. Data*, vol. 13, no. 1, pp. 1–48, 2019.

[2] Y. Liu, F. Shang, L. Jiao, J. Cheng, and H. Cheng, "Trace norm regularized CANDECOMP/PARAFAC decomposition with missing data," *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2437–2448, Nov. 2015.

[3] L. Karlsson, D. Kressner, and A. Uschmajew, "Parallel algorithms for tensor completion in the CP format," *Parallel Comput.*, vol. 57, pp. 222–234, Sep. 2016.

[4] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.

[5] J. Fan, "Multi-mode deep matrix and tensor factorization," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–25.

[6] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, Mar. 2017.

[7] P. Zhou, C. Lu, Z. Lin, and C. Zhang, "Tensor factorization for low-rank tensor completion," *IEEE Trans. Image Process.*, vol. 27, pp. 1152–1163, 2018.

[8] W. Wang, V. Aggarwal, and S. Aeron, "Efficient low rank tensor ring completion," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5697–5705.

[9] H. Huang, Y. Liu, J. Liu, and C. Zhu, "Provable tensor ring completion," *Signal Process.*, vol. 171, Jun. 2020, Art. no. 107486.

[10] J. Yu, C. Li, Q. Zhao, and G. Zhao, "Tensor-ring nuclear norm minimization and application for visual: Data completion," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2019, pp. 3142–3146.

[11] J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do, "Efficient tensor completion for color image and video recovery: Low-rank tensor train," *IEEE Trans. Image Process.*, vol. 26, pp. 2466–2479, 2017.

[12] R. Dian, L. Fang, and S. Li, "Hyperspectral image super-resolution via non-local sparse tensor factorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5344–5353.

[13] S. Zhang, L. Wang, Y. Fu, X. Zhong, and H. Huang, "Computational hyperspectral imaging based on dimension-discriminative low-rank tensor recovery," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 10183–10192.

[14] Y. Xie *et al.*, "Robust kernelized multiview self-representation for subspace clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 868–881, Feb. 2021.

[15] Y. Tang, Y. Xie, C. Zhang, Z. Zhang, and W. Zhang, "One-step multiview subspace segmentation via joint skinny tensor learning and latent clustering," *IEEE Trans. Cybern.*, early access, Mar. 4, 2021, doi: 10.1109/TCYB.2021.3053057.

[16] A. Cichocki *et al.*, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, Mar. 2015.

[17] Y. Tang, Y. Xie, X. Yang, J. Niu, and W. Zhang, "Tensor multi-elastic kernel self-paced learning for time series clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 3, pp. 1223–1237, Mar. 2021.

[18] K. Xie *et al.*, "Recover corrupted data in sensor networks: A matrix completion solution," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1434–1448, May 2017.

[19] Q. Wang, L. Chen, Q. Wang, H. Zhu, and X. Wang, "Anomaly-aware network traffic estimation via outlier-robust tensor completion," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2677–2689, Dec. 2020.

[20] D. Goldfarb and Z. Qin, "Robust low-rank tensor recovery: Models and algorithms," *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 1, pp. 225–253, 2014.

[21] L. Yang, J. Fang, H. Li, and B. Zeng, "An iterative reweighted method for tucker decomposition of incomplete tensors," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4817–4829, Sep. 2016.

[22] H. Huang, Y. Liu, Z. Long, and C. Zhu, "Robust low-rank tensor ring completion," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 1117–1126, Jul. 2020.

[23] Q. Jiang and M. Ng, "Robust low-tubal-rank tensor completion via convex optimization." in *Proc. IJCAI*, 2019, pp. 2649–2655.

[24] Q. Liu, X. P. Li, H. Cao, and Y. Wu, "From simulated to visual data: A robust low-rank tensor completion approach using ℓ_p-regression for outlier resistance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 3462–3474, Jun. 2022.

[25] X. P. Li and H. C. So, "Robust low-rank tensor completion based on tensor ring rank via $\ell_{p,\epsilon}$-norm," *IEEE Trans. Signal Process.*, vol. 69, pp. 3685–3698, May 2021.

[26] T. Xie, S. Li, L. Fang, and L. Liu, "Tensor completion via nonlocal low-rank regularization," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2344–2354, Jun. 2019.

[27] L. Zhang, L. Song, B. Du, and Y. Zhang, "Nonlocal low-rank tensor completion for visual data," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 673–685, Feb. 2021.

[28] D. Meng, H. Tingzhu, Z. Xile, K. N. Michael, and M. Tianhui, "Tensor train rank minimization with nonlocal self-similarity for tensor completion," *Inverse Problems Imag.*, vol. 15, no. 3, pp. 475–498, 2021.

[29] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, "Tensor ring decomposition," 2016, *arXiv:1606.05535*.

[30] D. E. Tyler, "A distribution-free M-estimator of multivariate scatter," *Ann. Stat.*, vol. 50, no. 1, pp. 234–251, 1987.

[31] M. Nikolova and M. K. Ng, "Analysis of half-quadratic minimization methods for signal and image recovery," *SIAM J. Sci. Comput.*, vol. 27, no. 3, pp. 937–966, 2005.

[32] C.-F. Chen, C.-P. Wei, and Y.-C. F. Wang, "Low-rank matrix recovery with structural incoherence for robust face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2618–2625.

[33] Z. Kang, C. Peng, and Q. Cheng, "Top-n recommender system via matrix completion," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, 2016, pp. 179–185.

[34] Z. Yu, J. Yu, J. Fan, and D. Tao, "Multi-modal factorized bilinear pooling with co-attention learning for visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1821–1830.

[35] J. Fan, C. Yang, and M. Udell, "Robust non-linear matrix factorization for dictionary learning, denoising, and clustering," *IEEE Trans. Signal Process.*, vol. 69, pp. 1755–1770, Mar. 2021.

[36] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.

[37] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an 'explanatory' multimodal factor analysis," in *UCLA Working Papers Phonetics*. vol. 16. Los Angeles, CA, USA: Phonetics Lab, 1970, pp. 1–84.

[38] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[39] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, 2011.

[40] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, 2011.

[41] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 1–37, 2011.

[42] J. Fan and T. W. Chow, "Exactly robust kernel principal component analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 3, pp. 749–761, Mar. 2020.

[43] Z. Kang, H. Pan, S. C. Hoi, and Z. Xu, "Robust graph learning from noisy data," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 1833–1843, May 2020.

[44] L. Cambier and P.-A. Absil, "Robust low-rank matrix completion by Riemannian optimization," *SIAM J. Sci. Comput.*, vol. 38, no. 5, pp. S440–S460, 2016.

[45] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, pp. 2080–2095, 2007.

[46] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Trans. Image Process.*, vol. 22, pp. 119–133, 2013.

[47] W. Li, L. Zhao, Z. Lin, D. Xu, and D. Lu, "Non-local image Inpainting using low-rank matrix completion," *Comput. Graph. Forum*, vol. 34, no. 6, pp. 111–122, 2015.

[48] X.-L. Zhao, J.-H. Yang, T.-H. Ma, T.-X. Jiang, M. K. Ng, and T.-Z. Huang, "Tensor completion via complementary global, local, and nonlocal priors," *IEEE Trans. Image Process.*, vol. 31, pp. 984–999, 2022.

[49] J. E. Dennis and R. E. Welsch, "Techniques for nonlinear least squares and robust regression," *Commun. Stat. Simulat. Comput.*, vol. 7, no. 4, pp. 345–359, 1978.

[50] Y. Yang, Y. Feng, and J. A. K. Suykens, "Robust low-rank tensor recovery with regularized redescending M-estimator," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 9, pp. 1933–1946, Sep. 2016.

[51] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int. Conf. Comput. Vis.*, vol. 2, Jul. 2001, pp. 416–423.

[52] J. Yu, Y. Rui, and D. Tao, "Click prediction for Web image reranking using multimodal sparse coding," *IEEE Trans. Image Process.*, vol. 23, pp. 2019–2032, 2014.

[53] R. He, B. Hu, X. Yuan, and L. Wang, *Robust Recognition via Information Theoretic Learning*. Cham, Switzerland: Springer, 2014.

[54] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Trans. Image Process.*, vol. 6, pp. 298–311, 1997.

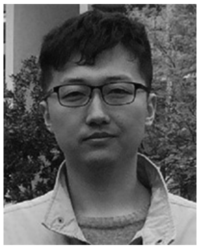
[55] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[56] S. Boyd, N. Parikh, and E. Chu, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Hanover, MA, USA: Now Publ., Inc., 2011.

[57] B. Jiang, S. Ma, and S. Zhang, "Alternating direction method of multipliers for real and complex polynomial optimization models," *Optimization*, vol. 63, no. 6, pp. 883–898, 2014.

[58] S. Magnússon, P. C. Weeraddana, M. G. Rabbat, and C. Fischione, "On the convergence of alternating direction lagrangian methods for nonconvex structured optimization problems," *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 3, pp. 296–309, Sep. 2016.

[59] D. Qiu, M. Bai, M. K. Ng, and X. Zhang, "Robust low-rank tensor completion via transformed tensor nuclear norm with total variation regularization," *Neurocomputing*, vol. 435, pp. 197–215, May 2021.



Yicong He received the B.S. degree in automation and the Ph.D. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2012 and 2019, respectively.

He is currently a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL, USA. His current research interests include robust signal processing and machine learning.

George K. Atia (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical and computer engineering from Alexandria University, Alexandria, Egypt, in 2000 and 2003, respectively, and the Ph.D. degree in electrical and computer engineering from Boston University, Boston, MA, USA, in 2009.

He joined the University of Central Florida, Orlando, FL, USA, in Fall 2012, where he is currently an Associate Professor with the Department of Electrical and Computer Engineering. He was a Visiting Faculty with the Air Force Research Laboratory, Wright-Patterson AFB, OH, USA, in 2019 and 2020. From 2009 to 2012, he was a Postdoctoral Research Associate with the Coordinated Science Laboratory, University of Illinois at Urbana–Champaign, Champaign, IL, USA. His research interests include machine learning and data analytics, explainable AI, statistical signal processing, detection and estimation theory, and information theory.

Dr. Atia is the recipient of many awards, including the UCF Reach for the Stars Award and the CECS Research Excellence Award in 2018, the Dean's Advisory Board Fellowship and the Inaugural UCF Luminary Award in 2017, the NSF CAREER Award in 2016, and the Charles Millican Faculty Fellowship Award (2015–2017). He is an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING.