Fast-Parallel Singular Value Thresholding for Many Small Matrices based on Geometric Feature of Singular Values

Takayuki Sasaki*, Ryuichi Tanida*, Masaki Kitahara*, Hideaki Kimata†
* NTT Media Intelligence Laboratories, NTT Corporation, Yokosuka, Japan
E-mail: {takayuki.sasaki.fb, ryuichi.tanida.pc, masaki.kitahara.ve}@hco.ntt.co.jp Tel: +81-46-859-8967
† Kogakuin University, Tokyo, Japan
E-mail: kimata@ieee.org

***Abstract*—This paper proposes a singular value thresholding (SVT) method called fast-parallel SVT for many small matrices that is based on the geometric feature of singular values. By discovering the geometric feature of singular values, SVT is expressed in a completely different format from conventional SVT methods. The new format does not require singular value decomposition, which is the calculation bottleneck with conventional SVT methods, so the amount of floating-point calculation can be reduced. The new format can also be described almost linearly, allowing many matrices to be processed in data in parallel. Through an experiment, the proposed method calculated SVT up to 162.84 times faster with higher calculation accuracy than conventional SVT methods.**

I. Introduction

Low-rankness is important for modeling analysis targets such as data and physical phenomena. Many analysis targets may have low-rankness. When the rank of the matrix derived from the analysis target is extremely small compared with the number of its rows and columns, such a matrix is said to be low-rank. Many researchers have been studying low-rank analysis in computer vision [1]–[6], image processing [7], [8], genome data analysis [9], and graph analysis [10].

However, the direct use of the rank function to low-rank optimization is not a good policy for the following two reasons.

- The rank function is discontinuous, non-differentiable, and non-convex. Therefore, the optimization problem based on the rank function is NP-hard combinatorial optimization.
- The actual analysis targets are not exactly low-rank but are approximately low-rank. Therefore, using the rank function is too restrictive to model the targets.

For these reasons, many researchers have used a relaxation approach that regularizes the nuclear norm instead of the rank function. The nuclear norm is the convex envelope of the rank function [11]. Therefore, the optimization problem regularizing the nuclear norm can promote low-rankness.

Iterative calculation based on a first-order method, such as alternating direction method of multipliers [12], can solve the optimization problem including the nuclear norm. The first-order method repeatedly executes singular value thresholding (SVT), which is the proximal mapping of the nuclear norm. However SVT is a computationally intensive process because it includes calculation of singular value decomposition (SVD), which requires a large amount of calculation.

TABLE I
Types of low-rank optimization and speedup methods

	1) Few large matrices	2) Many small matrices
Application	Robust PCA [1], [2] Image interpolation [3], [4] Optical flow estimation [5], [6] Dynamic MRI analysis [7] Genome data analysis [9]	Color-artifact removal [8] Graph simplification [10]
Speedup	FSVT [13], FRSVT [14]	FPSVT(proposed)

We classify each application field by the number and size of matrices to which low-rank optimization is applied (Table I). There are two types of optimization problems;

1) the problem of regularizing a few large matrices
2) the problem of regularizing many small matrices

With this classification, we can understand the difference between high-speed processing methods of SVT. For optimization problem 1), methods for speeding up SVT have been proposed. Cai et al. [13] proposed Fast SVT (FSVT), which calculates SVT without SVD. Oh et al. [14] proposed Fast Randomized SVT (FRSVT), which reduces the SVD input size and speeds up SVT by approximating a large matrix to the product of an orthogonal matrix and a small core matrix. Both methods reduce the amount of calculation when the input matrix size is large (number of rows, number of columns $= 500$ - 2000) and show significant increase in SVT speed.

For optimization problem 2), a method for speeding up SVT has not been developed. Even if one applies such a method for problem 1) to problem 2), its effectiveness will be limited. In fact, FSVT is about 20 times slower and FRSVT is about 4 times slower than the method using SVD. In addition, these methods cannot use a data-parallel approach to simultaneously process many matrices. Therefore, they cannot effectively use the resources of recent parallel computing architectures such as central-processing-unit and graphics-processing-unit single instruction, multiple data functions.

We now discuss the requirements for the number and size of matrices for many small matrices. For color-artifact removal [8], an input image is divided into blocks that allow overlapping, and the pixel values of the blocks are arranged to form a matrix. Let L be the number of blocks, M be the number of pixels of a block, and N be the number of color channels. Then the number of matrices is L and the size of matrices is $M \times N$. Assuming an RGB color image, we can estimate that L is several thousand, M is several hundred, and N is 3. For graph simplification [10], if the number of vertices with degree 2 is L, then the number of matrices is L, and the matrix size is 2×2. If we assume that the number of vertices in the graph is several thousand, L is also several thousand. From the above, we need to consider SVT with thousands of matrices and a matrix size of several hundred $\times$ 2 or 3.

We propose Fast-parallel SVT (FPSVT), which calculates SVT in data parallel with reduced computational complexity. FPSVT involves calculating SVT without requiring SVD and reduces the complexity and parallelizes data at the same time. FPSVT can speed up the SVT for L matrices of size $M \times 2$. Although it only partially satisfies the aforementioned requirements for matrix size, the performance is sufficient to speed up graph simplification. The calculation accuracy is high because it is not an approximation method. The core of this derivation is based on the discovery that we can represent the nuclear norm geometrically with an $L_{\infty,2}$ mixed norm when the size of the matrix is limited to $M \times 2$. We conducted an experiment to evaluate the speed and accuracy of FPSVT.

There are two contributions for this paper. The first is the derivation of several theorems for SVD and SVT. The second is the derivation of fast SVT computations for many small matrices.

The remainder of the paper is organized as follows. In Section II, we discuss the low-rank optimization by nuclear norm regularization as prior knowledge. In Section III we derive FPSVT. In Section IV, we discuss our experiment and present the experimental results. Finally, we conclude the paper in Section V.

II. Low-rank optimization by nuclear norm regularization

A. Notation and definition

The function vec is a linear transformation that rearranges the input matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2] \in \mathbb{R}^{M\times 2}$ into a column vector $\mathrm{vec}\mathbf{X} = \begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix} \in \mathbb{R}^{2M}$. Conversely, $\mathrm{vec}^{\mathrm{T}}$ is an adjoint transform of vec, which transforms a vector into a matrix ($\mathbf{X} = \mathrm{vec}^{\mathrm{T}}\mathrm{vec}\mathbf{X}$).

The subspace $\mathrm{Im}\mathbf{Y} = \{\mathbf{Y}\mathbf{x} = \sum_{i=1}^{N} x_i \mathbf{y}_i | \mathbf{x} \in \mathbb{R}^N\} \subset \mathbb{R}^M$ is called the image of $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N] \in \mathbb{R}^{M\times N}$. From this definition, $\mathbf{y}_1, \cdots, \mathbf{y}_N \in \mathrm{Im}\mathbf{Y}$ holds.

A function that non-negatively clips each element of a vector or matrix is called a ramp function and is denoted as $(\cdot)_+$. For example, if $\mathbf{Y} = \mathbf{X}_+$, each component is $y_{i,j} = x_{i,j} (\textit{if } x_{i,j} \geq 0), 0 (\textit{otherwise})$.

The Euclidean norm of a vector or the Frobenius norm of a matrix is simply represented as the norm $\|\cdot\|$. For a function g and positive real number $\mu > 0$, the proximal mapping [15] is defined as

$$\mathrm{prox}_{\mu g}(\mathbf{Y}) = \underset{\mathbf{Z}\in\mathbb{R}^{M\times N}}{\arg\min}\, g(\mathbf{Z}) + \frac{1}{2\mu}\|\mathbf{Z} - \mathbf{Y}\|^2. \tag{1}$$

Several functions are known for efficient computation of proximal mappings, and such functions are called proximable.

B. Formulation and solution of optimization problems

We summarize low-rank optimization, which is the subject of this study. Let $\sigma_i(\mathbf{X}), i = 1, \cdots, K = \min(M, N)$ be the ith largest singular value of matrix $\mathbf{X} \in \mathbb{R}^{M\times N}$. The nuclear norm is defined as the sum of singular values

$$\|\mathbf{X}\|_* = \sum_{i=1}^{K} \sigma_i(\mathbf{X}). \tag{2}$$

Since the nuclear norm is a convex envelope of the function $\mathrm{rank}\mathbf{X}$, it promotes low-rankness by regularizing it [11].

We focus on an optimization problem in which many small matrices are regularized with a nuclear norm. Typically, this is an optimization problem of the following form for the variable x in Hilbert space $\mathcal{X}$;

$$\min_{x\in\mathcal{X}} f(x) + \lambda \sum_{i=1}^{L} \|\mathit{\Phi}_i(x)\|_*, \tag{3}$$

where the first term of the objective function is a fidelity term, and the second term is a low-rank regularization term using the nuclear norm. The function $\mathit{\Phi}_i : \mathcal{X} \to \mathbb{R}^{M\times N}$ is a low-rank matrix-generation function, typically a linear mapping.

For the low-rank regularization of the second term, it is enough to introduce an auxiliary variable $\mathbf{Y}_i = \mathit{\Phi}_i(x)$ for each i and compute the proximal mapping of $g(\mathbf{Y}_1, \cdots, \mathbf{Y}_L) = \sum_{i=1}^{L} \|\mathbf{Y}_i\|_*$. This can be computed independently for each $\mathbf{Y}_i$, and

$$\mathrm{prox}_{\mu\|\cdot\|_*}(\mathbf{Y}_i), \quad i = 1, \cdots, L \tag{4}$$

can be computed separately. Cai et al. [3] showed that the proximal mapping of the nuclear norm $\|\cdot\|_*$ is equivalent to SVT. SVT is calculated as

$$\mathrm{prox}_{\mu\|\cdot\|_*}(\mathbf{Y}) = \mathbf{U}(\mathbf{\Sigma} - \mu\mathbf{I})_+\mathbf{V}^{\mathrm{T}} \tag{5}$$

$$\textit{where} \quad \mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}} \tag{6}$$

for the input matrix $\mathbf{Y}$, where (6) is SVD [1].

Therefore, the proximal mapping (4) of the function $g(\mathbf{Y}_1, \cdots, \mathbf{Y}_L)$ requires L times of SVT calculation (5) and (6), and it is necessary to calculate SVD repeatedly with a first-order method. SVD calculation is generally more computationally intensive than other processes, and about 99% of the calculation time is spent on SVD calculation.

[1]SVD in this article is "thin SVD". Therefore, for $\mathbf{Y} \in \mathbb{R}^{M\times N}$ and $K = \min(M, N)$, the singular vectors are $\mathbf{U} \in \mathbb{R}^{M\times K}$ and $\mathbf{V} \in \mathbb{R}^{N\times K}$. The calculation of $K+1$ and subsequent singular vectors is omitted. The singular value matrix $\mathbf{\Sigma}$ is a $K \times K$ diagonal matrix.

III. Fast-parallel Singular Value Thresholding

In this section, we discuss in more detail FPSVT, which processes many SVT calculations at high speed. This method is based on the discovery of a geometric property in which the nuclear norm is characterized by the distance of a vector in a subspace. Because of this property, we can represent the nuclear norm without singular values. We obtain new SVT without SVD. Since we can describe most of this SVT calculation with linear transformation, data-parallel algorithms can be derived.

With FPSVT, we limit the size of the input matrix to $M \times 2$ and $2 \times N$. Since $\mathrm{prox}_{\mu\|\cdot\|_*}(\mathbf{Y}) = (\mathrm{prox}_{\mu\|\cdot\|_*}(\mathbf{Y}^{\mathrm{T}}))^{\mathrm{T}}$ holds, we can limit the size of the input to $M \times 2$ without loss of generality. Therefore, in the following subsections, we consider the SVT of a vertically long $M \times 2$ matrix.

A. Geometric properties of singular values and new SVT representations

This subsection describes the geometric properties of the singular values and representation of SVT that does not require SVD, which make up the core of FPSVT derivation. This subsection presents only the results, and we give derivation details in the appendix. Let $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2] \in \mathbb{R}^{M\times 2}, \mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^M$ be the input matrix, and let $\sigma_1, \sigma_2 (\sigma_1 \geq \sigma_2 \geq 0)$ be its singular values.

Proposition 1 The sum and difference of singular values $\sigma_1 \pm \sigma_2$ are

$$\sigma_1 \pm \sigma_2 = \sqrt{\mathrm{tr}\mathbf{Y}^{\mathrm{T}}\mathbf{Y} \pm 2\sqrt{\det \mathbf{Y}^{\mathrm{T}}\mathbf{Y}}} \tag{7}$$
$$= \|\mathbf{y}_1 \mp \mathbf{R}\mathbf{y}_2\|, \tag{8}$$

where $\mathbf{R} \in \mathbb{R}^{M\times M}$ is a rotation matrix that rotates the vector on $\mathrm{Im}\mathbf{Y}$ by $\pi/2$ rad around the origin. This rotation matrix is characterized by the following conditions.

$$\mathbf{R}\mathbf{y} \in \mathrm{Im}\mathbf{Y}, \|\mathbf{R}\mathbf{y}\| = \|\mathbf{y}\|, \mathbf{y}^{\mathrm{T}}\mathbf{R}\mathbf{y} = 0, \mathbf{R}^{\mathrm{T}}\mathbf{R}\mathbf{y} = \mathbf{R}\mathbf{R}^{\mathrm{T}}\mathbf{y} = \mathbf{y} \tag{9}$$

for any $\mathbf{y} \in \mathrm{Im}\mathbf{Y}$.

The direction that reaches the shortest distance from $\mathbf{y}_1$ to $\mathbf{y}_2$ is the positive direction of rotation, as shown in Fig. 1.

We give the proof in Appendix A. Proposition 1 claims that the sum and difference of singular values is the Euclidean distance between vectors $\mathbf{y}_1$ and $\pm\mathbf{R}\mathbf{y}_2$. The intermediate representation (7) is easy to calculate because $\mathbf{Y}^{\mathrm{T}}\mathbf{Y}$ is a 2×2 matrix; $\mathbf{Y}^{\mathrm{T}}\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^{\mathrm{T}}\mathbf{y}_1 & \mathbf{y}_1^{\mathrm{T}}\mathbf{y}_2 \\ \mathbf{y}_1^{\mathrm{T}}\mathbf{y}_2 & \mathbf{y}_2^{\mathrm{T}}\mathbf{y}_2 \end{bmatrix}$.

From Proposition 1 and the definition of the nuclear norm (2), we can immediately find the following corollary.

Corollary 2 Nuclear norm $\|\mathbf{Y}\|_*$ can be represented as

$$\|\mathbf{Y}\|_* = \|\mathbf{A}\mathrm{vec}\mathbf{Y}\|_{\infty,2}, \tag{10}$$

where matrix $\mathbf{A} \in \mathbb{R}^{2M\times 2M}$ is

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & -\mathbf{R} \\ \mathbf{I} & \mathbf{R}, \end{bmatrix} \tag{11}$$

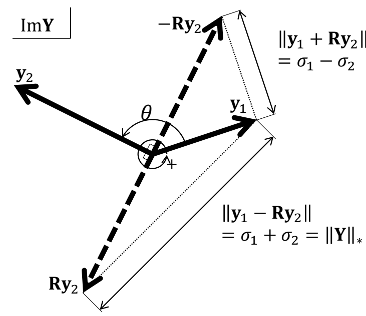

Fig. 1. Geometric properties of nuclear norm

and $\|\cdot\|_{\infty,2}$ is the $L_{\infty,2}$ mixed norm [2].

Corollary 2 claims that the nuclear norm can be represented without using singular values. This suggests that SVT, which is the proximal mapping of the nuclear norm, is also obtained without SVD. In fact, combining Corollary 2 with Lemmas 7 and 8 in the Appendix yields the following proposition.

Proposition 3 For the SVT of matrix $\mathbf{Y}$,

$$\mathrm{prox}_{\mu\|\cdot\|_*}(\mathbf{Y}) = \mathrm{vec}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathrm{prox}_{\mu\|\cdot\|_{\infty,2}}\left(\frac{1}{2}\mathbf{A}\mathrm{vec}\mathbf{Y}\right) \tag{12}$$

holds.

The proof is given in Appendix B. This gives a new representation of SVT, which is a composition of linear transformation $\mathbf{A}\mathrm{vec}(\cdot)$ and $L_{\infty,2}$ norm proximal mapping $\mathrm{prox}_{\mu\|\cdot\|_{\infty,2}}(\cdot)$, where the proximal mapping $\mathrm{prox}_{\mu\|\cdot\|_{\infty,2}}(\cdot)$ is nonlinear but can be described as a linear transformation that depends on the input $\mathbf{x}$ as follows.

Proposition 4 For $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^M$,

$$\mathrm{prox}_{\mu\|\cdot\|_{\infty,2}}(\mathbf{x}) = \begin{bmatrix} k_1\mathbf{x}_1 \\ k_2\mathbf{x}_2 \end{bmatrix}$$

where

$$(k_1, k_2) = \begin{cases} \left(\frac{(\|\mathbf{x}_1\| + \|\mathbf{x}_2\| - \mu)_+}{2\|\mathbf{x}_1\|}, \frac{(\|\mathbf{x}_1\| + \|\mathbf{x}_2\| - \mu)_+}{2\|\mathbf{x}_2\|}\right) \\ \quad \textit{if } |\|\mathbf{x}_1\| - \|\mathbf{x}_2\|| \leq \mu \\ \left(1 - \frac{\mu}{\|\mathbf{x}_1\|}, 1\right) \quad \textit{if } \|\mathbf{x}_1\| - \|\mathbf{x}_2\| > \mu \\ \left(1, 1 - \frac{\mu}{\|\mathbf{x}_2\|}\right) \quad \textit{if } \|\mathbf{x}_1\| - \|\mathbf{x}_2\| < -\mu \end{cases} \tag{13}$$

holds [3].

The proof of this proposition (Appendix C) is based on the Moreau decomposition [16] and projection onto the $L_{1,2}$ mixed norm sphere [17].

From the above, it was clarified that SVT can be represented by linear transformation, excluding calculation of coefficients

[2] $L_{\infty,2}$ mixed norm is a composite function of L_∞ norm and L_2 norm. For any $\mathbf{x} = [\mathbf{x}_1^{\mathrm{T}}, \mathbf{x}_2^{\mathrm{T}}]^{\mathrm{T}}$, $\|\mathbf{x}\|_{\infty,2} = \max(\|\mathbf{x}_1\|, \|\mathbf{x}_2\|)$ holds.

[3] It is $k_i\mathbf{x}_i = (0/0)\mathbf{0}$ if $\mathbf{x}_i = \mathbf{0}$, but here it is exceptionally $k_i\mathbf{x}_i = \mathbf{0}$.

k_1 and k_2. In the following subsection, we expand the above formula and present an SVT calculation formula and algorithms.

B. SVT calculation

From the previous subsection propositions, we obtain the following theorem for calculating SVT.

Theorem 5 For $\mathrm{rank}\mathbf{Y} = 2$ and $\sigma_1 \neq \sigma_2$,

$$\mathrm{prox}_{\mu\|\cdot\|_*}(\mathbf{Y}) = \gamma(1-\delta)\mathbf{Y} + \gamma\delta\mathbf{R}\overline{\mathbf{Y}} \tag{14}$$

where

$$\gamma = \left(1 - \frac{(\mu-\sigma_2)_+}{\sigma_1-\sigma_2}\right)_+, \delta = \frac{\min(\mu,\sigma_2)}{\sigma_1+\sigma_2}, \tag{15}$$

$$\overline{\mathbf{Y}} = [\mathbf{y}_2, -\mathbf{y}_1] \tag{16}$$

holds.

The above theorem is obtained by expanding the prox calculation of (12).

Theorem 5 claims that SVT can be calculated from the linear combination of matrices $\mathbf{Y}$ and $\mathbf{R}\overline{\mathbf{Y}}$. The coefficients $\gamma(1-\delta)$ and $\gamma\delta$ are composed of the amplitude parameter γ and internal ratio parameter δ.

To apply Theorem 5, it is necessary to find the concrete value of transformation $\mathbf{R}\overline{\mathbf{Y}}$ by rotation matrix $\mathbf{R}$. This approach is divided into two cases: $M \geq 3$ and $M = 2$. First, if $M \geq 3$, matrix $\mathbf{R}$ must satisfy all the conditions in (9) and have the rotation direction described in Proposition 1. We found that matrix $\mathbf{R}$ satisfying these conditions is

$$\mathbf{R} = \frac{1}{\sqrt{\det \mathbf{Y}^{\mathrm{T}}\mathbf{Y}}}(\mathbf{y}_2\mathbf{y}_1^{\mathrm{T}} - \mathbf{y}_1\mathbf{y}_2^{\mathrm{T}}). \tag{17}$$

However, the procedure for calculating the matrix product $\mathbf{R}\overline{\mathbf{Y}}$ after calculating $\mathbf{R}$ with (17) has $\mathcal{O}(M^2)$ order complexity. Therefore, we calculate the inner product first and obtain the representation

$$\mathbf{R}\overline{\mathbf{Y}} = \frac{1}{\sqrt{\det \mathbf{Y}^{\mathrm{T}}\mathbf{Y}}}\mathbf{Y}\begin{bmatrix} -\mathbf{y}_2^{\mathrm{T}}\mathbf{y}_2 & \mathbf{y}_1^{\mathrm{T}}\mathbf{y}_2 \\ \mathbf{y}_1^{\mathrm{T}}\mathbf{y}_2 & -\mathbf{y}_1^{\mathrm{T}}\mathbf{y}_1. \end{bmatrix}. \tag{18}$$

As a result, the calculation order can be reduced to $\mathcal{O}(M)$. For (14), we obtain

$$\mathrm{prox}_{\mu\|\cdot\|_*}(\mathbf{Y}) = \mathbf{Y}\left(\gamma(1-\delta)\mathbf{I}_2 + \frac{\gamma\delta}{\sqrt{\det \mathbf{Y}^{\mathrm{T}}\mathbf{Y}}}\begin{bmatrix} -\mathbf{y}_2^{\mathrm{T}}\mathbf{y}_2 & \mathbf{y}_1^{\mathrm{T}}\mathbf{y}_2 \\ \mathbf{y}_1^{\mathrm{T}}\mathbf{y}_2 & -\mathbf{y}_1^{\mathrm{T}}\mathbf{y}_1 \end{bmatrix}\right) \tag{19}$$

and further reduce the amount of calculation.

Then, for $M = 2$, the establishment of

$$\mathbf{R} = \mathrm{sgn}(\det \mathbf{Y})\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{20}$$

is obvious using the sign function sgn. Therefore, we can calculate $\mathbf{R}\overline{\mathbf{Y}}$ using element replacement and sign inversion.

From the above, we described calculating SVT using Theorem 5. However, this theorem imposes the conditions of $\mathrm{rank}\mathbf{Y} = 2$ and $\sigma_1 \neq \sigma_2$. In other cases, it can be easily calculated as follows.

TABLE II
DETERMINING $\mathrm{rank}\mathbf{Y}$

	$\det \mathbf{Y}^{\mathrm{T}}\mathbf{Y} = 0$	$\det \mathbf{Y}^{\mathrm{T}}\mathbf{Y} \neq 0$
$\mathbf{Y} = \mathbf{O}$	$\mathrm{rank}\mathbf{Y} = 0$	—
$\mathbf{Y} \neq \mathbf{O}$	$\mathrm{rank}\mathbf{Y} = 1$	$\mathrm{rank}\mathbf{Y} = 2$

Algorithm 1 SVT calculation ($M \geq 3$)

Input: $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2] \in \mathbb{R}^{M\times 2}, \mu > 0$
Output: $\mathbf{Z} = \mathrm{prox}_{\mu\|\cdot\|_*}(\mathbf{Y})$
$a \leftarrow \mathbf{y}_1^{\mathrm{T}}\mathbf{y}_1, b \leftarrow \mathbf{y}_1^{\mathrm{T}}\mathbf{y}_2, c \leftarrow \mathbf{y}_2^{\mathrm{T}}\mathbf{y}_2, d \leftarrow ac - b^2$
$e \leftarrow \sqrt{d}, f \leftarrow a+c, g \leftarrow \sqrt{f+2e}, h \leftarrow \sqrt{f-2e}$
$\sigma_2 \leftarrow \frac{1}{2}(g-h)$
if $\mathbf{Y} = \mathbf{O}$ **then**
  $\mathbf{Z} \leftarrow \mathbf{O}$
else if $d = 0$ **then**
  $\mathbf{Z} \leftarrow \left(1 - \frac{\mu}{\sqrt{f}}\right)_+ \mathbf{Y}$
else if $h = 0$ **then**
  $\mathbf{Z} \leftarrow \left(1 - \frac{\sqrt{2}\mu}{\sqrt{f}}\right)_+ \mathbf{Y}$
else
  $\gamma \leftarrow \left(1 - \frac{(\mu-\sigma_2)_+}{h}\right)_+, \delta \leftarrow \frac{\min(\mu,\sigma_2)}{g}$
  $\mathbf{Z} \leftarrow \mathbf{Y}\left(\gamma(1-\delta)\mathbf{I}_2 + \frac{\gamma\delta}{e}\begin{bmatrix} -c & b \\ b & -a \end{bmatrix}\right)$
end if

Theorem 6 For cases other than "$\mathrm{rank}\mathbf{Y} = 2$ and $\sigma_1 \neq \sigma_2$",

$$\mathrm{prox}_{\mu\|\cdot\|_*}(\mathbf{Y}) = \begin{cases} \mathbf{O} & \textit{if } \mathrm{rank}\mathbf{Y} = 0 \\ \left(1 - \frac{\mu}{\|\mathbf{Y}\|}\right)\mathbf{Y} & \textit{if } \mathrm{rank}\mathbf{Y} = 1 \\ \left(1 - \frac{\sqrt{2}\mu}{\|\mathbf{Y}\|}\right)\mathbf{Y} & \textit{if } \mathrm{rank}\mathbf{Y} = 2 \text{ and } \sigma_1 = \sigma_2 \end{cases} \tag{21}$$

holds.

The proof is given in Appendix D. To calculate SVT using Theorems 5 and 6, it is necessary to distinguish cases appropriately from $\mathrm{rank}\mathbf{Y}$ and singular values σ_1 and σ_2. As shown in Table II, the rank can be determined by checking whether $\mathbf{Y}$ is a zero matrix and whether $\det \mathbf{Y}^{\mathrm{T}}\mathbf{Y}$ is 0. For singular values, $\sigma_1 \pm \sigma_2$ can be calculated using (7).

From the above considerations, we introduce two algorithms derived with FPSVT 1 and 2 to calculate SVT. Algorithm 1 shows the procedure when $M \geq 3$, and Algorithm 2 shows the procedure when $M = 2$. To reduce the amount of calculation, the values necessary for case classification and the coefficients γ and δ are all calculated using the inner product $\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_j (i, j = 1, 2)$. Most of the processing of Algorithms 1 and 2 consists of basic operations related to vectors and matrices, suggesting that parallelization is highly effective.

IV. EXPERIMENT

A. Implementation and methods for comparison

We conducted an experiment to evaluate the SVT-calculation performance of FPSVT. All source codes were implemented using MATLAB R2018a, CPU: Intel Core i7-3930K@3.20 GHz (6 cores, 12 threads), RAM: 32.0 GB. The

Algorithm 2 SVT calculation ($M=2$)

Input: $\mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2]=\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22}\end{bmatrix}\in\mathbb{R}^{2\times 2}, \mu>0$

Output: $\mathbf{Z}=\mathrm{prox}_{\mu\|\cdot\|_*}(\mathbf{Y})$

$a\leftarrow \mathbf{y}_1^{\mathrm{T}}\mathbf{y}_1, c\leftarrow \mathbf{y}_2^{\mathrm{T}}\mathbf{y}_2, d\leftarrow \det\mathbf{Y}$

$e\leftarrow |d|, f\leftarrow a+c, g\leftarrow \sqrt{f+2e}, h\leftarrow\sqrt{f-2e}$

$\sigma_2\leftarrow \frac{1}{2}(g-h)$

if $\mathbf{Y}=\mathbf{O}$ **then**

  $\mathbf{Z}\leftarrow\mathbf{O}$

else if $d=0$ **then**

  $\mathbf{Z}\leftarrow\left(1-\frac{\mu}{\sqrt{f}}\right)_+\mathbf{Y}$

else if $h=0$ **then**

  $\mathbf{Z}\leftarrow\left(1-\frac{\sqrt{2}\mu}{\sqrt{f}}\right)_+\mathbf{Y}$

else

  $\gamma\leftarrow\left(1-\frac{(\mu-\sigma_2)_+}{h}\right)_+, \delta\leftarrow\frac{\min(\mu,\sigma_2)}{g}$

  $\mathbf{Z}\leftarrow\gamma(1-\delta)\mathbf{Y}+\mathrm{sgn}(d)\gamma\delta\begin{bmatrix}-y_{22} & y_{21}\\ y_{12} & -y_{11}\end{bmatrix}$

end if

SVT methods used in the experiment were SVD-based SVT (SVDSVT) [3], as shown in (5) and (6), FSVT [13], FRSVT [14], and FPSVT, i.e., using Algorithm 1 or 2, as a non-parallel version (FPSVT-np) and parallel version (FPSVT-p). SVDSVT was implemented using the MATLAB standard SVD function and matrix product. We implemented FSVT in MATLAB with the number of iterations of polar decomposition and projection on L_2-induced norm ball set to 7 and 9, respectively, with reference to Cai et al.'s study [13]. FRSVT used the original MATLAB code created by Oh et al. [14]. The FRSVT core matrix size was set to 1, and the other parameters were set to the same settings as in Oh et al.'s study [14].

B. Experiment with random matrices

We applied the different SVT formats calculated with the above methods to the matrix generated from random numbers and observed the execution time and calculation accuracy.

1) Conditions: The test matrices used in the experiment were sets of random input test matrices $\mathbf{Y}_i, i=1,\cdots,L$ and output test matrices $\mathbf{Z}_i, i=1,\cdots,L$ that were the result of SVT. The output test matrices were used to verify the accuracy of the SVT calculated with each method.

We describe the test matrix-generation method as follows. First, we generated $M\times 2$ matrices $\mathbf{X}_i$, the elements of which are normal random numbers. Next, we carried out SVD on $\mathbf{X}_i$ to obtain a set of random singular vectors $\mathbf{U}_i$ and $\mathbf{V}_i$. We then generated random singular values $\sigma_1\in[0.5,1]$ and $\sigma_2\in[0,0.5]$ with uniform random numbers. Finally, we calculated an input test matrix $\mathbf{Y}_i=\mathbf{U}_i\mathrm{diag}(\sigma_1,\sigma_2)\mathbf{V}_i^{\mathrm{T}}$ and an output test matrix $\mathbf{Z}_i=\mathbf{U}_i\mathrm{diag}((\sigma_1-\mu)_+,(\sigma_2-\mu)_+)\mathbf{V}_i^{\mathrm{T}}$.

We calculated SVT following the above method with double-precision floating-point arithmetic. We used five pattern matrices, the column size of which was $M=2,3,10,50,100$, and used four matrix sets, the numbers of which were $L=10,100,1000,10000$.

TABLE III
EXECUTION SPEED DURING EXPERIMENT

Time unit [ms]

	SVDSVT [3]	FSVT [13]		FRSVT [14]		FPSVT-np		FPSVT-p	
(M,L)	Time	Time	Ratio	Time	Ratio	Time	Ratio	Time	Ratio
$(2,10)$	0.185	3.905	×0.05	0.678	×0.27	0.082	×2.26	**0.054**	×**3.40**
$(2,100)$	1.829	40.518	×0.05	6.552	×0.28	0.743	×2.46	**0.067**	×**27.45**
$(2,1000)$	18.318	393.040	×0.05	65.562	×0.28	7.193	×2.55	**0.194**	×**94.62**
$(2,10000)$	180.640	3951.800	×0.05	663.890	×0.27	73.568	×2.46	**1.136**	×**159.07**
$(3,10)$	0.210	4.103	×0.05	0.649	×0.32	0.106	×1.99	**0.063**	×**3.32**
$(3,100)$	2.084	40.191	×0.05	6.464	×0.32	1.028	×2.03	**0.078**	×**26.65**
$(3,1000)$	20.735	405.970	×0.05	65.216	×0.32	10.270	×2.02	**0.241**	×**86.06**
$(3,10000)$	213.470	4172.700	×0.05	645.860	×0.33	103.470	×2.06	**1.311**	×**162.84**
$(10,10)$	0.214	4.229	×0.05	0.647	×0.33	0.107	×2.00	**0.064**	×**3.34**
$(10,100)$	2.135	42.670	×0.05	6.468	×0.33	1.056	×2.02	**0.089**	×**24.10**
$(10,1000)$	21.204	424.680	×0.05	64.944	×0.33	10.381	×2.04	**0.429**	×**49.43**
$(10,10000)$	212.420	4149.900	×0.05	652.760	×0.33	105.460	×2.01	**2.326**	×**91.31**
$(50,10)$	0.257	4.316	×0.06	0.671	×0.38	0.116	×2.22	**0.073**	×**3.50**
$(50,100)$	2.383	42.675	×0.06	6.689	×0.36	1.126	×2.12	**0.145**	×**16.49**
$(50,1000)$	23.781	429.580	×0.06	66.921	×0.36	10.853	×2.19	**0.823**	×**28.91**
$(50,10000)$	243.230	4288.100	×0.06	681.150	×0.36	111.760	×2.18	**15.623**	×**15.57**
$(100,10)$	0.406	5.580	×0.07	0.896	×0.45	0.126	×3.21	**0.083**	×**4.88**
$(100,100)$	3.963	51.332	×0.08	8.887	×0.45	1.159	×3.42	**0.273**	×**14.50**
$(100,1000)$	38.920	509.600	×0.08	91.022	×0.43	11.750	×3.31	**1.614**	×**24.12**
$(100,10000)$	384.550	5129.600	×0.07	911.460	×0.42	117.950	×3.26	**30.042**	×**12.80**

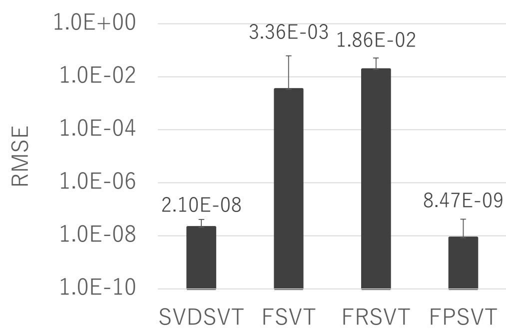

Fig. 2. Calculation accuracy during experiment
Error bars show standard deviation

2) Results: Table III shows the execution time required to calculate SVT with each method. We executed each calculation by double-precision arithmetic. Under all conditions, FPSVT-p was the fastest, which was up to 162.84 times faster than SVDSVT. Since the effect of computational complexity reduction of FPSVT-np was about 1.99–3.42 times faster than SVDSVT, we confirmed that data parallelization was remarkably high.

Next, we explain calculation accuracy using Fig. 2. Figure 2 shows the root-mean-square error (RMSE) between the output results of each SVT-calculation method executed in single-precision and the output test matrices. For FRSVT, the result of double-precision operation is shown because Oh's code cannot be used for single-precision calculations. Since FPSVT-np and FPSVT-p had the same calculation results, they are collectively described as FPSVT. FPSVT had about 60% less calculation error compared with SVDSVT, and the average RMSE was 8.47×10^{-9}. We presume that FPSVT has fewer floating-point operations than SVDSVT and suppresses the expansion of calculation errors.

From these results, we confirm that FPSVT-p has excellent calculation speed and accuracy.

V. Conclusion

We proposed the fast-parallel singular-value-thresholding calculation method (FPSVT) for finding a solution of an optimization problem that regularizes many small matrices at low-rank. On the basis of the discovery that the nuclear norm can be represented by the Euclidean distance in a subspace, we derived two SVT algorithms with FPSVT without the need for singular value decomposition. Through an experiment, we confirmed that FPSVT had higher calculation accuracy than the conventional methods, and its calculation speed increased up to 162.84 times.

Appendix

A. Proof of Proposition 1

Proof: Let $\lambda_1, \lambda_2 (\lambda_1 \geq \lambda_2 \geq 0)$ be the eigenvalue of the matrix $\mathbf{Y}^{\mathrm{T}}\mathbf{Y}$. From $\lambda_1 = \sigma_1^2, \lambda_2 = \sigma_2^2, \lambda_1 + \lambda_2 = \mathrm{tr}\mathbf{Y}^{\mathrm{T}}\mathbf{Y} = \|\mathbf{y}_1\|^2 + \|\mathbf{y}_2\|^2, \lambda_1\lambda_2 = \det \mathbf{Y}^{\mathrm{T}}\mathbf{Y} = \|\mathbf{y}_1\|^2\|\mathbf{y}_2\|^2 - (\mathbf{y}_1^{\mathrm{T}}\mathbf{y}_2)^2$,

$$
\begin{aligned}
\sigma_1 \pm \sigma_2 &= \sqrt{\sigma_1^2 + \sigma_2^2 \pm 2\sigma_1\sigma_2} = \sqrt{\lambda_1 + \lambda_2 \pm 2\sqrt{\lambda_1\lambda_2}} \\
&= \sqrt{\mathrm{tr}\mathbf{Y}^{\mathrm{T}}\mathbf{Y} \pm 2\sqrt{\det \mathbf{Y}^{\mathrm{T}}\mathbf{Y}}} \\
&= \sqrt{\|\mathbf{y}_1\|^2 + \|\mathbf{y}_2\|^2 \pm \sqrt{\|\mathbf{y}_1\|^2\|\mathbf{y}_2\|^2 - (\mathbf{y}_1^{\mathrm{T}}\mathbf{y}_2)^2}} \qquad \text{(A.1)}
\end{aligned}
$$

holds. Considering $\mathrm{Im}\mathbf{Y}$, it is at most a two-dimensional subspace, so $\mathrm{Im}\mathbf{Y}$ is embedded, as shown in Figure 1. We focus on the geometric properties of the vectors $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^M$. From $\mathbf{y}_1^{\mathrm{T}}\mathbf{y}_2 = \|\mathbf{y}_1\|\|\mathbf{y}_2\|\cos\theta$,

$$
\text{(A.1)} = \sqrt{\|\mathbf{y}_1\|^2 + \|\mathbf{y}_2\|^2 \pm 2\|\mathbf{y}_1\|\|\mathbf{y}_2\|\sin\theta} \qquad \text{(A.2)}
$$

holds. Also, since the angle between $\mathbf{y}_1$ and $\mathbf{R}\mathbf{y}_2$ is $\theta + \pi/2$ [rad],

$$
\begin{aligned}
\|\mathbf{y}_1\|\|\mathbf{y}_2\|\sin\theta &= -\|\mathbf{y}_1\|\|\mathbf{R}\mathbf{y}_2\|\cos(\theta + \pi/2) \\
&= -\mathbf{y}_1^{\mathrm{T}}\mathbf{R}\mathbf{y}_2 \qquad \text{(A.3)}
\end{aligned}
$$

holds. Therefore,

$$
\begin{aligned}
\text{(A.2)} &= \sqrt{\|\mathbf{y}_1\|^2 + \|\mathbf{y}_2\|^2 \mp 2\mathbf{y}_1^{\mathrm{T}}\mathbf{R}\mathbf{y}_2} \\
&= \|\mathbf{y}_1 \mp \mathbf{R}\mathbf{y}_2\| \qquad \text{(A.4)}
\end{aligned}
$$

holds. Therefore, the subject was satisfied. ■

B. Proof of Proposition 3

Before proof of proposition 3, we show the following three lemmas.

Lemma 7 For any $\mathbf{x} \in (\mathrm{Im}\mathbf{Y})^2$, the matrix $\mathbf{A}$ in (11) holds for

$$
\mathbf{A}\mathbf{A}^{\mathrm{T}}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} = 2\mathbf{x}. \qquad \text{(B.1)}
$$

Proof: By setting $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}, \mathbf{x}_1, \mathbf{x}_2 \in \mathrm{Im}\mathbf{Y}$, from the property (9),

$$
\mathbf{A}\mathbf{A}^{\mathrm{T}}\mathbf{x} = \begin{bmatrix} \mathbf{I} + \mathbf{R}\mathbf{R}^{\mathrm{T}} & \mathbf{I} - \mathbf{R}\mathbf{R}^{\mathrm{T}} \\ \mathbf{I} - \mathbf{R}\mathbf{R}^{\mathrm{T}} & \mathbf{I} + \mathbf{R}\mathbf{R}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = 2\mathbf{x} \qquad \text{(B.2)}
$$

$$
\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \begin{bmatrix} 2\mathbf{I} & \mathbf{O} \\ \mathbf{O} & 2\mathbf{R}^{\mathrm{T}}\mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = 2\mathbf{x} \qquad \text{(B.3)}
$$

holds. ■

Lemma 8 Let $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^M$ be $[\mathbf{z}_1, \mathbf{z}_2] = \mathrm{prox}_{\mu\|\cdot\|_*}(\mathbf{Y})$. Then $\mathbf{z}_1, \mathbf{z}_2 \in \mathrm{Im}\mathbf{Y}$ holds.

Proof: Let $\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$ be the SVD,

$$
\begin{aligned}
\mathbf{U} &= [\mathbf{u}_1, \mathbf{u}_2], \mathbf{u}_i \in \mathbb{R}^M \\
\mathbf{\Sigma} &= \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}, \sigma_i \in \mathbb{R} \\
\mathbf{V} &= \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}, v_{ij} \in \mathbb{R}
\end{aligned}
$$

holds. From $\mathbf{y}_i = \sigma_1 v_{i1}\mathbf{u}_1 + \sigma_2 v_{i2}\mathbf{u}_2$,

$$
\mathrm{Im}\mathbf{Y} = \begin{cases} \mathrm{Im}\mathbf{U} & \textit{if } \mathrm{rank}\mathbf{Y} = 2 \\ \mathrm{Im}\mathbf{u}_1 & \textit{if } \mathrm{rank}\mathbf{Y} = 1 \\ \{0\} & \textit{if } \mathrm{rank}\mathbf{Y} = 0 \end{cases} \qquad \text{(B.4)}
$$

holds. Similarly, from the SVT definition (5), $\mathbf{z}_i = (\sigma_1 - \mu)_+ v_{i1}\mathbf{u}_1 + (\sigma_2 - \mu)_+ v_{i2}\mathbf{u}_2$ holds. Then, $\mathbf{z}_i \in \mathrm{Im}\mathbf{Y}$ can be confirmed. ■

Lemma 9 Let $\mathcal{X}$ be a real Hilbert space, let $\mathcal{S}$ be a subspace of $\mathcal{X}$, let $f : \mathcal{X} \to \mathbb{R}$ and let $\mathbf{L} : \mathcal{X} \to \mathcal{X}$ be a linear operator. Suppose that the composition of $\mathbf{L}$ and $\mathbf{L}^{\mathrm{T}}$ satisfies $\mathbf{L}^{\mathrm{T}}\mathbf{L}\mathbf{x} = \mathbf{L}\mathbf{L}^{\mathrm{T}}\mathbf{x} = \alpha\mathbf{x}$, for any $\mathbf{x} \in \mathcal{S}$ some $\alpha > 0$ and the proximal operator $\mathrm{prox}_f(\mathbf{x}) \in \mathcal{S}$ for any $\mathbf{x} \in \mathcal{S}$. Then

$$
\mathrm{prox}_{f(\mathbf{L}(\cdot))}(\mathbf{x}) = \frac{1}{\alpha}\mathbf{L}^{\mathrm{T}}\mathrm{prox}_f(\mathbf{L}\mathbf{x}). \qquad \text{(B.5)}
$$

Proof:

$$
\begin{aligned}
\mathrm{prox}_{f(\mathbf{L}(\cdot))}(\mathbf{x}) &= \arg\min_{\mathbf{z}\in\mathcal{S}} f(\mathbf{L}\mathbf{z}) + \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|^2 \\
&= \arg\min_{\mathbf{z}\in\mathcal{S}} f(\mathbf{L}\mathbf{z}) + \frac{1}{2\alpha}(\mathbf{x} - \mathbf{z})^{\mathrm{T}}\mathbf{L}^{\mathrm{T}}\mathbf{L}(\mathbf{x} - \mathbf{z}) \\
&= \arg\min_{\mathbf{z}\in\mathcal{S}} f(\mathbf{L}\mathbf{z}) + \frac{1}{2\alpha}\|\mathbf{L}\mathbf{x} - \mathbf{L}\mathbf{z}\|^2 \\
&= \frac{1}{\alpha}\mathbf{L}^{\mathrm{T}} \arg\min_{\mathbf{y}\in\mathcal{S}} f(\mathbf{y}) + \frac{1}{2\alpha}\|\mathbf{L}\mathbf{x} - \mathbf{y}\|^2 \\
&= \frac{1}{\alpha}\mathbf{L}^{\mathrm{T}}\mathrm{prox}_f(\mathbf{L}\mathbf{x}). \qquad \text{(B.6)}
\end{aligned}
$$

■

On the basis of the above lemmas, we prove Proposition 3.

Proof: From Corollary 2 and definition of proximal mapping (1),

$$
\begin{aligned}
\mathrm{prox}_{\mu\|\cdot\|_*}(\mathbf{Y}) &= \mathrm{prox}_{\mu\|\mathbf{A}\mathrm{vec}(\cdot)\|_{\infty,2}}(\mathbf{Y}) \\
&= \frac{1}{2}\mathrm{vec}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathrm{prox}_{2\mu\|\cdot\|_{\infty,2}}(\mathbf{A}\mathrm{vec}\mathbf{Y}) \\
&= \mathrm{vec}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathrm{prox}_{\mu\|\cdot\|_{\infty,2}}\left(\frac{1}{2}\mathbf{A}\mathrm{vec}\mathbf{Y}\right) \qquad \text{(B.7)}
\end{aligned}
$$

holds using Lemmas 7, 8 and 9. This satisfies the subject. ■

C. Proof of Proposition 4

Proof: Since the dual norm of $L_{\infty,2}$ norm is $L_{1,2}$ norm, from Moreau decomposition [4] [16],

$$\mathrm{prox}_{\mu\|\cdot\|_{\infty,2}}(\mathbf{x}) = \mathbf{x} - \mathrm{proj}_{\mu B_{1,2}}(\mathbf{x}) \tag{C.1}$$

holds, where $\mathrm{proj}_{\mu B_{1,2}}(\mathbf{x})$ is the Euclidean distance projection onto the $L_{1,2}$ sphere $\mu B_{1,2}$ and is the solution to the following optimization problem.

$$\begin{aligned}\mathrm{proj}_{\mu B_{1,2}}(\mathbf{x}) &= \arg\min_{\mathbf{z}\in\mu B_{1,2}} \|\mathbf{x}-\mathbf{z}\|^2 \\ \mu B_{1,2} &= \{\mathbf{z}=\begin{bmatrix}\mathbf{z}_1\\ \mathbf{z}_2\end{bmatrix} \mid \|\mathbf{z}_1\|+\|\mathbf{z}_2\|\le\mu\}\end{aligned} \tag{C.2}$$

According to Van den Berg et al. [17], problem (C.2) can be separated into L_1 and L_2 sphere projection problems. In $\mathbf{x} = \begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix}$, this is equivalent to the two problems of

$$\mathrm{proj}_{\mu B_{1,2}}(\mathbf{x}) = \begin{bmatrix}\mathrm{proj}_{\eta_1 B_2}(\mathbf{x}_1)\\ \mathrm{proj}_{\eta_2 B_2}(\mathbf{x}_2)\end{bmatrix} \tag{C.3}$$

$$\begin{bmatrix}\eta_1\\ \eta_2\end{bmatrix} = \mathrm{proj}_{\mu B_1}\left(\begin{bmatrix}\|\mathbf{x}_1\|\\ \|\mathbf{x}_2\|\end{bmatrix}\right). \tag{C.4}$$

Problem (C.3) can be calculated by $\mathrm{proj}_{\eta_i B_2}(\mathbf{x}_i) = (\eta_i/\|\mathbf{x}_i\|)\mathbf{x}_i$. However, problem (C.4) can be calculated by

$$(\mathrm{C.4}) = \begin{cases}\begin{bmatrix}\|\mathbf{x}_1\| - \frac{(\|\mathbf{x}_1\|+\|\mathbf{x}_2\|-\mu)_+}{2}\\ \|\mathbf{x}_2\| - \frac{(\|\mathbf{x}_1\|+\|\mathbf{x}_2\|-\mu)_+}{2}\end{bmatrix} & \mathit{if}\ |\|\mathbf{x}_1\|-\|\mathbf{x}_2\||\le\mu \\ \begin{bmatrix}\mu\\ 0\end{bmatrix} & \mathit{if}\ \|\mathbf{x}_1\|-\|\mathbf{x}_2\|>\mu \\ \begin{bmatrix}0\\ \mu\end{bmatrix} & \mathit{if}\ \|\mathbf{x}_1\|-\|\mathbf{x}_2\|<-\mu\end{cases} \tag{C.5}$$

The subject is obtained by expanding the above into an expression (C.1). ■

D. Proof of Theorem 6

Proof: Let $\mathbf{Z} = \mathrm{prox}_{\mu\|\cdot\|_*}(\mathbf{Y})$. For $\mathrm{rank}\mathbf{Y} = 0$, it is obvious because of $\mathbf{Y} = \mathbf{O}$. In the following, the SVD is $\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$ and

$$\begin{aligned}\mathbf{U} &= [\mathbf{u}_1, \mathbf{u}_2], \mathbf{u}_i \in \mathbb{R}^M \\ \mathbf{\Sigma} &= \begin{bmatrix}\sigma_1 & 0\\ 0 & \sigma_2\end{bmatrix}, \sigma_i \in \mathbb{R} \\ \mathbf{V} &= [\mathbf{v}_1, \mathbf{v}_2], \mathbf{v}_i \in \mathbb{R}^2.\end{aligned}$$

For $\mathrm{rank}\mathbf{Y}=1$,

$$\begin{aligned}\mathbf{Z} &= \mathbf{U}\begin{bmatrix}(\sigma_1-\mu)_+ & 0\\ 0 & 0\end{bmatrix}\mathbf{V}^{\mathrm{T}} \\ &= \frac{(\sigma_1-\mu)_+}{\sigma_1}\mathbf{U}\begin{bmatrix}\sigma_1 & 0\\ 0 & 0\end{bmatrix}\mathbf{V}^{\mathrm{T}} = \left(1-\frac{\mu}{\sigma_1}\right)_+\mathbf{Y}\end{aligned} \tag{D.1}$$

holds because $\sigma_1>0, \sigma_2=0$. Since

$$\begin{aligned}\|\mathbf{Y}\| &= \sqrt{\mathrm{tr}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}} = \sqrt{\mathrm{tr}(\sigma_1\mathbf{u}_1\mathbf{v}_1^{\mathrm{T}})^{\mathrm{T}}(\sigma_1\mathbf{u}_1\mathbf{v}_1^{\mathrm{T}})} \\ &= \sigma_1,\end{aligned} \tag{D.2}$$

this satisfies the subject. For $\mathrm{rank}\mathbf{Y}=2$ and $\sigma_1=\sigma_2$,

$$\begin{aligned}\mathbf{Z} &= \mathbf{U}\begin{bmatrix}(\sigma_1-\mu)_+ & 0\\ 0 & (\sigma_1-\mu)_+\end{bmatrix}\mathbf{V}^{\mathrm{T}} \\ &= \frac{(\sigma_1-\mu)_+}{\sigma_1}\mathbf{U}\begin{bmatrix}\sigma_1 & 0\\ 0 & \sigma_1\end{bmatrix}\mathbf{V}^{\mathrm{T}} = \left(1-\frac{\mu}{\sigma_1}\right)_+\mathbf{Y}\end{aligned} \tag{D.3}$$

holds. Since

$$\begin{aligned}\|\mathbf{Y}\| &= \sqrt{\mathrm{tr}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}} = \sqrt{\mathrm{tr}(\sigma_1\mathbf{U}\mathbf{V}^{\mathrm{T}})^{\mathrm{T}}(\sigma_1\mathbf{U}\mathbf{V}^{\mathrm{T}})} \\ &= \sqrt{2}\sigma_1,\end{aligned} \tag{D.4}$$

this satisfies the subject. ■

[4] For a convex function $f(\mathbf{x})$ and its Legendre transformation $f^*(\mathbf{x})$, $\mathbf{x} = \mathrm{prox}_f(\mathbf{x}) + \mathrm{prox}_{f^*}(\mathbf{x})$ is called Moreau decomposition. Especially when the convex function is norm $f(\mathbf{x}) = \mu\|\mathbf{x}\|$, using $\mathrm{proj}_{\mu B_d}(\mathbf{x})$ to project the dual norm $\|\mathbf{x}\|_d$ to sphere μB_d, $\mathbf{x} = \mathrm{prox}_{\mu\|\cdot\|}(\mathbf{x}) + \mathrm{proj}_{\mu B_d}(\mathbf{x})$ holds.

References

[1] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Advances in neural information processing systems*, 2009, pp. 2080–2088.

[2] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 11:1–11:37, 2011.

[3] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[4] S. Gandy and I. Yamada, "Convex optimization techniques for the efficient recovery of a sparsely corrupted low-rank matrix," *Journal of Math-for-Industry*, vol. 2, no. 2010B-5, pp. 147–156, 2010.

[5] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 3, pp. 597–610, 2012.

[6] W. Dong, G. Shi, X. Hu, and Y. Ma, "Nonlocal sparse and low-rank regularization for optical flow estimation," *IEEE Transactions on Image Processing*, vol. 23, no. 10, pp. 4527–4538, 2014.

[7] R. Otazo, E. Candes, and D. K. Sodickson, "Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components," *Magnetic Resonance in Medicine*, vol. 73, no. 3, pp. 1125–1136, 2015.

[8] S. Ono and I. Yamada, "Color-line regularization for color artifact removal," *IEEE Transactions on Computational Imaging*, vol. 2, no. 3, pp. 204–217, 2016.

[9] G. Ye, M. Tang, J.-F. Cai, Q. Nie, and X. Xie, "Low-rank regularization for learning gene expression programs," *PloS one*, vol. 8, no. 12, p. e82146, 2013.

[10] T. Sasaki, R. Tanida, and A. Shimizu, "Graph shape simplification based on local linear approximation of graph signal," in *Forum on Information Technology 2016*, 2016, pp. 3:1–3:4, (in Japanese).

[11] M. Fazel, "Matrix rank minimization with applications," Ph.D. dissertation, Stanford University, 2002.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[13] J.-F. Cai and S. Osher, "Fast singular value thresholding without singular value decomposition," *Methods and Applications of Analysis*, vol. 20, no. 4, pp. 335–352, 2013.

[14] T.-H. Oh, Y. Matsushita, Y.-W. Tai, and I. So Kweon, "Fast randomized singular value thresholding for nuclear norm minimization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4484–4493.

[15] J. Moreau, "Fonctions convexes duales et points proximaux dans un espace hilbertien," *Comptes Rendus de l'Académie des Sciences*, vol. 225, pp. 2897–2899, 1962.

[16] ——, "Décomposition orthogonale d'un espace hilbertien selon deux cones mutuellement polaires," *Comptes Rendus de l'Académie des Sciences*, vol. 225, pp. 238–240, 1962.
[17] E. van den Berg, M. Schmidt, M. P. Friedlander, and K. Murphy, "Group sparsity via linear-time projection," Department of Computer Science, University of British Columbia, Vancouver, BC, Canada, Tech. Rep. TR-2008-09, 2008.