A Hypercomplex Tensor-SVD and Its Application

Takehiko Mizoguchi* and Isao Yamada†
*NEC Laboratories America, Inc., Princeton, NJ, USA
†Department of Information and Communications Engineering, Tokyo Institute of Technology, Tokyo, Japan
E-mail: {mizoguchi,isao}@sp.ce.titech.ac.jp

***Abstract*—Expressing multidimensional information as a value in hypercomplex number systems (e.g., quaternion, octonion, etc.) has great potential in signal processing applications. The benefits from the algebraically natural operations of hypercomplex numbers can be inherited as well in hypercomplex tensors, which can be widely used for representing huge volume of multi-relational datasets. In this paper, we present a new algebraic real translations of hypercomplex tensors in order to extend tensor-SVD (t-SVD) and related mathematical tools such as tensor tubal rank and tensor low tubal rank approximation into general (Cayley-Dickson) hypercomplex domain. We then propose an algorithmic solution to the hypercomplex tensor principal component pursuit based on a proximal splitting technique. Numerical experiments are performed in quaternion domain and show that the proposed algorithm outperforms a part-wise state-of-art real and complex algorithm.**

I. Introduction

The hypercomplex number system is one of the most effective models for representing multidimensional information because we can express such information not in terms of vectors but in terms of numbers among which we can define the four basic arithmetic operations. Indeed, it has been used in many areas such as computer graphics [1], robotics [2], [3] and wind forecasting [4], [5]. In the statistical signal processing field, effective utilization of the m-dimensional Cayley-Dickson number system (C-D number system) [6], [7], which is a standard class of hypercomplex number systems [8], including, e.g., real $\mathbb{R}$, complex $\mathbb{C}$, quaternion $\mathbb{H}$, octonion $\mathbb{O}$ and sedenion $\mathbb{S}$ etc., have been investigated [9].

A hypercomplex number has one real part and possibly many imaginary parts, and it can represent multidimensional data as a single number. For any pair of such numbers, the four arithmetic operations including multiplication and division are available (Note: the four arithmetic operations are not available with ordinary real multidimensional vectors). Certainly the natural operations in hypercomplex number system have great potential for such modelings of various correlations. For example, the multiplication of hypercomplex numbers can enjoy interactions among real and imaginary parts, algebraically. However, because of the "singularity" of higher dimensional C-D number systems (see e.g., Example 1), few mathematical tools have been maintained [10]. To overcome this situation, over the past few years, we have proposed several essential mathematical tools, such as algebraic real translations for clarifying the relation between C-D linear system and real vector valued linear systems [11], singular value decomposition and rank evaluation of C-D matrices which is consistent with those of complex as well as quaternion matrices [12].

The benefits from the algebraically natural operations can be inherited as well in hypercomplex multi-way arrays, i.e., tensors. In the areas of signal, image processing, and data science, representing huge volume of multi-relational datasets by tensors assuming low rank structure, has results in remarkable achievements in many applications [13], [14]. To exploit such low rank structure, the discussion about low-rankness of tensors is indispensable.

However, the numerical algebra of tensors is fraught with hardness results for applications [15]. The main issue for estimating low-rank tensor, is originated from the definition of tensor rank. Unlike the matrix rank, the tensor rank has not yet been well-established from application point of view as several different definitions of tensor rank have been proposed. For example, CANDECOMP/PARAFAC (canonical polyadic, CP) decomposition [16]–[18], which is typically used to decompose a tensor into the sum of smallest number of rank one tensors, is generally NP-hard and its convex relaxation is intractable. Another direction is to use Tucker rank [14] (n-rank) introduced as a vector-valued rank based on the matrix ranks of mode-n matricization of a tensor and the sum of convex relaxations of their components (We have already extended this approach to hypercomplex domains in [12]). Recently, the tensor tubal rank based on a new tensor decomposition scheme called *tensor singular value decomposition (t-SVD)* has been proposed [19]. The t-SVD is based on the *t-product*, a new notion for tensor-tensor product. The t-SVD enjoys many similar properties to the matrix case. However, t-SVD is well-defined only up to complex domain since it requires to perform SVD in Fourier domain for efficient computation. In quaternion domain, Fourier transform is available [20] but it is hard to be utilized for t-SVD in quaternion domain because of the singularity of quaternion such as non-commutativity of multiplication and arbitrarity of quaternion Fourier transform.

In this paper, to extend the notion of t-SVD to be applicable in hypercomplex domain, we propose new algebraic real translations of hypercomplex tensors, all of whose entries are C-D numbers. The proposed translations are based on the algebraic translations of C-D matrices proposed in [11]. We show that the t-product of any two hypercomplex tensors can be equivalently transformed into that of two translated real tensors, which can be efficiently computed with the fast Fourier transform (FFT) in complex domain, thanks to the algebraic properties of proposed real translations. We then propose new hypercomplex variants of t-SVD, tensor multi rank, tensor tubal rank and a tensor low tubal rank approximation technique based on the proposed real translations. The proposed approaches are natural extensions of t-SVD related techniques in [19]. As an application to practical hypercomplex tensor recovery problems, we also present hypercomplex tensor principal component pursuit (C-D tensor PCP) as a convex relaxation of the tensor robust principal component analysis (tensor RPCA) in C-D domain. Similar to the matrix case in [21], the C-D tensor PCP can be modeled as a convex optimization under a certain structure in real domain and solvable with proximal splitting techniques. We finally propose a hypercomplex tensor principal component pursuit algorithm, $\mathbb{A}_m$-DRS-TPCP based on *Douglas-Rachford splitting (DRS)* [22]. The proposed algorithm is a higher order tensor generalization of hypercomplex PCP algorithm ($\mathbb{A}_m$-DR-PCP) proposed in [21] and can be applied to general C-D domains.

Numerical experiments are performed in the context of recovering sparsely corrupted low tubal rank tensors in quaternion domain and demonstrate that the proposed algorithm successfully utilizes algebraically natural correlations of real and all imaginary parts to recover much more faithfully the original tensors, corrupted randomly by noise, than a part-wise real and complex tensor PCP algorithms.

II. Preliminaries

A. Hypercomplex Number System

Let $\mathbb{N}$ and $\mathbb{R}$ be respectively the set of all non-negative integers and the set of all real numbers. An m-dimensional *hypercomplex* number in $\mathbb{A}_m$ ($m \in \mathbb{N} \setminus \{0\}$) is defined as [6]

$$a := a_1\mathbf{i}_1 + a_2\mathbf{i}_2 + \cdots + a_m\mathbf{i}_m \in \mathbb{A}_m,\ a_1, \ldots, a_m \in \mathbb{R} \quad (1)$$

with imaginary units $\mathbf{i}_1, \ldots, \mathbf{i}_m$, where $\mathbf{i}_1 = 1$ represents the vector identity element. Any hypercomplex number is expressed uniquely in the form of (1). For $a \in \mathbb{A}_m$ the coefficient a_ℓ ($\ell = 1, \ldots, m$) of each imaginary unit $\mathbf{i}_\ell \in \mathbb{A}_m$ is represented as $a_\ell = \Im_\ell(a)$. A *multiplication table* defines the products of any imaginary unit with each other or with itself (e.g., $\mathbf{i}_1^2 = 1, \mathbf{i}_2^2 = -1$ and $\mathbf{i}_1\mathbf{i}_2 = \mathbf{i}_2\mathbf{i}_1 = \mathbf{i}_2$ for $\mathbb{A}_2 (=: \mathbb{C})$). The *addition* and the *subtraction* of two hypercomplex numbers are defined as component-wise operations:

$$a \pm b := (a_1 \pm b_1)\mathbf{i}_1 + (a_2 \pm b_2)\mathbf{i}_2 + \cdots + (a_m \pm b_m)\mathbf{i}_m$$

for $a, b \in \mathbb{A}_m$, where $b := b_1\mathbf{i}_1 + b_2\mathbf{i}_2 + \cdots + b_m\mathbf{i}_m$, $b_1, \ldots, b_m \in \mathbb{R}$. From the unique expression of (1), the *multiplication* of two hypercomplex numbers

$$ab = (a_1\mathbf{i}_1 + a_2\mathbf{i}_2 + \cdots + a_m\mathbf{i}_m)(b_1\mathbf{i}_1 + b_2\mathbf{i}_2 + \cdots + b_m\mathbf{i}_m) := \sum_{k=1}^{m}\sum_{\ell=1}^{m} a_k b_\ell \mathbf{i}_k\mathbf{i}_\ell \in \mathbb{A}_m$$

is determined uniquely according to the multiplication table. The *conjugate* of hypercomplex number a is defined as

$$a^* := a_1\mathbf{i}_1 - a_2\mathbf{i}_2 - \cdots - a_m\mathbf{i}_m. \quad (2)$$

In this paper, we consider the hypercomplex number systems which are constructed recursively by the *Cayley-Dickson construction* (*C-D construction*) [6]. The C-D construction is a standard method for extending a number system. This method has been used in extending $\mathbb{R}$ to $\mathbb{C}$, $\mathbb{C}$ to $\mathbb{H}$ and $\mathbb{H}$ to $\mathbb{O}$. By using the C-D construction, an m-dimensional hypercomplex number $\mathbb{A}_m$ is extended to $\mathbb{A}_{2m}$ [6], [7] as

$$z := x + y\mathbf{i}_{m+1} \in \mathbb{A}_{2m}, \quad x, y \in \mathbb{A}_m,$$

where $\mathbf{i}_{m+1} \notin \mathbb{A}_m$ is a newly introduced imaginary unit for doubling the dimension of $\mathbb{A}_m$ and satisfies $\mathbf{i}_{m+1}^2 = -1$, $\mathbf{i}_1\mathbf{i}_{m+1} = \mathbf{i}_{m+1}\mathbf{i}_1 = \mathbf{i}_{m+1}$ and $\mathbf{i}_v\mathbf{i}_{m+1} = -\mathbf{i}_{m+1}\mathbf{i}_v =: \mathbf{i}_{m+v}$ for all $v = 2, \ldots, m$. For example, the real number system ($\mathbb{A}_1 :=$) $\mathbb{R}$ is extended into complex number system $\mathbb{C}$ ($= \mathbb{A}_2$) by the C-D construction. Note that the value of m is restricted to the form of 2^n ($n \in \mathbb{N}$). The hypercomplex number systems constructed inductively from the real number by the C-D construction are called *Cayley-Dickson number system* (*C-D number system*). The imaginary units appeared in the C-D number systems have many properties such as $\mathbf{i}_\alpha^2 = -1$ and $\mathbf{i}_\alpha\mathbf{i}_\beta = -\mathbf{i}_\beta\mathbf{i}_\alpha (\alpha \neq \beta)$ for all $\alpha, \beta \in \{2, \ldots, m\}$. These properties ensures $aa^* = \sum_{\ell=1}^m a_\ell^2 \geq 0$ for any $a \in \mathbb{A}_m$ in (1) and $a^* \in \mathbb{A}_m$ in (2) and enable us to define the absolute value of C-D number a as $|a| := \sqrt{aa^*}$ (see, e.g., [11]).

Example 1. 1) A representative example of hypercomplex number is the *quaternion* $\mathbb{H}$. A quaternion number is a 4-dimensional hypercomplex number which is defined as

$$q = q_1 + q_2\imath + q_3\jmath + q_4\kappa \in \mathbb{H}, \quad q_1, q_2, q_3, q_4 \in \mathbb{R}$$

with the multiplication table:

$$\imath\jmath = -\jmath\imath = \kappa,\ \jmath\kappa = -\kappa\jmath = \imath,\ \kappa\imath = -\imath\kappa = \jmath, \quad \imath^2 = \jmath^2 = \kappa^2 = -1 \quad (3)$$

by letting $m = 4$, $\mathbf{i}_1 = 1$, $\mathbf{i}_2 = \imath$, $\mathbf{i}_3 = \jmath$ and $\mathbf{i}_4 = \kappa$. From (3), quaternions are not *commutative*, i.e., $pq = qp$ or $pq \neq qp$ for $p, q \in \mathbb{H}$ in general.

2) The octonion $\mathbb{O}$ can be constructed from the quaternion $\mathbb{H}$ by the C-D construction. Note that the multiplication in $\mathbb{O}$ is neither commutative nor *associative*, i.e., neither $pq = qp$ nor $(pq)r = p(qr)$ for $p, q, r \in \mathbb{O}$ holds in general [8]. For the octonion multiplication table, see, e.g., [8].

The C-D number system can be seen as an algebraically natural higher dimensional generalization of our familiar fields, i.e., $\mathbb{R}$ and $\mathbb{C}$.

We also define $\mathbb{A}_m^N := \{[x_1, \ldots, x_N]^\top | x_i \in \mathbb{A}_m (i = 1, \ldots, N)\}$ for $\forall N \in \mathbb{N} \setminus \{0\}$, where $(\cdot)^\top$ stands for the transpose. Define $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathbb{A}_m^N} := \boldsymbol{x}^{\mathsf{H}}\boldsymbol{y} := \sum_{i=1}^N x_i^* y_i \in \mathbb{A}_m, \forall \boldsymbol{x} := [x_1, \ldots, x_N]^\top, \forall \boldsymbol{y} := [y_1, \ldots, y_N]^\top \in \mathbb{A}_m^N$ and $\|\boldsymbol{x}\|_{\mathbb{A}_m^N} := \langle \boldsymbol{x}, \boldsymbol{x} \rangle_{\mathbb{A}_m^N}^{1/2}$, $\forall \boldsymbol{x} \in \mathbb{A}_m^N$, where $(\cdot)^{\mathsf{H}}$ denotes the *Hermitian transpose* of vectors or matrices (e.g., $\boldsymbol{x}^{\mathsf{H}} := [x_1^*, \ldots, x_N^*]$). We also define the *addition* of two hypercomplex vectors $\boldsymbol{x} + \boldsymbol{y} := [x_1 + y_1, \cdots, x_N + y_N]^\top \in \mathbb{A}_m^N$ for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{A}_m^N$. Let $\mathcal{S} := \mathbb{R}$, $\mathcal{S} := \mathbb{C}$ or $\mathcal{S} := \mathbb{A}_m$ ($m \geq 4$), and call the element of $\mathcal{S}$ *scalar*. If we define the *left scalar multiplication* as $\alpha\boldsymbol{x} := [\alpha x_1, \ldots, \alpha x_N]^\top \in \mathbb{A}_m^N$ for $\alpha \in \mathcal{S}$ and $\boldsymbol{x} \in \mathbb{A}_m^N$, we have $\alpha\boldsymbol{x} + \beta\boldsymbol{y} \in \mathbb{A}_m^N$, $\forall\alpha, \beta \in \mathcal{S}, \forall\boldsymbol{x}, \boldsymbol{y} \in \mathbb{A}_m^N$. We can also define the *right scalar multiplication* $\boldsymbol{x}\alpha \in \mathbb{A}_m^N$ in a similar way. For the case of $\mathcal{S} = \mathbb{R}$, the set of all $\mathbb{A}_m$ matrices can be regarded as a vector space over $\mathbb{R}$.

B. Algebraic Translations

In this section, we introduce algebraic translation of C-D vectors and matrices proposed in [11]. A trivial correspondence (mapping) of hypercomplex vectors or matrices to real ones is

$$\widehat{(\cdot)} : \mathbb{A}_m^{M\times N} \to \mathbb{R}^{mM\times N} : \boldsymbol{A} \mapsto \widehat{\boldsymbol{A}} := \begin{bmatrix} \boldsymbol{A}_1 \\ \vdots \\ \boldsymbol{A}_m \end{bmatrix}, \qquad (4)$$

where $\boldsymbol{A} = \boldsymbol{A}_1\mathbf{i}_1 + \cdots + \boldsymbol{A}_m\mathbf{i}_m \in \mathbb{A}_m^{M\times N}$ and $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m \in \mathbb{R}^{M\times N}$. This correspondence is just concatenating a real and all imaginary parts in the hypercomplex matrices. Obviously, this mapping is invertible and thus we can define

$$\smile{(\cdot)} : \mathbb{R}^{mM\times N} \to \mathbb{A}_m^{M\times N} : \widehat{\boldsymbol{A}} \mapsto \boldsymbol{A}.$$

Only in terms of the mappings $\widehat{(\cdot)}$ and $\breve{(\cdot)}$, it is hard to obtain the correspondence of matrix-vector product $\boldsymbol{A}\boldsymbol{x}$, so we also introduce the following non-trivial mapping:

$$\widetilde{(\cdot)} : \mathbb{A}_m^{M\times N} \to \mathbb{R}^{mM\times mN} : \boldsymbol{A} \mapsto \widetilde{\boldsymbol{A}} := \left[\boldsymbol{L}_M^{(1)\top}\widehat{\boldsymbol{A}}, \ldots, \boldsymbol{L}_M^{(m)\top}\widehat{\boldsymbol{A}}\right], \qquad (5)$$

where the matrix $\boldsymbol{L}_M^{(\ell)} \in \mathbb{R}^{mM\times mM}$ $(\ell = 1, \ldots, m)$ is defined for the m-dimensional hypercomplex number $\mathbb{A}_m$ as

$$\boldsymbol{L}_M^{(\ell)} = \begin{bmatrix} \delta_{1,1}^{(\ell)}\boldsymbol{I}_M & \delta_{1,2}^{(\ell)}\boldsymbol{I}_M & \cdots & \delta_{1,m}^{(\ell)}\boldsymbol{I}_M \\ -\delta_{2,1}^{(\ell)}\boldsymbol{I}_M & -\delta_{2,2}^{(\ell)}\boldsymbol{I}_M & \cdots & -\delta_{2,m}^{(\ell)}\boldsymbol{I}_M \\ \vdots & \vdots & \ddots & \vdots \\ -\delta_{m,1}^{(\ell)}\boldsymbol{I}_M & -\delta_{m,2}^{(\ell)}\boldsymbol{I}_M & \cdots & -\delta_{m,m}^{(\ell)}\boldsymbol{I}_M \end{bmatrix}, \qquad (6)$$

with the M-dimensional identity matrix $\boldsymbol{I}_M$ and

$$\delta_{\alpha,\beta}^{(\gamma)} := \begin{cases} 1 & (\text{if } \mathbf{i}_\alpha\mathbf{i}_\beta = \mathbf{i}_\gamma), \\ -1 & (\text{if } \mathbf{i}_\alpha\mathbf{i}_\beta = -\mathbf{i}_\gamma), \\ 0 & (\text{otherwise}). \end{cases} \qquad (7)$$

By (5), the degree of freedom of $\widetilde{\boldsymbol{A}}$ is at most that of $\widehat{\boldsymbol{A}} \in \mathbb{R}^{mM\times N}$. More precisely, $\widetilde{(\cdot)}$ is a mapping onto

$$\mathfrak{S}_{\mathbb{A}_m}(M, N) := \{\widetilde{\boldsymbol{A}} \in \mathbb{R}^{mM\times mN} | \boldsymbol{A} \in \mathbb{A}_m^{M\times N}\} = \left\{ \left[\boldsymbol{L}_M^{(1)\top}\boldsymbol{B}, \ldots, \boldsymbol{L}_M^{(m)\top}\boldsymbol{B}\right] \middle| \boldsymbol{B} \in \mathbb{R}^{mM\times N} \right\}.$$

The set $\mathfrak{S}_{\mathbb{A}_m}(M, N)$ will play important roles in the context of optimization such as in Section IV. Similar to the trivial mapping, $\widetilde{(\cdot)}$ is also invertible and thus we can define

$$\underset{\smile}{(\cdot)} : \mathfrak{S}_{\mathbb{A}_m}(M, N) \to \mathbb{A}_m^{M\times N} : \widetilde{\boldsymbol{A}} \mapsto \boldsymbol{A}.$$

These mappings have the following useful properties:

Fact 1 (Algebraic correspondence between real and C-D vectors and matrices [11])**.** *For all $\boldsymbol{A}, \boldsymbol{A}' \in \mathbb{A}_m^{M\times N}$, $\boldsymbol{B} \in \mathbb{A}_m^{N\times L}$ and $\boldsymbol{x} \in \mathbb{A}_m^N$,*

1) $\widehat{(\boldsymbol{A} + \boldsymbol{A}')} = \widehat{\boldsymbol{A}} + \widehat{\boldsymbol{A}'}$, $\widehat{(\alpha\boldsymbol{A})} = \alpha\widehat{\boldsymbol{A}}$, $\widetilde{(\boldsymbol{A} + \boldsymbol{A}')} = \widetilde{\boldsymbol{A}} + \widetilde{\boldsymbol{A}'}$, $\widetilde{(\alpha\boldsymbol{A})} = \alpha\widetilde{\boldsymbol{A}}$ *for all* $\alpha \in \mathbb{R}$,
2) $\widetilde{(\boldsymbol{A}^{\mathsf{H}})} = \widetilde{\boldsymbol{A}}^\top$,
3) $\|\boldsymbol{x}\|_{\mathbb{A}_m^N} = \|\widehat{\boldsymbol{x}}\|_{\mathbb{R}^{mN}}$,
4) $\widehat{(\boldsymbol{A}\boldsymbol{B})} = \widetilde{\boldsymbol{A}}\widehat{\boldsymbol{B}}$ *and* $\widehat{(\boldsymbol{A}\boldsymbol{x})} = \widetilde{\boldsymbol{A}}\widehat{\boldsymbol{x}}$,
5) $\widetilde{(\boldsymbol{A}\boldsymbol{B})} = \widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{B}}$ *if* $m \leq 4$*, i.e., if* $\mathbb{A}_m = \mathbb{R}$ *or* $\mathbb{C}$ *or* $\mathbb{H}$.

Example 2. For a quaternion matrix $\boldsymbol{A} := \boldsymbol{A}_1 + \boldsymbol{A}_2\imath + \boldsymbol{A}_3\jmath + \boldsymbol{A}_4\kappa \in \mathbb{H}^{M\times N}$, $\widehat{\boldsymbol{A}} \in \mathbb{R}^{4M\times N}$ and $\widetilde{\boldsymbol{A}} \in \mathbb{R}^{4M\times 4N}$ are given as

$$\widehat{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \boldsymbol{A}_3 \\ \boldsymbol{A}_4 \end{bmatrix}, \widetilde{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A}_1 & -\boldsymbol{A}_2 & -\boldsymbol{A}_3 & -\boldsymbol{A}_4 \\ \boldsymbol{A}_2 & \boldsymbol{A}_1 & -\boldsymbol{A}_4 & \boldsymbol{A}_3 \\ \boldsymbol{A}_3 & \boldsymbol{A}_4 & \boldsymbol{A}_1 & -\boldsymbol{A}_2 \\ \boldsymbol{A}_4 & -\boldsymbol{A}_3 & \boldsymbol{A}_2 & \boldsymbol{A}_1 \end{bmatrix}.$$

Remark 1. Obviously, $\mathfrak{S}_{\mathbb{A}_m}(M, N)$ is an mMN-dimensional real vector space and the non-trivial mapping $\widetilde{(\cdot)}$ is guaranteed to be an isomorphism between $\mathbb{A}_m^{M\times N}$ and $\mathfrak{S}_{\mathbb{A}_m}(M, N)$ regarding $\mathbb{A}_m^{M\times N}$ as a vector space over $\mathbb{R}$ (see Section II-A).

III. Hypercomplex Tensor Singular Value Decomposition

A. Hypercomplex Extension of Tensor Basics

In this section, we introduce the fundamental tensor notations in hypercomplex domain.

A tensor is a generalization of a matrix to higher dimension. In this paper, we denote it by a calligraphic letter, e.g., $\mathcal{X} \in \mathbb{A}_m^{N_1\times\cdots\times N_n}$. The *order* (also called *ways* or *modes*) n of tensor is the number of dimensions. A matrix is denoted by a boldface capital letter, e.g., $\boldsymbol{X} \in \mathbb{A}_m^{M\times N}$; a vector is denoted by a lower case bold letter, e.g., $\boldsymbol{x} \in \mathbb{A}_m^N$, and a scalar is denoted by lower case letter, e.g., $x \in \mathbb{A}_m$. For a 3-way tensor $\mathcal{X} \in \mathbb{A}_m^{N_1\times N_2\times N_3}$, we denote its (i, j, k)-th entry as $\mathcal{X}_{ijk}$ or x_{ijk}, and we use MATLAB-like notations $\boldsymbol{X}_{i::} \in \mathbb{A}_m^{N_2\times N_3}$, $\boldsymbol{X}_{:j:} \in \mathbb{A}_m^{N_1\times N_3}$ and $\boldsymbol{X}_{::k} \in \mathbb{A}_m^{N_1\times N_2}$ to denote respectively the i-th horizontal, j-th lateral, and k-th frontal slices. Especially, we denote the frontal slice $\boldsymbol{X}_{::k}$ simply as $\boldsymbol{X}^{(k)}$ or $(\mathcal{X})^{(k)}$. We also define the *tube (fiber)* by fixing the first two indices and denote it by, e.g., $\boldsymbol{x}_{ij:}$ following [14].

Especially for real 3-way tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{N_1\times N_2\times N_3}$, the inner product of two same sized tensors is defined as $\langle\mathcal{X}, \mathcal{Y}\rangle_{\mathbb{R}^{N_1\times N_2\times N_3}} := \sum_{k=1}^{N_3} \mathrm{tr}(\boldsymbol{X}^{(k)\top}\boldsymbol{Y}^{(k)}) = \sum_{i,j,k=1}^{N_1,N_2,N_3} x_{ijk}y_{ijk}$ with its induced norm $\|\mathcal{X}\|_{\mathbb{R}^{N_1\times N_2\times N_3}}$. Also, we denote $\overline{\mathcal{X}}$ as the result of discrete Fourier transform of $\mathcal{X} \in \mathbb{C}^{N_1\times N_2\times N_3}$ along the third dimension using the MATLAB fast Fourier transform (FFT) function, i.e., $\overline{\mathcal{X}} := \mathrm{fft}(\mathcal{X}, [], 3)$. In the same fashion, $\mathcal{X}$ can be recovered from $\overline{\mathcal{X}}$ by the inverse FFT, i.e., $\mathcal{X} =: \mathrm{ifft}(\overline{\mathcal{X}}, [], 3)$.

B. Hypercomplex Extension of 3-way Tensor Algebra

We can also formally extend tensor algebra related to tensor product (t-product [19]). For $\mathcal{X} \in \mathbb{A}_m^{N_1\times N_2\times N_3}$, its *block* *circulant matrix* $\mathrm{bcirc}(\mathcal{X}) \in \mathbb{A}_m^{N_1N_3\times N_2N_3}$ is defined as:

$$\mathrm{bcirc}(\mathcal{X}) := \begin{bmatrix} \boldsymbol{X}^{(1)} & \boldsymbol{X}^{(N_3)} & \cdots & \boldsymbol{X}^{(2)} \\ \boldsymbol{X}^{(2)} & \boldsymbol{X}^{(1)} & \cdots & \boldsymbol{X}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{X}^{(N_3)} & \boldsymbol{X}^{(N_3-1)} & \cdots & \boldsymbol{X}^{(1)} \end{bmatrix}.$$

We also define a kind of matricization operator and its inverse:

$$\mathrm{unfold}(\mathcal{X}) := \begin{bmatrix} \boldsymbol{X}^{(1)} \\ \boldsymbol{X}^{(2)} \\ \vdots \\ \boldsymbol{X}^{(N_3)} \end{bmatrix} \in \mathbb{A}_m^{N_1N_3\times N_2}, \ \mathrm{fold}(\mathrm{unfold}(\mathcal{X})) := \mathcal{X}.$$

Note that $\mathrm{unfold}(\mathcal{X})$ is not equivalent to *mode-n* unfoldings of $\mathcal{X}$, another kind of well-known matricization of tensors introduced in e.g., [14]. We then extend the *t-product* [19] between two 3-way tensors, and conjugate transpose to hypercomplex domain.

Definition 1 (t-product between two hypercomplex 3-way tensors)**.** Let $\mathcal{X} \in \mathbb{A}_m^{N_1\times N_2\times N_3}$ and $\mathcal{Y} \in \mathbb{A}_m^{N_2\times L\times N_3}$. The *t-product* $\mathcal{X} * \mathcal{Y} \in \mathbb{A}_m^{N_1\times L\times N_3}$ is defined as:

$$\mathcal{X} * \mathcal{Y} := \mathrm{fold}(\mathrm{bcirc}(\mathcal{X}) \cdot \mathrm{unfold}(\mathcal{Y})).$$

Definition 2 (Conjugate transpose of hypercomplex 3-way tensors)**.** The *conjugate transpose* of a tenspr $\mathcal{X} \in \mathbb{A}_m^{N_1\times N_2\times N_3}$ is the hypercomplex tensor $\mathcal{X}^{\mathsf{H}} \in \mathbb{A}_m^{N_2\times N_1\times N_3}$ obtained by conjugate transposing each frontal slice and then reversing the order of the conjugate transposed frontal slices from 2-nd through N_3-th ones.

Since tensor algebra related to t-product and t-SVD is based on multiplications of frontal slices, we define real translation of hypercomplex 3-way tensors accordingly as follows:

Definition 3 (Real translation of hypercomplex 3-way tensors)**.** For a hypercomplex 3-way tensor $\mathcal{A} \in \mathbb{A}_m^{N_1\times N_2\times N_3}$, we define its real translations $\widehat{\mathcal{A}} \in \mathbb{R}^{mN_1\times N_2\times N_3}$ and $\widetilde{\mathcal{A}} \in \mathbb{R}^{mN_1\times mN_2\times N_3}$ by applying algebraic real translations of hypercomplex matrices for each frontal slice, i.e.,

$$\widehat{\mathcal{A}} := \mathrm{fold}\left(\begin{bmatrix} \widehat{\boldsymbol{A}^{(1)}} \\ \widehat{\boldsymbol{A}^{(2)}} \\ \vdots \\ \widehat{\boldsymbol{A}^{(N_3)}} \end{bmatrix}\right) \text{ and } \widetilde{\mathcal{A}} := \mathrm{fold}\left(\begin{bmatrix} \widetilde{\boldsymbol{A}^{(1)}} \\ \widetilde{\boldsymbol{A}^{(2)}} \\ \vdots \\ \widetilde{\boldsymbol{A}^{(N_3)}} \end{bmatrix}\right).$$

We can also define the inverse of these translations:

$$\widecheck{(\cdot)} : \mathbb{R}^{mN_1\times N_2\times N_3} \to \mathbb{A}_m^{N_1\times N_2\times N_3} : \widehat{\mathcal{A}} \mapsto \mathcal{A},$$
$$\underset{\smile}{(\cdot)} : \mathfrak{S}_{\mathbb{A}_m}(N_1,N_2,N_3) \to \mathbb{A}_m^{N_1\times N_2\times N_3} : \widetilde{\mathcal{A}} \mapsto \mathcal{A},$$

where

$$\mathfrak{S}_{\mathbb{A}_m}(N_1,N_2,N_3) := \{\widetilde{\mathcal{A}} \in \mathbb{R}^{mN_1\times mN_2\times N_3} | \mathcal{A} \in \mathbb{A}_m^{N_1\times N_2\times N_3}\} \subset \mathbb{R}^{mN_1\times mN_2\times N_3}.$$

If we define the real translation in this way, many algebraic properties like Fact 1 are inherited in 3-way tensor cases. We can easily verify that the following facts hold.

Fact 2 (Algebraic properties of real translations of hypercomplex 3-way tensors)**.** *For any hypercomplex tensors $\mathcal{A}, \mathcal{B} \in \mathbb{A}_m^{N_1\times N_2\times N_3}$, the following proerties hold:*

1) $\widehat{(\mathcal{A}+\mathcal{B})} = \widehat{\mathcal{A}} + \widehat{\mathcal{B}}$, $\widehat{(\alpha\mathcal{A})} = \alpha\widehat{\mathcal{A}}$,
$\widetilde{(\mathcal{A}+\mathcal{B})} = \widetilde{\mathcal{A}} + \widetilde{\mathcal{B}}$, $\widetilde{(\alpha\mathcal{A})} = \alpha\widetilde{\mathcal{A}}$, $\forall\alpha \in \mathbb{R}$,

2) $\widetilde{(\mathcal{A}^{\mathsf{H}})} = \widetilde{\mathcal{A}}^{\top}$.

Moreover, we can show that the algebraic properties of matrix multiplication such as Fact 1-4) and Fact 1-5) are also hold for t-product of hypercomplex 3-way tensors.

Theorem 1 (t-product of real translated hypercomplex tensors)**.** *For the real translations of hypercomplex tensors and their t-product, the following relations hold for any hypercomplex tensors $\mathcal{A} \in \mathbb{A}_m^{N_1\times N_2\times N_3}$ and $\mathcal{B} \in \mathbb{A}_m^{N_2\times L\times N_3}$*

1) $\widehat{(\mathcal{A}*\mathcal{B})} = \widetilde{\mathcal{A}} * \widehat{\mathcal{B}}$,

2) $\widetilde{(\mathcal{A}*\mathcal{B})} = \widetilde{\mathcal{A}} * \widetilde{\mathcal{B}}$ *if* $\mathbb{A}_m = \mathbb{C}$ *or* $\mathbb{H}$.

Proof: It can be verified with the original definition of t-product in real domain (see [19]), Definition 1, and Fact 1-4) and 1-1). ■

Note that Theorem 1 also gives a concrete computation of t-product between any two C-D (hypercomplex) tensors.

Corollary 1. *The t-product of any two C-D tensors $\mathcal{A} \in \mathbb{A}_m^{N_1\times N_2\times N_3}$ and $\mathcal{B} \in \mathbb{A}_m^{N_2\times L\times N_3}$ can be efficiently calculated with FFT in real domain by*

$$\mathcal{A} * \mathcal{B} = \widecheck{\mathcal{C}}, \ \mathcal{C} := \widetilde{\mathcal{A}} * \widehat{\mathcal{B}}. \tag{8}$$

The second equation in (8) is just the t-product in real domain, so we use the fact that t-product can be computed efficiently with FFT.

Using real translations of hypercomplex tensor, we then extend tensor singular value decomposition (t-SVD) [19] to hypercomplex 3-way tensors.

Theorem 2 (C-D t-SVD)**.** *The real translation of any hypercomplex tensor $\mathcal{A} \in \mathbb{A}_m^{N_1\times N_2\times N_3}$ can be decomposed as*

$$\widetilde{\mathcal{A}} = \mathcal{U} * \mathcal{S} * \mathcal{V}^{\top},$$

where $\mathcal{U} \in \mathbb{R}^{mN_1\times mN_1\times N_3}$, $\mathcal{V} \in \mathbb{R}^{mN_2\times mN_2\times N_3}$ are orthogonal[1] *tensors and $\mathcal{S} \in \mathbb{R}^{mN_1\times mN_2\times N_3}$ is an f-diagonal*[2] *tensor.*

Fig. 1 illustrates the C-D t-SVD factorization. Note that since this decomposition is performed in real domain, so it can be efficiently computed based on the matrix SVD in the Fourier domain similar to real tensor cases. This is based on a key property that the block circulant matrix is diagonalizable:

[1] A real tensor $\mathcal{Q} \in \mathbb{R}^{N\times N\times N_3}$ is *orthogonal* if it satisfies $\mathcal{Q}^{\mathsf{H}} * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^{\mathsf{H}} = \mathcal{I}$, where $\mathcal{I} \in \mathbb{R}^{N\times N\times N_3}$ is the identity tensor, whose first frontal slice is $N \times N$ identity matrix, and whose other frontal slices are all zeros.

[2] A real tensor is called f-diagonal if each of its frontal slices is a diagonal matrix.

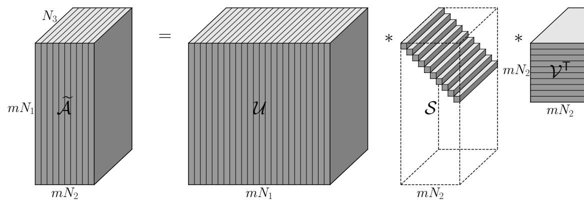

Fig. 1. C-D t-SVD

$$(\boldsymbol{F}_{N_3} \otimes \boldsymbol{I}_{mN_1})\mathrm{bcirc}(\widetilde{\mathcal{A}})(\boldsymbol{F}_{N_3}^{\mathsf{H}} \otimes \boldsymbol{I}_{mN_2}) = \mathrm{diag}\left\{\overline{\widetilde{\boldsymbol{A}^{(1)}}}, \ldots, \overline{\widetilde{\boldsymbol{A}^{(N_3)}}}\right\},$$

where $\boldsymbol{F}_{N_3} \in \mathbb{C}^{N_3 \times N_3}$ denotes the discrete Fourier transform (DFT) matrix and '$\otimes$' is the Kronecker product. We also define the *multi rank*, *tubal rank* and *tensor nuclear norm* of hypercomplex 3-way tensors based on C-D t-SVD.

Definition 4 (C-D tensor multi rank and tubal rank)**.** For a hypercomplex tensor $\mathcal{A} \in \mathbb{A}_m^{N_1 \times N_2 \times N_3}$, we define the *multi rank* as a vector $\boldsymbol{r} \in \mathbb{N}^{N_3}$, whose i-th entry is the rank of the i-th frontal slice of $\overline{\widetilde{\mathcal{A}}} \in \mathbb{C}^{mN_1 \times mN_2 \times N_3}$, i.e., $r_i = \mathrm{rank}\left\{\overline{\widetilde{\boldsymbol{A}^{(i)}}}\right\}$. We also define the *tubal rank* of hypercomplex tensor $\mathcal{A}$, denoted by $\mathrm{rank}^{\mathrm{t}}(\mathcal{A})$ as the number of non-zero singular tubes of $\mathcal{S}$, where $\mathcal{S}$ is from C-D t-SVD of $\widetilde{\mathcal{A}} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$. The tubal rank can be defined also as the largest rank of all frontal slices of $\overline{\widetilde{\mathcal{A}}}$, i.e., $\max_i r_i$.

Definition 5 (C-D tensor nuclear norm)**.** We define the tensor *nuclear norm* of a hypercomplex tensor $\mathcal{A} \in \mathbb{A}_m^{N_1 \times N_2 \times N_3}$, denoted as $\|\mathcal{A}\|_{\mathsf{TNN}}$ as the average of the nuclear norm of all frontal slices of $\overline{\widetilde{\mathcal{A}}}$, i.e., $\|\mathcal{A}\|_{\mathsf{TNN}} := \frac{1}{N_3}\sum_{i=1}^{N_3}\left\|\overline{\widetilde{\boldsymbol{A}^{(i)}}}\right\|_*$, where the nuclear norm of a real matrix $\|\boldsymbol{X}\|_*$ is defined as the sum of all positive singular values of $\boldsymbol{X} \in \mathbb{R}^{M \times N}$.

Lemma 1 (Best p rank approximation of hypercomplex 3-way tensors)**.** *For a C-D tensor $\mathcal{A} \in \mathbb{A}_m^{N_1 \times N_2 \times N_3}$ of tubal rank R, a best p-tubal rank approximation is achieved by truncated p-rank approximation of all frontal slices of $\overline{\widetilde{\mathcal{A}}}$.*

Theorem 3 (Existence of hypercomplex tensor corresponding to the best low tubal rank approximation)**.** *For complex case, i.e., $m = 2$ the followings hold:*

1) *There always exists $\mathcal{A}^\star \in \mathbb{A}_m^{N_1 \times N_2 \times N_3}$ that satisfies $\widetilde{\mathcal{A}^\star} = \mathcal{A}_{mp}$, i.e., $\mathcal{A}_{mp} \in \mathfrak{S}_{\mathbb{A}_m}(N_1, N_2, N_3)$ where $\mathcal{A}_{mp} \in \mathbb{R}^{mN_1 \times mN_2 \times N_3}$ is the best mp tubal rank approximation of $\mathcal{A} \in \mathbb{A}_m^{N_1 \times N_2 \times N_3}$ by Lemma 1.*
2) *$\mathcal{A}_{mp}$ also achieves the best p tubal rank approximation of $\mathcal{A}$ in the sense of the original t-SVD in the complex domain.*

IV. Hypercomplex Tensor Robust Principal Component Analysis

A. Tools for Convex Optimization for Tensors

In this section, we formulate the tensor robust principal component analysis (tensor RPCA) in C-D domain. Since the tubal rank defined in Definition 4 is available for general C-D 3-way tensors, we can formulate it in C-D domain as a problem of decomposing an observation $\mathcal{M}$ into a low rank $\mathcal{L}$ and a sparse $\mathcal{S}$ tensors:

$$\underset{\mathcal{L},\mathcal{S} \in \mathbb{A}_m^{N_1 \times N_2 \times N_3}}{\mathrm{minimize}} \ \mathrm{rank}^{\mathrm{t}}(\mathcal{L}) + \lambda\|\mathcal{S}\|_{0,\mathbb{A}_m}, \ \text{s.t.}\ \mathcal{M} = \mathcal{L} + \mathcal{S},$$

where $\lambda > 0$ and $\|\mathcal{A}\|_{0,\mathbb{A}_m}$ is the number of non-zero entries in $\mathcal{A} \in \mathbb{A}_m^{N_1 \times N_2 \times N_3}$. Similar to matrix cases in [21], we can relax $\|\cdot\|_{0,\mathbb{A}_m}$ to the sum of absolute values of all entries in $\mathcal{A}$

$$\|\mathcal{A}\|_{1,\mathbb{A}_m} := \sum_{i,j,k=1}^{N_1,N_2,N_3} |\mathcal{A}_{ijk}| =: \|\widehat{\mathcal{A}}\|_{1,\mathbb{A}_m}.$$

By relaxing $\mathrm{rank}^{\mathrm{t}}(\cdot)$ by tensor nuclear norm defined in Definition 5, we can relax hypercomplex tensor RPCA to the following convex optimization problem, which we call in this paper *C-D tensor principal component pursuit (C-D tensor PCP)*:

$$\underset{\mathcal{L},\mathcal{S} \in \mathbb{A}_m^{N_1 \times N_2 \times N_3}}{\mathrm{minimize}} \ \|\mathcal{L}\|_{\mathsf{TNN}} + \lambda\|\mathcal{S}\|_{1,\mathbb{A}_m} \ \text{s.t.}\ \mathcal{M} = \mathcal{L} + \mathcal{S}. \quad (9)$$

Since $\|\mathcal{S}\|_{1,\mathbb{A}_m}$ can be calculated in real domain similar to matrix cases, both terms in (9) can be processed in real domain.

Fact 3 (Proximity operator of tensor nuclear norm [23])**.** *From Fact 3, the proximity operator of tensor nuclear norm*

$$\mathrm{prox}_{\gamma\|\cdot\|_{\mathsf{TNN}}}(\mathcal{X}) := \underset{\mathcal{Y} \in \mathbb{R}^{N_1 \times N_2 \times N_3}}{\arg\min} \left\{\|\mathcal{Y}\|_{\mathsf{TNN}} + \frac{1}{2\gamma}\|\mathcal{X} - \mathcal{Y}\|_F^2\right\} =: \mathrm{shrink}^{\mathrm{t}}(\mathcal{X}, \gamma)$$

can be calculated in the Fourier domain as

$$\overline{\left\{\mathrm{shrink}^{\mathrm{t}}(\mathcal{X}, \gamma)\right\}^{(i)}} = \underset{\boldsymbol{Y} \in \mathbb{C}^{N_1 \times N_2}}{\arg\min} \left\{\|\boldsymbol{Y}\|_* + \frac{1}{2\gamma}\left\|\overline{\boldsymbol{X}^{(i)}} - \boldsymbol{Y}\right\|_F^2\right\} = \mathrm{shrink}(\overline{\boldsymbol{X}^{(i)}}, \gamma) \quad (i = 1, \ldots, N_3),$$

where the shrinkage operator of complex matrix $\boldsymbol{X} \in \mathbb{C}^{M \times N}$ is defined as $\mathrm{shrink}(\boldsymbol{X}, \tau) := \boldsymbol{U}\boldsymbol{\Sigma}_\tau\boldsymbol{V}^{\mathsf{H}}$, with the singular value decomposition of a complex matrix of rank r, $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{H}}$ and the shrunk diagonal matrix $(\boldsymbol{\Sigma}_\tau)_{ij} := (\boldsymbol{\Sigma}_{ij} - \tau)_+$ $((\cdot)_+ := \max(0, \cdot))$.

Fact 4 (Proximity operator of tensor ℓ_1-norm)**.** *Similar to matrix cases in [21], the proximity operator of hypercomplex tensor ℓ_1-norm can be calculated group-wise as*

$$\left[\mathrm{prox}_{\gamma\|\cdot\|_{1,\mathbb{A}_m}}(\widehat{\mathcal{A}})\right]_{ijk} = \frac{\widehat{\mathcal{A}_{ijk}}}{\left\|\widehat{\mathcal{A}_{ijk}}\right\|_2}\left[\left\|\widehat{\mathcal{A}_{ijk}}\right\|_2 - \gamma\right]_+, =: \mathrm{ST}(\mathcal{A}, \gamma). \quad (10)$$

B. Hypercomplex Tensor Principal Component Pursuit via Convex Optimization

In this section, we derive a new algorithm based on the Douglas-Rachford splitting technique [22] to solve the C-D tensor PCP (9) efficiently. Denote the 2-fold Cartesian product of the spaces of real tensors by $\mathcal{H}_0 := \mathbb{R}^{mN_1\times mN_2\times N_3} \times \mathbb{R}^{mN_1\times N_2\times N_3}$. By defining the inner product $\langle \mathscr{X}, \mathscr{Y}\rangle_{\mathcal{H}_0} := \frac{1}{2}\langle \mathcal{X}_1, \mathcal{Y}_1\rangle_{\mathbb{R}^{mN_1\times mN_2\times N_3}} + \frac{1}{2}\langle \mathcal{X}_2, \mathcal{Y}_2\rangle_{\mathbb{R}^{mN_1\times N_2\times N_3}}$, where $\mathscr{X} := [\mathcal{X}_1, \mathcal{X}_2] \in \mathcal{H}_0$ and $\mathscr{Y} := [\mathcal{Y}_1, \mathcal{Y}_2] \in \mathcal{H}_0$, ($\mathcal{X}_1, \mathcal{Y}_1 \in \mathbb{R}^{mN_1\times mN_2\times N_3}$, $\mathcal{X}_2, \mathcal{Y}_2 \in \mathbb{R}^{mN_1\times N_2\times N_3}$) and induced norm $\|\mathscr{X}\|_{\mathcal{H}_0} := \sqrt{\langle \mathscr{X}, \mathscr{X}\rangle_{\mathcal{H}_0}}$, $\mathcal{H}_0$ becomes a real Hilbert space. First, we reformulate the problem (9) as an unconstrained the sum of two functions as follows:

$$\underset{\mathscr{Z}\in\mathcal{H}_0}{\text{minimize}} \quad f(\mathscr{Z}) + g(\mathscr{Z}), \tag{11}$$

where

$$\begin{cases} f(\mathscr{Z}) := f_1(\mathcal{Z}_1) + f_2(\mathcal{Z}_2) = \|\mathcal{Z}_1\|_{\mathsf{TNN}} + \|\mathcal{Z}_2\|_{1,\mathbb{A}_m}, \\ g(\mathscr{Z}) := \iota_{D_1}(\mathscr{Z}) = \begin{cases} 0 & (\text{if } \mathscr{Z} \in D_1), \\ +\infty & (\text{otherwise}), \end{cases} \end{cases}$$

$$\begin{aligned} &\mathscr{Z} := [\mathcal{Z}_1, \mathcal{Z}_2] \in \mathcal{H}_0, \\ &D_1 := \left\{ [\mathcal{Z}_1, \mathcal{Z}_2] \in D_2 \,\middle|\, \mathcal{M} = \underset{\sim}{\mathcal{Z}}_1 + \check{\mathcal{Z}}_2 \right\} \subset D_2, \\ &D_2 := \mathfrak{S} \times \mathbb{R}^{mN_1\times N_2\times N_3} \subset \mathcal{H}_0, \\ &\mathfrak{S} := \mathfrak{S}_{\mathbb{A}_m}(N_1, N_2, N_3) \subset \mathbb{R}^{mN_1\times mN_2\times N_3}. \end{aligned}$$

Note that the subspace D_1 represents the constraint that the observation $\mathcal{M}$ is from the sum of low rank and sparse tensors. This requests that both $\mathcal{Z}_1$ belong to $\mathfrak{S}$, so we need the subspace D_2.

Apparently this reformulation (11) is equivalent to (9), so all we need is to identify the concrete calculation of the proximity operators of f and g. In the same way as [24], the proximity operator of f is given by

$$\text{prox}_{\gamma f}(\mathscr{X}) = \left[\text{prox}_{2\gamma f}(\mathcal{X}_1), \text{prox}_{2\gamma f}(\mathcal{X}_2)\right].$$

From Fact 3, the proximity operator of f_1, i.e., the tensor nuclear norm with index 2γ is given by

$$\text{prox}_{2\gamma f_1}(\mathcal{X}_1) = \text{shrink}^{\text{t}}(\mathcal{X}_1, 2\gamma).$$

The concrete procedure of $\text{shrink}^{\text{t}}(\mathcal{X}, \tau)$ is described in Algorithm 1.

Algorithm 1: $\text{shrink}^{\text{t}}(\mathcal{X}, \tau)$

Input : $\mathcal{X} \in \mathbb{R}^{N_1\times N_2\times N_3}$, $\tau > 0$
Output: $\mathcal{Z} := \text{shrink}^{\text{t}}(\mathcal{X}, \tau)$
1 $\overline{\mathcal{X}} \leftarrow \text{fft}(\mathcal{X}, [], 3)$;
2 **for** $i = 1, 2, \ldots, N_3$ **do**
3 $\quad \overline{\boldsymbol{Z}^{(i)}} \leftarrow \text{shrink}(\overline{\boldsymbol{X}^{(i)}}, \tau)$;
4 $\mathcal{Z} \leftarrow \text{ifft}(\overline{\mathcal{Z}}, [], 3)$

By Fact 4, the proximity operator of f_2, reduces to the group-wise soft-thresholding (10):

$$\text{prox}_{2\gamma f_2}(\mathcal{X}_2) = \text{ST}(\breve{\mathcal{X}}_2, 2\gamma\lambda). \tag{12}$$

For the function g, the proximity operator of the indicator function ι_{D_1} is the orthogonal projection P_{D_1} onto the subspace D_1, i.e.,

$$\text{prox}_{\gamma g}(\mathscr{X}) = P_{D_1}(\mathscr{X}) := \underset{\mathscr{Y}\in D_1}{\arg\min}\, \|\mathscr{X} - \mathscr{Y}\|_{\mathcal{H}_0}.$$

Since $D_1 \subset D_2 \subset \mathcal{H}_0$, we have by [25, 5.14, Reduction principle]

$$P_{D_1}(\mathscr{X}) = P_{D_1}|D_2 \circ P_{D_2}(\mathscr{X}).$$

Note that '$|D_2$' in $P_{D_1}|D_2$ stands for the restriction of the domain to the subspace D_2 and '$\circ$' stands for the composition of mappings. The orthogonal projection $P_{D_2} : \mathcal{H}_0 \to D_2$ and $P_{D_1}|D_2 : D_2 \to D_1$ respectively can be calculated as

$$P_{D_2}(\mathscr{X}) = [P_{\mathfrak{S}}(\mathcal{X}_1), \mathcal{X}_2]$$

and

$$P_{D_1}|D_2(\mathscr{X}) = \frac{1}{2}\left[\widetilde{\mathcal{M}} + \mathcal{X}_1 - \widetilde{\mathcal{X}}_2^{\star}, \widehat{\mathcal{M}} - \widehat{\mathcal{X}}_1^{\star} + \mathcal{X}_2\right],$$

where $\mathcal{X}_1^{\star} := \underset{\sim}{\mathcal{X}}_1 \in \mathbb{A}_m^{N_1\times N_2\times N_3}$ and $\mathcal{X}_2^{\star} := \check{\mathcal{X}}_2 \in \mathbb{A}_m^{M\times N}$. For $P_{\mathfrak{S}}(\mathcal{X}_1)$, let $\mathcal{E}_{pqr\ell} := \mathcal{E}_{pqr}\mathbf{i}_\ell \in \mathbb{A}_m^{N_1\times N_2\times N_3}$ ($\ell = 1, \ldots, m$), where $\mathcal{E}_{pqr} \in \mathbb{R}^{N_1\times N_2\times N_3}$ is the tensor only whose (p, q, r)-th entry ($p = 1, \ldots, N_1$, $q = 1, \ldots, N_2$, $r = 1, \ldots, N_3$) is 1 and all other entries are 0. Then, we can easily verify that

$$\langle \widetilde{\mathcal{E}}_{pqr\ell}, \widetilde{\mathcal{E}}_{p'q'r'\ell'}\rangle_{\mathbb{R}^{mN_1\times mN_2\times N_3}} = \begin{cases} m & (\text{if } (p,q,r,\ell) = (p',q',r',\ell')), \\ 0 & (\text{otherwise}). \end{cases}$$

and therefore, $\{\frac{1}{\sqrt{m}}\widetilde{\mathcal{E}}_{pqr\ell}\}_{p=1,q=1,r=1,\ell=1}^{N_1,N_2,N_3,m}$ is an orthonormal basis of $\mathfrak{S}$ and thus $P_{\mathfrak{S}}(\mathcal{X}_1)$ can be easily calculated as:

$$P_{\mathfrak{S}}(\mathcal{X}_1) = \frac{1}{m}\sum_{p=1}^{N_1}\sum_{q=1}^{N_2}\sum_{r=1}^{N_3}\sum_{\ell=1}^{m}\langle \mathcal{X}_1, \widetilde{\mathcal{E}}_{pqr\ell}\rangle_{\mathbb{R}^{mN_1\times mN_2\times N_3}}\widetilde{\mathcal{E}}_{pqr\ell}.$$

Now, we can calculate

$$\begin{aligned} \text{prox}_{\gamma g}(\mathscr{X}) &= P_{D_1}|D_2 \circ P_{D_2}(\mathscr{X}) = P_{D_1}|D_2\,[P_{\mathfrak{S}}(\mathcal{X}_1), \mathcal{X}_2] \\ &= \frac{1}{2}\left[\widetilde{\mathcal{M}} + P_{\mathfrak{S}}(\mathcal{X}_1) - \widetilde{\mathcal{X}}_2^{\star}, \widehat{\mathcal{M}} - \widehat{\mathcal{X}}_1^{\star\star} + \mathcal{X}_2\right], \end{aligned}$$

where $\mathcal{X}_1^{\star\star} := \underset{\sim}{P_{\mathfrak{S}}(\mathcal{X}_1)} \in \mathbb{A}_m^{N_1\times N_2\times N_3}$. Since all ingredients are identified, we can summarize the proposed hypercomplex principal component pursuit algorithm in Algorithm 2. Here, $(t_k)_{k\geq 0} \subset [0, 2]$ satisfied $\sum_{k\geq 0} t_k(2 - t_k) = +\infty$, $\gamma \in (0, +\infty)$. Note that the shrinkage operator does not keep the special structure of $\widetilde{(\cdot)}$, i.e., $\text{shrink}^{\text{t}}(\widetilde{\mathcal{A}}, 2\gamma) \notin \mathfrak{S}$ in general, so we need the projection onto the structure $P_{\mathfrak{S}}$. However, in complex and quaternion domain, we experimentally observe that it keeps the structure, so it seems that $\mathcal{L}_{(k)} \in \mathfrak{S}$ and $P_{\mathfrak{S}}(\mathcal{L}_{(k)}) = \mathcal{L}_{(k)}$ for all $k \geq 0$, but its strict discussion will be reported elsewhere. Especially if $N_3 = 1$ (i.e., matrix case),

Algorithm 2: $\mathbb{A}_m$-Douglas-Rachford splitting for hypercomplex tensor principal component pursuit ($\mathbb{A}_m$-DRS-TPCP)

Input : $\mathcal{M}$, t_k, γ, λ
Output: Low tubal rank $\mathcal{L}$ and sparse $\mathcal{S}$

Initialize $k \leftarrow 0$, $\mathcal{L}_{(k)} \leftarrow \mathbf{0}$, $\mathcal{S}_{(k)} \leftarrow \mathbf{0}$;
repeat
 $\mathcal{L}^{\star\star} \leftarrow \underset{\sim}{P_{\mathfrak{S}}(\mathcal{L}_{(k)})}$, $\mathcal{S}^{\star\star} \leftarrow \check{\mathcal{S}}_{(k)}$;
 $\mathcal{L}^{\star} \leftarrow (\widetilde{\mathcal{M} + P_{\mathfrak{S}}}(\mathcal{L}_{(k)}) - \widetilde{\mathcal{S}}^{\star\star})/2$;
 $\mathcal{S}^{\star} \leftarrow (\widehat{\mathcal{M}} - \widehat{\mathcal{L}}^{\star\star} + \mathcal{S}_{(k)})/2$;
 $\mathcal{L}_{(k+1)} \leftarrow$
  $\mathcal{L}_{(k)} + t_k \left(\mathrm{shrink}^{\mathrm{t}}(2\mathcal{L}^{\star} - \mathcal{L}_{(k)}, 2\gamma) - \mathcal{L}^{\star}\right)$;
 $\mathcal{S}_{(k+1)} \leftarrow \mathcal{S}_{(k)} + t_k \left(\mathrm{ST}(2\mathcal{S}^{\star} - \mathcal{S}_{(k)}, 2\gamma\lambda) - \mathcal{S}^{\star}\right)$;
 $k \leftarrow k + 1$;
until *convergence*;
$\mathcal{L}^{\star\star} \leftarrow \underset{\sim}{P_{\mathfrak{S}}(\mathcal{L}_{(k)})}$, $\mathcal{S}^{\star\star} \leftarrow \check{\mathcal{S}}_{(k)}$;
$\mathcal{L}^{\star} \leftarrow (\widetilde{\mathcal{M} + P_{\mathfrak{S}}}(\mathcal{L}_{(k)}) - \widetilde{\mathcal{S}}^{\star\star})/2$;
$\mathcal{S}^{\star} \leftarrow (\widehat{\mathcal{M}} - \widehat{\mathcal{L}}^{\star\star} + \mathcal{S}^{(k)})/2$;
$[\mathcal{L}, \mathcal{S}] \leftarrow [\underset{\sim}{\mathcal{L}^{\star}}, \check{\mathcal{S}}^{\star}]$;

Algorithm 2 is reduced to $\mathbb{A}_m$- DRS-PCP proposed in [21]. If $m = 1$, the proposed algorithm solves the same problem as tensor robust PCA for real cases in e.g., [23], [26]. Therefore, our proposed method is a natural generalization of these state-of-the-art methods to general C-D domain and solves tensor robust PCA in the most general cases.

Lastly, we state the convergence of the proposed algorithm.

Theorem 4 (Convergence of $\mathbb{A}_m$-DRS-TPCP)**.** *Let parameters of Algorithm 2 be chosen so that* $\gamma \in (0, +\infty)$, $(t_k)_{k \geq 0} \subset [0, 2]$ *satisfying* $\sum_{k \geq 0} t_k(2 - t_k) = +\infty$. *Then, the output of Algorithm 1 converges to a minimizer of (9).*

Remark 2. In this paper, we employ the DRS for solving (9) but it can be also solved by other advanced convex optimization techniques such as the *alternating direction method of multipliers (ADMM)* [27] as used in [23], [26] and the *primal-dual splitting (PDS)* [28], [29].

It is worth mentioning that this paper focuses on 3-way tensors. But it may not be difficult to generalize $\mathbb{A}_m$-DRS-TPCP to the case of order-p ($p \geq 3$) tensors, by using the t-SVD for order-p tensors in [30].

V. Numerical Examples

In this section, we perform some numerical experiments for examining the effectiveness of the proposed method. Following general settings in e.g., [23], [24], [26], [31], we randomly generate ground truth pairs $(\mathcal{L}, \mathcal{S})$ as follows: $\mathcal{L} := \mathcal{L}_L * \mathcal{L}_R^{\mathsf{H}} \in \mathbb{A}_m^{N_1 \times N_2 \times N_3}$, where $\mathcal{L}_L \in \mathbb{A}_m^{N_1 \times r \times N_3}$ and $\mathcal{L}_R \in \mathbb{A}_m^{N_2 \times r \times N_3}$ ($r < \min(N_1, N_2)$) with the all real and imaginary parts of each entry of $\mathcal{L}_L, \mathcal{L}_R$ being i.i.d from $\mathcal{N}(0, 1)$. We choose the support set of $\mathcal{S}$ uniformly at random from all support set of size $\rho N_1 N_2 N_3$ ($\rho \in (0, 1)$). All real and imaginary parts of the non-zero entries are independently drawn form $\mathcal{U}(-256, 256)$. The input is generated by $\mathcal{X} := \mathcal{L} + \mathcal{S} + \mathcal{N}$ with a sufficiently small perturbation $\mathcal{N}$ with the all real and imaginary parts being i.i.d. from $\mathcal{N}(0, \sigma^2)$. We fixed $\lambda = 1/\sqrt{\max(N_1, N_2) N_3}$, $\gamma = 1$, $\sigma = 10^{-8}$ in all experiments. We perform experiments in the case where $\mathbb{A}_m = \mathbb{H}$ ($m = 4$) and $\mathbb{O}$ ($m = 8$). We compare the proposed method $\mathbb{A}_m$-DRS-PCP and two part-wise DRS-TPCP method, $\mathbb{C}^n$-DRS-TPCP and $\mathbb{R}^{2n}$-DRS-TPCP ($n = 2$ or 4, for $\mathbb{H}$ or $\mathbb{O}$) based on the standard t-SVD in complex and real domains. These part-wise methods split $\mathbb{H}$ into $\mathbb{C}^2$ and $\mathbb{R}^4$, or $\mathbb{O}$ into $\mathbb{C}^4$ and $\mathbb{R}^8$, and then estimate those parts separately. TABLE I shows the performance comparisons of all three algorithms with their outputs $\mathcal{L}^*$ and $\mathcal{S}^*$. They show that the proposed method $\mathbb{A}_m$-DRS-PCP successfully recovers the original tensor up to the noise level and outperforms all part-wise methods in both $\mathbb{H}$ and $\mathbb{O}$, by exploiting all correlations among real and imaginary parts. $\mathbb{C}^2$-DRS-PCP performs recovery a little bit better than $\mathbb{R}^4$-DRS-PCP since it may utilize these correlations in part.

TABLE I
Performance comparison in $\mathbb{H}$ and $\mathbb{O}$

$\mathcal{L}, \mathcal{S} \in \mathbb{H}^{32 \times 32 \times 8}, \rho = 0.2, r = 4$		
Algorithm	$\frac{\\|\mathcal{L}^* - \mathcal{L}\\|_F}{\\|\mathcal{L}\\|_F}$	$\frac{\\|\mathcal{S}^* - \mathcal{S}\\|_F}{\\|\mathcal{S}\\|_F}$
$\mathbb{A}_m$-DRS-TPCP	**6.06e-10**	**1.13e-10**
$\mathbb{C}^2$-DRS-TPCP	2.46e-1	4.76e-2
$\mathbb{R}^4$-DRS-TPCP	8.15e-1	1.58e-1
$\mathcal{L}, \mathcal{S} \in \mathbb{O}^{16 \times 16 \times 4}, \rho = 0.2, r = 1$		
Algorithm	$\frac{\\|\mathcal{L}^* - \mathcal{L}\\|_F}{\\|\mathcal{L}\\|_F}$	$\frac{\\|\mathcal{S}^* - \mathcal{S}\\|_F}{\\|\mathcal{S}\\|_F}$
$\mathbb{A}_m$-DRS-TPCP	**1.30e-9**	**5.18e-11**
$\mathbb{C}^4$-DRS-TPCP	2.60e-1	1.17e-2
$\mathbb{R}^8$-DRS-TPCP	6.10e-1	2.73e-2

VI. Conclusion

In this paper, we have proposed new algebraic translations of hypercomplex 3-way tensors with C-D extensions of t-SVD, tensor multi rank, tensor tubal rank and tensor low tubal rank approximation. The proposed translations are based on algebraic translations of C-D matrices and we have shown that useful algebraic properties are available in t-product of two hypercomplex tensors. We also have proposed an algorithmic solution to hypercomplex tensor principal component pursuit based on a proximal splitting technique. This solution solves the hypercomplex tensor principal component pursuit, which is a convex relaxation of hypercomplex tensor robust principal component analysis, utilizes the proposed mathematical tools including C-D t-SVD and tensor nuclear norm of hypercomplex tensors. Numerical experiments show that the proposed algorithm separates the observed tensors into the sum of low rank and sparse ones much more faithfully than existing algorithms.

ACKNOWLEDGEMENTS

This work was supported in part by JSPS Grants-in-Aid (18K19804) and by JST SICORP (JPMJSC20C6).

APPENDIX

The *Douglas-Rachford splitting (DRS)* [22], [32], [33] is a well-defined proximal splitting method that solves the minimization of the sum of two functions

$$f(\boldsymbol{x}) + g(\boldsymbol{x}), \tag{13}$$

where f and g are assumed to be elements of the class, denoted by $\Gamma_0(\mathcal{H})$, of proper lower semicontinuous convex functions from a real Hilbert space $\mathcal{H}$ to $\mathbb{R} \cup \{+\infty\}$. For given $\gamma \in (0, +\infty)$, the DRS approximates a minimizer of (13) with $\left(\operatorname{prox}_{\gamma g}(x_k)\right)_{k\geq 0}$ by generating the following sequence $(\boldsymbol{x}_k)_{k\geq 0}$:

$$\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + t_k\{\operatorname{prox}_{\gamma f}[2\operatorname{prox}_{\gamma g}(\boldsymbol{x}_k) - \boldsymbol{x}_k] - \operatorname{prox}_{\gamma g}(\boldsymbol{x}_k)\}, \tag{14}$$

where $(t_k)_{k\geq 0} \subset [0, 2]$ satisfies $\sum_{k\geq 0} t_k(2 - t_k) = +\infty$ and the proximity operator [34] of index γ of $f \in \Gamma_0(\mathcal{H})$ is defined as

$$\operatorname{prox}_{\gamma f} : \mathcal{H} \to \mathcal{H} : \boldsymbol{x} \mapsto \underset{\boldsymbol{y}\in\mathcal{H}}{\arg\min} \left\{ f(\boldsymbol{y}) + \frac{1}{2\gamma} \|\boldsymbol{x} - \boldsymbol{y}\|_{\mathcal{H}}^2 \right\}$$

with the norm on $\mathcal{H}$ denoted by $\|\cdot\|_{\mathcal{H}}$. Indeed, if $\dim(\mathcal{H}) < \infty$, $\left(\operatorname{prox}_{\gamma g}(x_k)\right)_{k\geq 0}$ converges to a minimizer of (13) (see e.g., [35]).

REFERENCES

[1] S. B. Choe and J. J. Faraway, "Modeling head and hand orientation during motion using quaternions," *J. Aerospace*, vol. 113, no. 1, pp. 186–192, 2004.
[2] J. C. K. Chou, "Quaternion kinematic and dynamic differential equations," *IEEE Trans. Robot. Autom.*, vol. 8, no. 1, pp. 53–64, Feb. 1992.
[3] J. S. Yuan, "Closed-loop manipulator control using quaternion feedback," *IEEE Trans. Robot. Autom.*, vol. 4, no. 4, pp. 434–440, Aug. 1988.
[4] S. L. Goh, M. Chen, D. H. Popovi, K. Aihara, D. Obradovic, and D. P. Mandic, "Complex-valued forecasting of wind profile," *Renewable Energy*, vol. 31, pp. 1733–1750, 2006.
[5] C. C. Took, D. P. Mandic, and K. Aihara, "Quaternion-valued short term forecasting of wind profile," in *Proc. IJCNN*, 2010.
[6] I. Kantor and A. Solodovnikov, *Hypercomplex numbers, An Elementary Introduction to Algebras*. Springer-Verlag, New York, 1989.
[7] D. Alfsmann, "On families of 2^N-dimenisonal hypercomplex algebras suitable for digital signal processing," in *Proc. EUSIPCO*, Florence, Italy, 2006.
[8] J. C. Baez, "The octonions," *Bull. AMS*, vol. 39, no. 2, pp. 145–205, 2001.
[9] N. L. Bihan and S. Buchholz, "Quaternionic independent component analysis using hypercomplex nonlinearities," in *Proc. IMA Conf. Math. Signal Process.*, Cirenceister, UK, 2006.
[10] F. Zhang, "Quaternions and matrices of quaternions," *Linear Algebra and Its Applications*, vol. 251, pp. 21–57, 1997.
[11] T. Mizoguchi and I. Yamada, "An algebraic translation of Cayley-Dickson linear systems and its applications to online learning," *IEEE Trans. Signal Process.*, vol. 62, no. 6, pp. 1438–1453, Mar. 2014.
[12] ——, "Hypercomplex tensor completion via convex optimization," *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 4078–4092, Aug. 2019.
[13] N. Sidiropoulos and R. Bro, "On the uniqueness of multilinear decomposition of n-way arrays," *J. Chemometrics*, vol. 14, no. 3, pp. 229–239, 2000.
[14] T. Kolda and B. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
[15] C. J. Hillar and L.-H. Lim, "Most tensor problems are NP-hard," *J. ACM*, vol. 60, no. 6, pp. 1–39, Nov. 2013.
[16] L. D. Lathauwer, "A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization," *SIAM J. Matrix Anal. & Appl.*, vol. 28, no. 3, pp. 642–666, 2006.
[17] I. Domanov and L. D. Lathauwer, "Canonical polyadic decomposition of third-order tensors: Reduction to generalized eigenvalue decomposition," *SIAM J. Matrix Anal. & Appl.*, vol. 35, no. 2, pp. 636–660, 2014.
[18] R. Akema, M. Yamagishi, and I. Yamada, "Polyadic tensor decomposition via structured low-rank matrix approximation," *IEICE Trans. Fundamentals (to appear)*, 2021.
[19] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra and Its Applications*, vol. 251, pp. 641–658, 2011.
[20] T. A. Ell, N. L. Bihan, and S. J. Sangwine, *Quaternion Fourier Transforms for Signal and Image Processing*. Wiley, 2014.
[21] T. Mizoguchi and I. Yamada, "Hypercomplex principal component pursuit via convex optimization," in *Proc. APSIPA ASC*, Honolulu, HI, USA, 2018.
[22] P. L. Combettes and J. C. Pesquet, "A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 564–574, Dec. 2007.
[23] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-SVD," in *Proc. CVPR*, 2014.
[24] S. Gandy and I. Yamada, "Convex optimization techniques for the efficient recovery of a sparsely corrupted low-rank matrix," *J. Math-for-Industry*, vol. 2, pp. 147–156, 2010.
[25] F. R. Deutsch, *Best Approximation in Inner Product Spaces*. Springer-Verlag, New York, 2001.
[26] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization," in *Proc. CVPR*, 2018.
[27] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 2, pp. 1–122, 2011.
[28] L. Condat, "A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms," *J. Optim. Theory & Appl.*, vol. 158, no. 2, pp. 460–479, 2013.
[29] B. C. Vũ, "A splitting algorithm for dual monotone inclusions involving cocoercive operators," *Advances in Computational Mathematics*, vol. 38, no. 3, pp. 667–681, Apr. 2013.
[30] C. D. Martin, R. Shafer, and B. LaRue, "An order-*p* tensor factorization with applications in imaging," *SIAM J. Sci. Anal. Comput.*, vol. 35, no. 1, pp. A474–A490, 2013.
[31] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," UIUC Technical Report UILU-ENG-09-2214, Tech. Rep., Jul. 2009.
[32] J. Douglas and H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," *Trans. AMS*, vol. 82, pp. 421–439, 1956.
[33] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Programming*, vol. 55, pp. 293–318, Apr. 1992.
[34] J. J. Moreau, "Fonctions convexes duales et points proximaux dans unespace hilbertien," *Comptes Rendus de l'Académie des Sciences (Paris), Série A*, vol. 255, pp. 2897–2899, 1962.
[35] I. Yamada, M. Yukawa, and M. Yamagishi, "Minimizing the Moreau envelope of nonsmooth convex functions over the fixed point set of certain quasi-nonexpansive mappings," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. H. Baischke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, Eds. Springer, 2011, ch. 17, pp. 345–390.