Decentralized Dimensionality Reduction for Distributed Tensor Data across Sensor Networks

Junli Liang, Guoyang Yu, Badong Chen, and Minghua Zhao

Abstract—This paper develops a novel Decentralized Dimensionality Reduction (DDR) algorithm for distributed tensor data across sensor networks. The main contributions of this paper are: i) conventional centralized methods, which utilize entire data to simultaneously determine all vectors of the projection matrix along each tensor mode, are not suitable for the network environment. Here, we relax the simultaneous processing manner into the one-vector-by-one-vector (OVBOV) manner, i.e., determining the projection vectors (PVs) related to each tensor mode one by one; ii) we prove that in the OVBOV manner each PV can be determined without modifying any tensor data, which simplifies corresponding computations; iii) we cast the decentralized PV determination problem as a set of subproblems with consensus constraints so that it can be solved in the network environment only via local computations and information communications among neighboring nodes; iv) we introduce the null space and transform the PV determination problem with complex orthogonality constraints into an equivalent hidden convex one without any orthogonality constraint, which can be solved by the Lagrange multiplier method. Finally, experimental results are given to show that the proposed algorithm is an effective dimensionality reduction scheme for distributed tensor data across sensor networks.

Index Terms—Decentralized Dimensionality Reduction (DDR), Projection Vector (PV), Tensor data, Null space, Orthogonality constraint, Elementary Multilinear Projection (EMP), One-Vector-By-One-Vector (OVBOV).

I. INTRODUCTION

Tensors (multi-way arrays) are generalizations of vectors and matrices to an arbitrary number of indices [1], [2]. Images and videos are intrinsically such tensor objects [1]–[4]. Recently, automatic recognition of human actions, biological characteristics, and other objects [5]–[9] receives lots of attention in terms of multi-view images or image sequences due to its wide applications in machine vision, medical image analysis, content-based retrieval, remote surveillance, social security, and biometric authentication, etc [2], [3], [10]. Actually, it is a tensor analysis problem [2], [3], [11]. However, direct operations on tensors often suffer from the so-called “curse of dimensionality” due to the intrinsic high-dimensional characteristics of tensor spaces. Commonly, it is believed that the high-dimensional tensors reside (nearly) on a low-dimensional manifold. Hence, prior to applying recognition methods, the so-called dimensionality reduction (DR) techniques are typically performed to transform the tensor data into those with low-dimension that compactly represent these tensors while keeping their most underlying structure information.

Traditional DR methods unfold tensor data into a long vector and thus each vector can be viewed as a point in an ultra-high-dimensional vector space. However, such a vector-to-vector projection (VVP) manner increases the computational costs [11]. More seriously, the inherent structure and correlation in the original data are destroyed. Recently, many efficient methods (see [11] for details) are proposed for DR of tensor data [5]–[8], [10]–[25], including multilinear principal component analysis (MPCA) [5], discriminant locally linear embedding (DLLE) [6], marginal fisher analysis (MFA) [7], higher order singular value decomposition (HOSVD) [8], multilinear discriminant analysis (MDA) [12], tensor canonical correlation analysis (TCCA) [13], correlation tensor analysis (CTA) [14], concurrent subspace analysis (CSA) [15], nonnegative tensor factorization (NTF) [16], uncorrelated multilinear principal component analysis (UMPCA) [19], uncorrelated multilinear discriminant analysis (UMLDA) [20], tensor rank-one decomposition (TROD) [21], tensor rank one discriminant analysis (TR1DA) [22], and incremental tensor analysis (ITA) [23]–[25]. Here it must be pointed out that all the aforementioned methods are based on the assumption that some fusion centers can gather together the tensor data acquired by different sensors (or nodes) or have access to them.

In recent years there has been a growing interest in sensor networks, which benefits from the progress in sensor techniques and distributed processing [26]–[32]. Especially, vision networks are being deployed for various applications such as security and surveillance, disaster response and environmental modeling [28], [32]. The conventional centralized scenario of sensor networks has several shortcomings. First, the resource of the fusion center is limited and thus hardly adapts to the storage and computational requirement of increasing nodes. Second, the sensitive or private data captured by different sensors (or nodes) are discouraged from being shared. Third,
the breakdown of the fusion center may cause the entire network to collapse. Therefore, in many applications a decentralized scenario (or a distributed scheme) is often preferred [33]–[35]. Especially in such an era of data explosion, it is extremely important for the network to solve the large-scale problem, including the decentralized storage of these datasets and decentralized processing [33]–[35]. In such a decentralized scenario [33]–[35], each network node is able to process the locally stored data and exchange information with neighboring nodes to find a globally optimal (or an approximately optimal) solution to the large-scale task [33]–[35].

Distributed storage and processing in sensor networks have promoted the development of distributed DR techniques [36], [37]. Potential applications include: i) novel multimedia architectures such as visual sensor networks with a large number of intelligent camera nodes (with built-in image sensors) [26]–[28], each of which is able to capture and process image data locally, and can collaborate with other cameras to complete an application-specific task; ii) in such an era of data explosion, the large volume of data tends to be stored across network nodes rather than a single node due to the fact that a single node has limited storage space and computational capacity [34]. Since high redundancies exist in the data [38], it is necessary to implement DR on them; iii) in privacy-sensitive tasks such as digital art authentication [39], [40], the distributed authentic artistic works across multiple museums are neither public nor shareable but need to be “integrated” to protect authentic works and distinguish imitations; and iv) other applications related to DR, including distributed coding, distributed fusion [41]–[45], etc.

In this paper, we present a novel decentralized dimensionality reduction (DDR) algorithm for distributed tensor data across sensor networks. It is worth highlighting the main aspects of the proposed algorithm here:

1) The conventional centralized and offline vector-based methods [5]–[8], [12]–[22] depend on the complete collection of (many) distributed tensor data. Among them, the tensor-to-tensor projection (TTP)-based methods [5]–[8], [12]–[18] simultaneously obtain all the vectors of the projection matrix related to some tensor modes. As a result, they are not suitable for the network environment. Other tensor-to-vector projection (TVP)-based methods [19]–[22] obtain the vectors in each elementary multilinear projection (EMP) group one-vector-by-one-vector (OVBOV) and then they utilize each EMP group to obtain only one projection or extract merely one feature of tensor data. Therefore, from the view of DR (dimensions), the TVP-based methods [19]–[22] imitate the VVP (used for DR of vectors) and thus utilize the projection vectors (PVs) inefficiently. For the purpose of efficient and distributed implementation, the proposed unsupervised DR method relaxes the simultaneous processing manner into the OVBOV manner to solve such a distributed DR problem in the network environment, i.e., obtaining the consistent unitary PVs for all nodes one by one except the first EMP group and ensuring the orthogonality properties among different vectors as well. Although the proposed method is similar to TTP, its successively obtained PVs are different from other tensor analysis methods [5]–[8], [12]–[22], including the unsupervised MPCA, CSA, and UMPCA methods [5], [15], [19];

2) The incremental tensor analysis methods [23]–[25] incrementally learn a tensor subspace for a tensor (changing with time), so they are actually online and centralized. In contrast, the proposed offline and decentralized DR algorithm is designed for DR of many distributed tensor data across sensor networks;

3) Although the centralized and offline vector-based “recursive” DR methods [46], [47] also determine PVs one by one, they require modifying the training data or within/between-class scatter matrices. Unlike them, the proposed DDR algorithm performs DR for distributed tensor data without modifying any data (i.e., it is not necessary to subtract the components with respect to the newly obtained PV), thus simplifying related computations;

4) [37] presents a distributed linear discriminant analysis method for supervised DR of vectors by estimating the scatter matrices; whereas the proposed algorithm pays more attention to distributed unsupervised DR of more common tensor data encountered in an era of data explosion [1]–[3], [37];

5) We cast the PV determination problem as a set of separable subproblems with consensus constraints using variable splitting so that it can be solved in a fully distributed fashion [34], [48], including the local PV determination step using local tensor data, the broadcast step of the local PV estimates, and the Lagrange multiplier update step;

6) We introduce the null space and transform the PV determination problem with complex orthogonality constraints into an equivalent hidden convex one [49]–[51] without any orthogonality constraint, which can be solved by the Lagrange multiplier method [49], [50].

The rest of this paper is organized as follows. Problem formulation is provided in Section II. The decentralized DR algorithm is developed for distributed tensor data in Section III. Experimental results are presented in Section IV. Conclusions are drawn in Section V.

Notation: Vectors, matrices, and tensors are denoted by boldface lowercase, uppercase, and underlined uppercase letters, respectively. $\| \cdot \|$ denotes the Frobenius norm of a tensor or a matrix, $(\cdot)^T$ denotes the transpose of a matrix or vector. $I$ denotes the identity matrix of an appropriate dimension. $x_{k,l}$ represents the $l$th element of the related set of the $k$th node. $\text{trace}(\cdot)$ denotes the trace of a matrix, and $\langle \cdot, \cdot \rangle$ represents the inner product of two vectors. Other mathematical symbols are
A generic sensor network, where blue circles stand for the nodes and black straight lines denote links among communicating nodes.

defined after their first appearances.

II. PROBLEM FORMULATION

Consider a network with $K$ nodes modeled by an undirected graph $G(\Pi, E)$ with vertices $\Pi = \{1, 2, \ldots, K\}$ corresponding to the nodes (blue circles), and edges $E$ (black straight lines) describing links among communicating nodes [29]–[35], as shown in Fig.1. Each node is capable of storing some local data and performing computations on them, and exchanging messages with its neighbors.

Assume there are $\sum_{k=1}^{K} N_k$ tensor data (or training samples) $\{X_{k,l}\}_{k=1,l=1}^{K,N_k}$ with dimensions $(I_1 \times I_2 \times \cdots \times I_N)$ distributed across the $K$ nodes, each with $N_k$ samples for $k = 1, \ldots, K$. When all the samples are collected by a fusion center, $X_{k,l}$ can be reduced into a low-dimensional one $Y_{k,l} = \sum_{l=1}^{N_k} U_{k,l}^{T} X_{k,l} U_{k,l}$ by the recently published centralized DR methods [5]–[8], [12]–[18], where the $I_n \times I_n$-dimensional projection matrix $U_{k,l}$ (consisting of $I_n$ column vectors) along the $n$th tensor mode is orthogonal, i.e., $U_{k,l}^{T} U_{k,l} = I_{I_n \times I_n}$, for $n = 1, \ldots, N$.

However, when the sensitive (or private) data are stored or there is no fusion center or there exist distributed stored large volume of data across sensor networks, as mentioned in Section I [34], a decentralized counterpart is usually needed. The objective of this paper is to find out common projection matrices $\{U_{n}\}_{n=1}^{N}$ for the distributed tensor data $\{X_{k,l}\}_{k=1,l=1}^{K,N_k}$ across the $K$ nodes. By some local computations on the tensor data and the communications (without exchanging any tensor data) among neighboring nodes, the $K$ nodes can finally obtain the consistent projection matrices $\{U_{n}\}_{n=1}^{N}$, which can be used for DR of the distributed stored tensor data $\{X_{k,l}\}_{k=1,l=1}^{K,N_k}$.

III. PROPOSED ALGORITHM

In this section, we first present an OVBOV DR scheme for distributed stored tensor data across sensor networks. Then, we prove that in the OVBOV manner we do not need to modify any tensor data (i.e., it is not necessary to subtract the components with respect to the newly obtained PV). Furthermore, we consider the distributed scenario in the PV determination problem with consensus constraints and solve it by processing tensor data locally and communicating among neighboring nodes. Finally, we introduce the null space into the local PV determination problem with complex orthogonality constraints and thus it can be solved efficiently by the Lagrange multiplier method [49], [50].

A. OVBOV Manner

When the distributed stored tensor data could be gathered together, many centralized schemes [1], [2] can determine the corresponding core tensors $\{Y_{k,l}\}_{k=1,l=1}^{K,N_k}$ and orthogonal projection matrices $\{U_{n}\}_{n=1}^{N}$ to approximate the tensors $\{X_{k,l}\}_{k=1,l=1}^{K,N_k}$

\[
\min_{\{Y_{k,l}\}_{k=1,l=1}^{K,N_k}} \sum_{k=1}^{K} \sum_{l=1}^{N_k} \|Y_{k,l} \times I_{1} \times 2 \times \cdots \times N U_{n}\|^{2},
\]

where the $I_{n}^{T}$ orthogonal unitary vectors $\{u_{n,1}, u_{n,2}, \ldots, u_{n,I_{n}}\}$ of $U_{n}$ for all $n = 1, \ldots, N$. Additionally, for each projection matrix $U_{n}^{k}$, it is necessary to ensure that its vectors are unitary and orthogonal to each other [5]–[8], [12]–[18]. Obviously consensus constraints of $U_{n} \in R^{I_{n} \times I_{n}}$, between two neighboring nodes are actually equivalent constraints of $2 \times I_{n} \times I_{n}$ variables; whereas for $U_{n}^{k}$ there are $I_{n}(I_{n}^{k})$ orthogonality constraints and $I_{n}$ unit norm (unitary) constraints, resulting in $\sum_{n=1}^{N} I_{n}(I_{n}^{k} - 1)/2$ orthogonality constraints and $\sum_{n=1}^{N} I_{n}^{T}$ unitary constraints for $N$ projection matrices $\{U_{n}^{k}\}_{n=1}^{N}$ of each node. As a result, the above-mentioned simultaneous manner (simultaneously obtaining $I_{n}$ orthogonal unitary vectors $\{u_{n,1}, u_{n,2}, \ldots, u_{n,I_{n}}\}$ of $U_{n}$) is not suitable for the network environment due to the fact that there are so many complex constraints among the $N \times K$ projection matrices $\{U_{n}^{k}\}_{n=1}^{N}, k=1, \ldots, K$ obtained by the $K$ nodes. Especially, as many as $\sum_{n=1}^{N} I_{n}$ unitary constraints on each node may cause the optimization problem to be extremely complex and even infeasible. Furthermore, some TVP-based methods [19]–[22] do not utilize the $N$ PVs in each EMP group efficiently due to the fact that one EMP group is used for obtaining only one projection or extracting merely one feature of tensor data.

For the purpose of efficient and distributed implementation, this paper considers relaxing the simultaneous computation...
as an OVBOV manner to solve the distributed DR problem: estimating the $N$ vectors $\{u_{n,1}\}_{n=1}^N$ and then determining $\{u_{n,p}\}_{n=1,p=2}$. To easily describe the vectors of the projection matrices in the aforementioned OVBOV scheme, we define $N$ register sets $\{U_n^0\}_{n=1}^N$ to store all the obtained vectors of the $N$ modes before the $(t+1)$th step for $t = 0, \cdots$. Similarly, we define $\{X_{k,l}^t\}_{K;N_k}$ as the modified tensor sensor data (subtracting the components with respect to the newly obtained PV after the $t$th step). Especially, $U_n^0 = [\ ]$ for $n = 1, \cdots, N$ and $X_{k,l}^0 = X_{k,l}$ for $k = 1, \cdots, K; l = 1, \cdots, N_k$.

Similar to the common rank-$\{1, \cdots, 1\}$ decomposition [1], [2], $\{u_{n,1}\}_{n=1}^N$ can be determined in the first step ($t = 1$) by solving the following optimization problem:

$$\min_{\{u_{n,1}\}} \sum_{k=1}^K \sum_{l=1}^{N_k} \|X_{k,l} - X_{k,l}^0\times U_{n,1}u_{l,1}\|^2,$$

where we do not define a new variable for the corresponding low-dimensional representation (or the so-called core tensor) but directly use $X_{k,l}^0\times U_{n,1}u_{l,1}$ to reduce the number of optimization variables [15].

Conventionally, (2) can be solved in an iterative manner [1], [2], where the $N$ PVs $\{u_{n,1}\}_{n=1}^N$ are updated alternately, i.e., with the current $(N-1)$ vector estimates $\{u_{n,1}\}_{n=1,n\neq m}^N$ fixed, $u_{n,1}$ is determined by solving the following optimization problem:

$$\min_{u_{m,1}} \sum_{k=1}^K \sum_{l=1}^{N_k} \|X_{k,l} - X_{k,l}^0\times U_{m,1}u_{m,1}\|^2,$$

where

$$X_{k,l} = X_{k,l}^0 \times U_{n,1}u_{n,l}.$$

OVBOV according to the prespecified determination orders of $\{u_{n,p}\}_{n=1,p=2}$, which should obey the following criteria: i) within the $n$th mode $u_{n,q}$ should be determined earlier than $u_{n,p}$ if $q < p$; and ii) for different modes (i.e. $m \neq n$), $u_{n,q}$ may be determined later than $u_{n,p}$ even if $q < p$. Without loss of generality, we consider estimating $u_{m,p}$ in the $t$th step ($t = 2, \cdots$):

$$\min_{u_{m,p}} \sum_{k=1}^K \sum_{l=1}^{N_k} \|X_{k,l} - X_{k,l}^0\times U_{n,1}u_{m,1}u_{m,2}\|^2,$$

where

$$u_{m,p} = 0, for \ q = 1, \cdots, p - 1.$$  

### B. Proof for Unnecessary Tensor Data Modification

**Lemma 1:** The problem in (4) is equivalent to the following optimization problem with the original tensor data $\{X_{k,l}\}_{k=1,l=1}^{K;N_k}$ rather than the modified ones $\{X_{k,l}\}_{k=1,l=1}^{K;N_k}$ required in (4):

$$\min_{u_{m,p}} \sum_{k=1}^K \sum_{l=1}^{N_k} \|X_{k,l} - X_{k,l}^0\times U_{n,1}u_{m,1}\|^2,$$

where

$$u_{m,p} = 0, for \ q = 1, \cdots, p - 1.$$  

which implies that it is not necessary to modify the tensor data $\{X_{k,l}\}_{k=1,l=1}^{K;N_k}$ (i.e., subtracting the components with respect to the newly obtained PV) when $u_{m,p}$ is determined in the $t$th step.

**Proof:** Submitting (5) into (4), we rewrite (4) as:

$$\min_{u_{m,p}} \sum_{k=1}^K \sum_{l=1}^{N_k} \|X_{k,l} - E_{k,l}\times U_{m,1}u_{m,2}\|^2,$$

where

$$E_{k,l} = X_{k,l}^0 \times U_{n,1}u_{m,1}\|^2,$$

and

$$F_{k,l} = S_{k,l}^i \times U_{n,1}u_{m,1}\|^2,$$

where

$$S_{k,l}^i = X_{k,l}^{i-1} \times U_{n,1}u_{m,1}\|^2,$$

and

$$F_{k,l} = E_{k,l} \times U_{n,1}u_{m,1}\|^2.$$  

Finally, after determining $\{u_{n,1}\}_{n=1}^N$, we set $\bar{U}_n = [u_{n,1}]$ for $n = 1, \cdots, N$ and modify the tensor data $\bar{X}_{k,l} = X_{k,l}^0 - X_{k,l}^0 \times U_{n,1}u_{n,1}$, which requires $O\left(\sum_{n=1}^N I_n \times I_n^N\right)$ multiplications for each tensor (each modification). Then, other vectors $\{u_{n,p}\}_{n=1,p=2}$ can be determined
\[ \times_m \left( u_{m,p} u_{m,p}^T \mathbf{U}_m^i (\mathbf{U}_m^i)^T \right). \] (10)

Note that \( u_{m,p} \) is determined at the \( t \)-th step, but \( \mathbf{U}_m^i \) is determined before the \( t \)-th step due to \( i \leq t - 1 \). Obviously, \( u_{m,p}^T \mathbf{U}_m^i = 0 \) holds for \( i \leq t - 1 \), which implies that \( F_k = 0 \) also holds for this case. Therefore, Eq. (7) can be re-written in a trace computation form as:

\[
\begin{align*}
\min_{u_{m,p}} & \sum_{k=1}^K \sum_{l=1}^{N_k} \left\| \tilde{X}_{k,l}^0(m) - u_{m,p} u_{m,p}^T E_{k,l}(m) \right\|^2 \\
\text{subject to} & \quad u_{m,p}^T u_{m,p} = 1, \\
& \quad u_{m,p}^T u_{m,p} = 0, \quad q = 1, \ldots, p - 1,
\end{align*}
\] (11)

where \( \mathbf{U}_m^i \) \( u_{m,p} \) is applied due to \( i \leq t - 1 \).

We rewrite Eq. (11) as the tensor form, which yields Eq. (6). As shown in (3) and (6), all the PVs \( \{ \tilde{X}_{k,l} \}_{k=1, l=1}^{K, N_k} \) cannot be gathered together. Therefore, we consider determining the PV \( u_{m,p} \) only via the local data computation and the information communication among neighboring nodes rather than the entire collection of \( \{ \tilde{X}_{k,l} \}_{k=1, l=1}^{K, N_k} \).

Similar to (3), (6) can be rewritten in an \( m \)-th mode matricization form as:

\[
\begin{align*}
\min_{u_{m,p}} & \sum_{k=1}^K \sum_{l=1}^{N_k} u_{m,p}^T \mathbf{H}_{k,l}(m) u_{m,p}, \\
\text{subject to} & \quad u_{m,p}^T u_{m,p} = 1, \\
& \quad u_{m,p}^T u_{m,q} = 0, \quad q = 1, \ldots, p - 1,
\end{align*}
\] (12)

where

\[
\begin{align*}
\mathbf{H}_{k,l}(m) &= \mathbf{Z}_{k,l}(m)(\tilde{X}_{k,l}(m))^T \\
&\quad - \tilde{X}_{k,l}(m)(\mathbf{Z}_{k,l}(m))^T,
\end{align*}
\] (13)

and \( \mathbf{Z}_{k,l} = \mathbf{X}_{k,l} \prod_{n=1, n \neq m} (\mathbf{U}_n^{i-1})^T \). We first replace \( u_{m,p} \) in (12) with \( K \) copies, denoted as local variables \( \{ u_{m,p}^{k}, u_{m,p}^{k'}, \ldots, u_{m,p}^{k'} \} \) distributed across the \( K \) nodes, yielding the equivalent form as follows:

\[
\begin{align*}
\min_{u_{m,p}^{k}, u_{m,p}^{k'}, \ldots, u_{m,p}^{k'}} & \sum_{k=1}^K N_k \sum_{l=1}^{N_k} (u_{m,p}^{k})^T \mathbf{H}_{k,l}(m) u_{m,p}^{k} \\
\text{subject to} & \quad u_{m,p}^{k} \in \mathbb{R}^K, \quad k \in \mathbb{R}^N, \quad k = 1, \ldots, K.
\end{align*}
\] (14)

Motivated by that the alternating direction method of multipliers (ADMM) [34], [48] with the superior convergence property coordinates the solutions of small local subproblems to find a solution to a large global problem, we introduce Lagrange multipliers \( \lambda_{k,k'} \) and \( \lambda_{k,k} \) separately corresponding to the constraints \( u_{m,p}^{k} = w_{m,p}^{k} \) and \( u_{m,p}^{k} = w_{m,p}^{k'} \) for \( k = 1, \ldots, K \) and \( k' \in \mathbb{R}^N \). Thus, the augmented Lagrangian of Eq. (15) can be written in an unconstrained form to determine \( \{ u_{m,p}^{k}, \lambda_{k,k'}, \lambda_{k,k}, w_{m,p}^{k} \} \) by solving the following optimization problem:

\[
\begin{align*}
\min_{u_{m,p}^{k}, w_{m,p}^{k}, \lambda_{k,k'}, \lambda_{k,k}} & \sum_{k=1}^K N_k (u_{m,p}^{k})^T \mathbf{H}_{k,l}(m) u_{m,p}^{k} \\
&\quad + \sum_{k=1}^K \lambda_{k,k}(u_{m,p}^{k} - w_{m,p}^{k}) \\
&\quad + \frac{\rho}{2} \sum_{k=1}^K \sum_{k'=1}^K \lambda_{k,k'}(u_{m,p}^{k} - w_{m,p}^{k'}) \\
&\quad + \frac{\rho}{2} \sum_{k=1}^K \sum_{k'=1}^K \lambda_{k,k'}(u_{m,p}^{k} - w_{m,p}^{k'}),
\end{align*}
\] (14)

where \( \rho > 0 \) is the so-called augmented Lagrangian parameter in ADMM (see [34] for details).

Using initialization values \( \{ u_{m,p}^{k}(0), \lambda_{k,k'}(0), \lambda_{k,k}(0), w_{m,p}^{k}(0) \} \), one can solve (16) via the alternate and iterative manner shown in (17)-(20) (similar convergence proof can be found in [48]):

\[
\{ u_{m,p}^{k}(j + 1) \}
\]
\[
= \arg \min_{\{u^k_{m,p}\}} \sum_{k=1}^{K} \sum_{l=1}^{N_t} (u^k_{m,p})^T H_{k,l}(m) u^k_{m,p} \\
+ \sum_{k=1}^{K} \langle \lambda_{k,k}(j), u^k_{m,p} - w^k_{m,p}(j) \rangle \\
+ \frac{\rho}{2} \sum_{k=1}^{K} \sum_{j=1}^{N_t} (u^k_{m,p} - w^k_{m,p}(j))^2_F \\
+ \sum_{k=1}^{K} \sum_{k' \in N_e(k)} \langle \lambda_{k,k'}(j), u^k_{m,p} - w^k_{m,p}(j) \rangle \\
+ \frac{\rho}{2} \sum_{k=1}^{K} \sum_{k' \in N_e(k)} \|u^k_{m,p} - w^k_{m,p}(j)\|^2_F
\]

subject to

\[
\arg \min_{\{w^k_{m,p}\}} \sum_{k=1}^{K} \sum_{l=1}^{N_t} (w^k_{m,p})^T H_{k,l}(m) w^k_{m,p} \\
+ \sum_{k=1}^{K} \langle \lambda_{k,k}(j), w^k_{m,p} \rangle \\
+ \frac{\rho}{2} \|u^k_{m,p} - w^k_{m,p}(j)\|^2_F \\
+ \frac{\rho}{2} \sum_{k' \in N_e(k)} \|u^k_{m,p} - w^k_{m,p}(j)\|^2_F
\]

where \(j\) is the iteration number, Lagrange multipliers \(\lambda_k = \lambda_{k,k} + \sum_{k' \in N_e(k)} \lambda_{k,k'}\) and \(\lambda_k = \lambda_{k,k} + \sum_{k' \in N_e(k)} \lambda_{k,k'}\). Additionally, (18) utilizes the fact that the undirected graph is actually a symmetric one (i.e., if \(k' \in N_e(k)\), then \(k \in N_e(k')\)) [33–35].

Considering the relationships between \(\{\lambda_{k,k}, \lambda_{k,k'}, \lambda_{k,k'}\}\) and \(\{\lambda_k, \lambda_k\}\), we update \(\lambda_k\) and \(\lambda_k\) as:

\[
\hat{\lambda}_k(j + 1) = \lambda_k(j) + \sum_{k' \in N_e(k)} \rho u^k_{m,p}(j + 1) + \rho u^k_{m,p}(j + 1) \\
- (\rho \text{Card}(k) + 1) w^k_{m,p}(j + 1)
\]

for \(k = 1, \ldots, K, k' \in N_e(k), \) (19)

and

\[
\lambda_k(j + 1) = \lambda_k(j) + \rho \text{Card}(k + 1) u^k_{m,p}(j + 1) - \rho w^k_{m,p}(j + 1) \\
- \sum_{k' \in N_e(k)} \rho w^k_{m,p}(j + 1)
\]

for \(k = 1, \ldots, K, k' \in N_e(k), \) (20)

where \(\text{Card}(k)\) denotes the neighbor number of the \(k\)th node.

Eqs. (17) and (18) imply that \(u^k_{m,p}(j + 1)\) and \(w^k_{m,p}(j + 1)\) for \(k = 1, \ldots, K\) can be solved in a separate and parallel form [34]. Therefore, we divide Eqs. (17) and (18) into \(K\) subproblems (with unitary and orthogonality constraints):

\[
\text{arg} \min_{u^k_{m,p}} \sum_{l=1}^{N_t} (u^k_{m,p})^T H_{k,l}(m) u^k_{m,p} \\
+ \langle \lambda_k(j), u^k_{m,p} \rangle \\
+ \frac{\rho}{2} \|u^k_{m,p} - w^k_{m,p}(j)\|^2_F \\
+ \frac{\rho}{2} \sum_{k' \in N_e(k)} \|u^k_{m,p} - w^k_{m,p}(j)\|^2_F
\]

subject to \((u^k_{m,p})^T u^k_{m,p} = 1\),

\(u^k_{m,p} = 0, \) for \(q = 1, \ldots, p - 1\) (21)

and

\[
\text{arg} \min_{w^k_{m,p}} \sum_{l=1}^{N_t} (w^k_{m,p})^T H_{k,l}(m) w^k_{m,p} \\
+ \langle \lambda_k(j), -w^k_{m,p} \rangle \\
+ \frac{\rho}{2} \|u^k_{m,p} - w^k_{m,p}(j)\|^2_F \\
+ \frac{\rho}{2} \sum_{k' \in N_e(k)} \|u^k_{m,p} - w^k_{m,p}(j)\|^2_F
\]

subject to \((w^k_{m,p})^T w^k_{m,p} = 1\),

\(w^k_{m,p} = 0, \) for \(q = 1, \ldots, p - 1\) (22)

for \(k = 1, \ldots, K\).
Obviously, \( \{ u_{m,p}^k(j+1), w_{m,p}^k(j+1) \} \) can be determined only by processing the local data \( \{ X_{k,l}(m), Z_{k,l}(m) \} \) and exchanging \( \{ w_{m,p}^k(j), u_{m,p}^k(j+1) \} \).

### D. Solutions to (21) and (22)

Although \( \sum_{l=1}^{N} H_{k,l}(m) \) in (21) is symmetric but not nonnegative definite, (21) is actually hidden convex [51]. However, complex orthogonality and unitary constraints make (21) difficult to solve. Here we consider introducing null space to simplify (21). Define \( B_{m,p}^k \) as an orthogonal basis for the null space of \( [u_{m,1}^k \ u_{m,2}^k \ldots u_{m,p-1}^k] \) obtained from the related singular value decomposition (or see Matlab command “null”), i.e., \( [u_{m,1}^k \ u_{m,2}^k \ldots u_{m,p-1}^k]^T B_{m,p}^k = 0 \). Obviously, \( (B_{m,p}^k)^T B_{m,p}^k = I \) and thus \( u_{m,p}^k \) can be represented as the linear combination of \( B_{m,p}^k \), i.e., \( u_{m,p}^k = B_{m,p}^k e_{m,p}^k \).

Since
\[
(\sum_{l=1}^{N} H_{k,l}(m)) u_{m,p}^k = (\sum_{l=1}^{N} H_{k,l}(m)) B_{m,p}^k e_{m,p}^k = (e_{m,p}^k)^T \sum_{l=1}^{N} H_{k,l}(m) B_{m,p}^k
\]
and
\[
[\sum_{l=1}^{N} H_{k,l}(m)] u_{m,p}^k = \sum_{l=1}^{N} H_{k,l}(m) B_{m,p}^k e_{m,p}^k = 0 \times e_{m,p}^k = 0,
\]
Eq. (21) can be simplified in another form as:
\[
\min_{e_{m,p}^k} \sum_{l=1}^{N} (e_{m,p}^k)^T (B_{m,p}^k)^T H_{k,l}(m) B_{m,p}^k + \langle \lambda_k(j), B_{m,p}^k e_{m,p}^k \rangle
\]
\[
+ \frac{\rho}{2} \| B_{m,p}^k e_{m,p}^k - w_{m,p}^k(j) \|_F^2
\]
\[
+ \frac{\rho}{2} \sum_{k' \in N(k)} \| B_{m,p}^k e_{m,p}^k - w_{m,p}^k(j) \|_F^2
\]
\[
\text{Subject to } (e_{m,p}^k)^T e_{m,p}^k = 1,
\]
where the orthogonality constraints are hidden due to the introduction of the null space, yielding the simplified optimization problem.

Define
\[
A_k = \sum_{l=1}^{N} (B_{m,p}^k)^T H_{k,l}(m) B_{m,p}^k
\]
and
\[
b_k^T = \frac{1}{2} \left( \lambda_k(j) - \rho \sum_{k' \in N(k)} w_{m,p}^{k'}(j) - \rho w_{m,p}^k(j) \right)^T B_{m,p}^k
\]
Eq. (25) can be rewritten in a compact form as:
\[
\min_{e_{m,p}^k} (e_{m,p}^k)^T A_k e_{m,p}^k + 2b_k^T e_{m,p}^k
\]
\[
\text{Subject to } (e_{m,p}^k)^T e_{m,p}^k = 1.
\]
Although the matrix \( A_k \) is not nonnegative definite, it is symmetric and thus the optimization problem shown in Eq. (28) is still hidden convex [51]. It is worth mentioning that unlike (21), there is only one constraint in (28), leading to the efficient implementation by the Lagrange multiplier method [42], [49], [50].

Then, we solve the optimization problem in (22). Define
\[
d_k^T = (\lambda_k(j) + \rho u_{m,p}^k(j+1) + \rho \sum_{k' \in N(k)} u_{m,p}^{k'}(j+1))^T B_{m,p}^k
\]
and
\[
e_{m,p}^k = (B_{m,p}^k)^T w_{m,p}^k,
\]
and thus Eq. (22) can be rewritten in a compact form as:
\[
\min_{e_{m,p}^k} -d_k^T e_{m,p}^k \text{ Subject to } (e_{m,p}^k)^T e_{m,p}^k = 1.
\]
Obviously, the solution to (31) is given by
\[
e_{m,p}^k = \frac{d_k}{\| d_k \|}
\]
and thus \( w_{m,p}^k = B_{m,p}^k e_{m,p}^k \).

### E. Description and Discussion of the Proposed Algorithm

Note that (3) is actually a special case of (12), where \( p = 1 \) and there is no orthogonality constraint on \( u_{n,1}^k \) for \( m = 1, \ldots, N \). Additionally, the introduction of the null space simplifies (12) and transforms it into another problem (on \( e_{m,p}^k \)) without orthogonality constraints as well. As a result, the distributed implementations of both (12) and (3) reduce into the same form with only one unitary constraint (see (25) or (28)). Therefore, both of them can be solved by the Lagrange multiplier method [42], [49], [50].

Based on the above derivation and discussion, the proposed DDR algorithm can be summarized as follows:

#### DDR Algorithm

**Initialization:** Setting empty sets \( \overline{U}_0^i(k) \), the same initial projection vectors \( \{ u_{n,1}^k \}_{n=1}^N \) and \( \rho \) for all the \( K \) nodes; for \( t = 1, \ldots, \sum_{n=1}^{N} I_n - N + 1 \) for \( j = 1, \ldots, J \) for \( k = 1, \ldots, K \)

If \( t = 1 \) (no orthogonal constraint), one can determine \( u_{n,1}^k \) via (14)-(21) and the Lagrange multiplier method [49], [50], and broadcast \( u_{n,1}^k \) to the related neighbors for \( n = 1, \ldots, N \); Then, one applies (22) and (29)-(32) to
determine $w_{1,n}^k$ (without null space) and broadcasts $w_{2,n}^k$ to the related neighbors for $n = 1, \ldots, N$.

Else $t \neq 1$ (with null space), one can determine $u_{m,p}^k$ ($m$ and $p$ are obtained from the pre-specified orders) by applying (14)-(21), (23)-(28), and the Lagrange multiplier method [49], [50], and broadcast $u_{m,p}^k$ to the related neighbors. Then, one applies (22) and (29)-(32) to determine $w_{3,m,p}^k$ (with null space) and broadcasts $w_{4,m,p}^k$ to the related neighbors;

Update $\lambda_k(j + 1)$ using (20);
Update $\lambda_k(j + 1)$ using (19);
end for $k$.

end for $j$ until algorithm convergence.

Append $u_{1,n}^k$ to sets $\bar{U}_{2,n}(k)$ for $t = 1$; or append $u_{2,m,p}^k$ to sets $\bar{U}_{3,m}(k)$ for $t \neq 1$;
end for $t$ or other stopping criteria are met.

Here we evaluate the required communication and computation costs of the proposed DDR method. For each node, it requires $O(N^2 + \sum_{n=1}^N l_n^2) \times \prod_{n=1}^N l_n^2 + \sum_{n=1}^N \frac{1}{2} l_n^2 J + \sum_{n=1}^N \frac{1}{2} (I_n - 1) l_n^2 J$ multiplications, and broadcasts $\sum_{n=1}^N l_n J$ vectors to its neighbors.

Since the convergence of ADMM [34] plays an important role in the proposed DDR method, one can terminate the iteration related to $j$ by checking whether the maximal error $\max_{k \in \{1, \ldots, K\}} \|u_{m,p}^k - u_{m,p}^{k-1}\|_1$ is less than a prespecified threshold (e.g., $10^{-3}$) or by setting a large enough iteration number for $J$ (e.g., 1000 for the following experiments).

IV. SIMULATION AND EXPERIMENTAL RESULTS

To evaluate the proposed DDR method for the distributed stored tensor data across sensor networks, we use the databases, including Berkeley Multiview Wireless (BMW) database [9], UMIST face database [52], and Gait Challenge Data Sets (GCDS) [53]. To analyze the effect of network structure and number of nodes on the performance of the proposed DDR method, we use UMIST and GCDS databases as the simulated network data in the first and third experiments, which are often used for sensor network research [43], [45]; Whereas in the second experiment, we deal with the real sensor network data (BMW database), which are obtained via the fixed node and network configuration. To assess the DR performance of the proposed method, we determine the final classification after DR with the nearest neighbor or $k$-nearest neighbor criterion in all the experiments and compare the proposed decentralized DR method with centralized MPCA, DLLE, MFA, MDA, CSA, and CTA algorithms [5]-[7], [12], [14], [15].

A. Experiment 1: Decentralized DR on Multi-View Face Images (2nd-order tensor)

Similar to the distributed classification algorithm in [43], we also use the UMIST database [52] in the first experiment to simulate the data of the vision network in Fig. 1, where the faces of a subject are captured by different-view cameras in the network and each observation represents a facial image captured under some viewing angles [43]. Unlike [43], which designed a semi-supervised classifier based on the average-consensus scheme, this paper applies the UMIST database for decentralized unsupervised DR research.

The UMIST database [52] contains 575 multi-view face images of 20 individuals under different poses, which are cropped and then resized to $112 \times 92$ pixels. Fig. 2 illustrates a snapshot of the simulated network, in which the facial images denote the intelligent cameras with the corresponding viewing angles. In order to simulate the sensor network, we assign 200 samples to the 10 nodes at random, and use them as the training ones (20 images for each person) and the rest for testing.

To implement DDR on the 200 samples distributed across the 10-node network and obtain the consistent projection matrices $U_{2,n}(k)_{n=1,k=1}^{2,10}$ for the 10 nodes, the proposed algorithm initializes projection vectors $u_{1,n}^k_{n=1,k=1}^{2,10}$ using the randomly-generated normalized column vector. Additionally, the step size $\rho = 0.1$, the iteration number $J = 1000$, and the order of other projection vectors is set as $\{u_{1,1}, u_{2,2}, u_{1,3}, u_{2,3}, \ldots, u_{1,10}, u_{2,10}\}$, which mean that $I = 10$, $I_3 = 10$, and the original $112 \times 92$ second-order tensors are reduced into $I_1 \times I_2^2$-dimensional ones.

To evaluate the consistency and convergence performance of the proposed algorithm, we define four evaluation criteria:

1) Average coefficient Difference Errors Among Nodes (DEAN) for the related projection matrix estimates:

$$\text{DEAN}(n,k) = \frac{1}{I_n I_n^2 (K - 1)} \sum_{l=1}^K \|\bar{U}_n(l) - \bar{U}_n(k)\|_1,$$

where $\text{DEAN}(n,k)$ corresponds to the $n$th projection matrix at the $k$th node. For convenience of plotting, we construct them
as a $1 \times K$-dimensional vector $\text{DEAN}(n) = [\text{DEAN}(n, 1), \text{DEAN}(n, 2), \ldots, \text{DEAN}(n, K)]$ for $n = 1, \ldots, N$;  

2) Average coefficient Difference Errors between the projection matrices obtained from the Centralized (gathering all tensor data together but in the OVBOV manner) and Distributed (DECD) manners:

$$\text{DECD}(n, k) = \frac{1}{n} \sum_{i=1}^{n} || \mathbf{U}_n(k) - \mathbf{U}_n ||_1,$$  

where $\mathbf{U}_n$ represents that of the $n$th projection matrix obtained by the centralized manner. Similarly, we construct a $1 \times K$-dimensional vector $\text{DECD}(n) = [\text{DECD}(n, 1), \text{DECD}(n, 2), \ldots, \text{DECD}(n, K)]$ for $n = 1, \ldots, N$;  

3) Difference Errors (DE) of the vector estimate $\mathbf{u}_{n,p}$ of the $k$th node between the $j$th and $j + 1$th iterations:

$$\text{DE}(n, p, j, k) = \frac{1}{n} || \mathbf{u}_{n,p}^k(j + 1) - \mathbf{u}_{n,p}^k(j) ||_1,$$  

4) Orthogonal Property (OP) of PV pairs ($\mathbf{u}_{n,p}$ and $\mathbf{u}_{n,q}$) at the $k$th node, for $n = 1, \ldots, N, p, q = 1, \ldots, I_n$:

$$\text{OP}(n, k, p, q) = || (\mathbf{u}_{n,p}^k)^T \mathbf{u}_{n,q}^k ||.$$

We apply the proposed DDR algorithm into the face images and compute the DEANs and DECDs, as shown in Fig. 3. It can be found from Fig. 3 that all the average coefficient difference errors among these nodes or between the distributed and centralized ones are not larger than $10^{-11.9988}$. Clearly, the proposed algorithm can obtain consistent projection matrices with negligible difference for all nodes and also for the distributed and centralized cases. Therefore, the projection matrices obtained by the decentralized manner can be similarly used for the DR purpose as the centralized ones. Fig. 4 shows that the DEs are less than $10^{-12.0618}$ after 400 iterations (e.g., $n = 1, k = 10$), which implies that the proposed algorithm has the satisfactory convergence performance. Besides, Fig. 5 displays the inner products of any two PVs (for space reason only those of the 1-mode are given here, i.e., $n = 1, k = 1$) to test the orthogonality performance of the successively obtained PVs. It can be seen from Fig. 5 that the maximal value of inner products of any two PVs (except the inner product of the PV and itself) is not larger than $10^{-15.3525}$, which implies that the successively obtained PVs are orthogonal to each other.

To assess the DR performance of the proposed algorithm, we apply the nearest neighbor criterion to determine the class labels of the testing samples. Actually, the nodes with the almost entirely same projection matrices have the same recognition performance for all experiments due to the efficient consistency property of the proposed DDR method. Therefore, in all experiments we only plot the recognition rates from the projection matrices of the 1st node. When the number of features (NF) varies from 1 to 100, the recognition rates of MPCA, DLLE, MFA, MDA, CSA, CTA, and DDR are computed and plotted in Fig. 6 for comparison purpose. It can be seen that: i) the recognition rates of all the seven methods rise basically with the increase of NF (from 1 to 100); ii) the highest recognition rate achieved by the CSA method equals 0.9733; and iii) the DDR method with the distributed computation manner has the approximate recognition property as the centralized DR methods. Additionally, we compute the running time (Computer configuration: Intel R, CPU 2.0, 2-GB memory) of MPCA, DLLE, MFA, MDA, CSA, CTA, and DDR for the DR task in Fig. 6, which are 74.69, 9004.16, 17422.93, 102.84, 65.05, 23721.98 and 9325.91 seconds, respectively. However, it must be pointed out that in this simulation the DDR method imitates the 10-node network on
one computer.

Fig. 6: Recognition rates by different algorithms in Exp. 1

Next, we explore the influence of the varying neighbors of each node on the performance of the proposed algorithm (see Fig.7(a), Fig.1, Fig.7(b)). Due to space reason and that the consensus property among these nodes are nearly the same as described before, we only plot the DECDs of the 1st node in Fig. 8, which clearly shows that all DECDs are not larger than $10^{-10.8955}$.

Finally, the effect of the number of nodes on the performance of the proposed algorithm is investigated. When the number of nodes varies from 10 to 25 (the corresponding training sample number of each node varies from 20 to 8), we compute the DECDs and plot them in Fig. 9, which shows that all DECDs are less than $10^{-11.7342}$. Clearly, similar to the above test, the obtained PVs via the networks with different number of nodes are all almost equivalent to those from the centralized manner.

**B. Experiment 2: Decentralized DR on Landmark Buildings Images (3rd-order tensor)**

In the second experiment, we use the real vision network data (i.e., BMW database [9]), which is captured to aid the evaluation of distributed object recognition methods for the wireless surveillance scenario [44]. The BMW database consists of multiple-view images of 20 landmark buildings on the campus of University of California, Berkeley. For each building, 16 different vantage points have been selected to measure the 3-D appearance of the building. The apparatus for image acquisition incorporates five low-power CITRIC camera sensors [9] on a tripod, as shown in Fig. 10. The cameras on the periphery of the cross are named Cam 0, Cam 1, Cam 4, Cam 3 with a counter-clockwise naming convention, and the center camera is named Cam 2. Each camera acts as a node and thus they form a 5-node vision network. The obtained BMW database via this network consists of 1600 RGB color images, each of them is resized to $96 \times 128 \times 3$-dimension, as shown in Fig. 11. Unlike [44], which used the feature descriptors such as SIFT and SURF to recognize the objects, this paper applies the BMW database for decentralized unsupervised DR research.

In this experiment, we use 1000 images for training (10 images for each building and each camera) and the rest for testing. Additionally, we adopt the same step size and the iteration number as the first experiment. Furthermore, we initialize projection vectors $\{u_{n,1}^k\}_{n=1,k=1}^{3,5}$ for the distributed third-order tensors with the orders of other projection vectors as $\{u_{1,2}, u_{2,2}, u_{3,2}, u_{1,3}, u_{2,3}, u_{3,3}, u_{1,4}, u_{2,4}, u_{1,5}, \ldots, u_{1,10}\}$.
We compute the DEANs, DECDs, DEs and OPs, the first two of which are plotted in Fig. 12. From Fig. 12, we can find that they are all less than $10^{-11.5704}$. Additionally, DEs (attaining $10^{-14.8425}$ after 120 iterations) and OPs (maximum: $10^{-15.1764}$ except the inner product of the PV and itself) respectively plotted in Figs. 13 and 14 imply that the proposed algorithm has satisfactory convergence and orthogonality performance.

![Fig. 10: The apparatus that instruments five camera sensors in Exp. 2 [9]](image)

(a) Five small-baseline images captured at one vantage point.

![Fig. 11: Building (the Campanile at UC Berkeley) image examples of the BMW database in Exp. 2 [9]](image)

(b) Five large-baseline images captured at different vantage points.

For space reason, we only show the recognition rates of seven algorithms in Fig. 15, which are obtained via applying the nearest neighbor criterion to determine the classification of the testing samples. From Fig. 15, we can see that: i) when the dimensions increase to 108, the supervised MFA method with graph embedding obtains the highest recognition rate of 0.9466; ii) the recognition performance of the proposed DDR method is comparable to those of MPCA, DLLE, CTA, and CSA.

C. Experiment 3: Decentralized DR on Gait Video Sequences (3rd-order tensor)

To further assess the performance of the proposed DR method, the third experiment considers a simulated visual camera sensor network (with an annular topology) in a residential area [26]–[28], which consists of several intelligent surveillance cameras. Each camera is able to extract the gait sequences from the captured video and complete the gait recognition task for social security purpose [45].

In this experiment, we use the GCDS database with version 1.7 [53], in which there are 3 covariates for each person: viewing angles (Left/Right), shoe types (A/B), and walking surfaces (Grass/Cement). We use the 71 sequences (subjects) of $128 \times 88 \times 120$-dimension for training, which are captured under the condition (Grass surface, shoe type A, and Left view, abbreviated as GAL) while another 71 sequences with right view for testing. We assign 70 of all 71 sequences to
the 7 nodes at random, each with 10 sequences, which are captured by 7 intelligent surveillance cameras. Since each sequence is of 6 cycles, we divide each sequence into 6 tensors (i.e., 6 cycles, each with $128 \times 88 \times 20$-dimension) and thus in each node there are 60 tensors. Similarly, the related 70 test sequences are divided into tensors in a similar manner. This experiment adopts the similar parameters as other experiments. We initialize projection vectors $\{u_{n,1}\}_{n=1}^{3,7}$ for the distributed third-order tensors in this experiment but with the orders of other projection vectors as $\{u_{1,2}, u_{2,2}, u_{3,2}, u_{1,3}, u_{2,3}, u_{3,3}, u_{1,4}, u_{2,4}, u_{3,4}, \ldots, u_{1,10}, u_{2,10}, u_{3,10}\}$.

Similar to other gait recognition methods [6, 7], we use the rank−1/5 rates to evaluate the proposed DDR algorithm for determining the classification of the testing samples. For comparison purpose, we implement the MPCA, DLLE, MDA, MFA, CSA, and CTA methods simultaneously on the above-mentioned GAL condition and other conditions such as GBR, GBL, CAR, CBR, CAL, and CBL (similar abbreviations as GAL). Table I shows the best recognition results of all the methods with the rank−1/5 rates. From the table, we can see that: i) comparing with the rank−1 rates, the rank−5 rates can improve the recognition performance for all the methods and also all the capturing conditions; and ii) the proposed DDR algorithm has comparable recognition performance as these efficient centralized DR methods.

### Table I: Recognition rates by different algorithms in Exp. 3 (rank−1/5 rates)

<table>
<thead>
<tr>
<th>Condition</th>
<th>DDR</th>
<th>MPCA</th>
<th>DLLE</th>
<th>MDA</th>
<th>MFA</th>
<th>CSA</th>
<th>CTA</th>
<th>MDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAL</td>
<td>0.797/0.925</td>
<td>0.771/0.925</td>
<td>0.757/0.925</td>
<td>0.742/0.925</td>
<td>0.714/0.925</td>
<td>0.771/0.925</td>
<td>0.742/0.925</td>
<td>0.741/0.925</td>
</tr>
<tr>
<td>GBR</td>
<td>0.775/0.950</td>
<td>0.800/0.950</td>
<td>0.650/0.950</td>
<td>0.675/0.975</td>
<td>0.575/0.825</td>
<td>0.750/0.950</td>
<td>0.750/0.950</td>
<td>0.750/0.950</td>
</tr>
<tr>
<td>GBL</td>
<td>0.750/0.9250</td>
<td>0.775/0.9250</td>
<td>0.725/0.9250</td>
<td>0.725/0.9250</td>
<td>0.725/0.9250</td>
<td>0.725/0.9250</td>
<td>0.725/0.9250</td>
<td>0.725/0.9250</td>
</tr>
<tr>
<td>CAR</td>
<td>0.414/2.785</td>
<td>0.414/2.785</td>
<td>0.528/3.714</td>
<td>0.457/3.8000</td>
<td>0.285/7.0000</td>
<td>0.400/7.0000</td>
<td>0.400/7.285</td>
<td></td>
</tr>
<tr>
<td>CBR</td>
<td>0.550/0.9250</td>
<td>0.525/0.9250</td>
<td>0.600/0.9000</td>
<td>0.725/0.9500</td>
<td>0.575/0.8500</td>
<td>0.525/0.9000</td>
<td>0.625/0.8250</td>
<td>0.625/0.8250</td>
</tr>
<tr>
<td>CAL</td>
<td>0.457/0.8142</td>
<td>0.442/0.8000</td>
<td>0.557/0.8571</td>
<td>0.657/0.8571</td>
<td>0.471/0.8571</td>
<td>0.457/0.8142</td>
<td>0.571/0.8571</td>
<td>0.571/0.8571</td>
</tr>
<tr>
<td>CBL</td>
<td>0.650/0.9500</td>
<td>0.650/0.9500</td>
<td>0.725/0.9250</td>
<td>0.650/0.9000</td>
<td>0.500/0.8000</td>
<td>0.650/0.9500</td>
<td>0.625/0.9750</td>
<td>0.625/0.9750</td>
</tr>
<tr>
<td>Average</td>
<td>0.620/0.8969</td>
<td>0.625/0.9228</td>
<td>0.648/0.8989</td>
<td>0.661/0.9497</td>
<td>0.549/0.8372</td>
<td>0.618/0.8811</td>
<td>0.634/0.8933</td>
<td>0.634/0.8933</td>
</tr>
</tbody>
</table>

**Fig. 15:** Recognition rates by different algorithms in Exp. 2

**REFERENCES**


Junli Liang was born in China. He received the Ph.D. degree in signal and information processing from the Institute of Acoustics, Chinese Academy of Sciences. Currently, he is working as a professor at School of Electronics and Information, Northwestern Polytechnical University, China. His research interests include signal processing and image processing, and their applications.

Guoyang Yu was born in China. His research interests include signal processing, and image processing. Badong Chen was born in China. Currently, he is working as a professor at Xi’an Jiaotong University. His research interests include neural network and signal processing.

Minghua Zhao was born in China. Her research interests include image processing and recognition.