IMAGE COLORIZATION BASED ON ADMM WITH FAST SINGULAR VALUE THRESHOLDING BY CHEBYSHEV POLYNOMIAL APPROXIMATION

Masaki Onuki[†]*, Shunsuke Ono*[‡]*, Keiichiro Shirai*[#]*, and Yuichi Tanaka*[†]

[†]Grad. School of BASE, Tokyo Univ. of Agri. and Tech., Koganei, Tokyo, 184-8588 Japan
[‡]Imaging Sci. and Eng. Lab., Tokyo Inst. of Tech., Midori-ku, Kanagawa, 226-8503 Japan
[#]Dept. of Comput. Sci. and Eng., Shinshu Univ., Wakasato, Nagano, 380-8553 Japan
Email: masaki.o@msp-lab.org, ono@isl.titech.ac.jp, keiichi@shinshu-u.ac.jp, ytnk@cc.tuat.ac.jp

ABSTRACT

We propose an image colorization method using fast soft-thresholding of singular values (singular value thresholding). An image colorization method with nuclear norm minimization (NNM) has been proposed and brings good results. NNM usually requires iterative application of singular value decomposition (SVD) for singular value thresholding. However, the computational cost of SVD in the colorization method becomes too expensive to handle high-resolution images. In this paper, we reduce its computational cost by using Chebyshev polynomial approximation (CPA). Singular value thresholding is expressed by a multiplication of certain matrices derived from the characteristic of CPA. As a result, our CPA-based technique makes the image colorization method much more efficient. In addition, we replace the optimization method used in the image colorization method by alternating direction method of multipliers, which further accelerates the computation. Experimental results verify the effectiveness of our method with respect to the computation time and the approximation precision.

Index Terms— Chebyshev polynomial approximation, singular value thresholding, image colorization

1. INTRODUCTION

Image colorization, assigning suitable colors to grayscale images, is an important task in image processing and computer vision. Many image colorization methods have been proposed [1–9] and successfully applied not only to art restoration [10, 11] but also to image compression [12,13]. Among them, recent work [9] revealed that the state-of-the-art performance of image colorization can be achieved with the help of nuclear norm minimization (NNM).

Specifically, in the image colorization method [9], the RGB components of an image are rearranged to a certain matrix, and the matrix is assumed to be low rank. Since the nuclear norm is the tightest convex relaxation of the rank function, the method adopts the nuclear norm of the matrix as a regularization for colorization. Therefore, they colorize images by minimizing objective functions involving the nuclear norm.

Usually, NNM requires iterative application of singular value decomposition (SVD) to shrink singular values. For example, the method with NNM [9] adopts the so-called *augmented Lagrangian method* (ALM) [14], which calls SVD about 50 times to obtain a colorized image. However, it is clear that the iterative SVDs need much computation time in the case of high-resolution images.

This work was supported in part by MEXT Tenure-Track Promotion Program.

The shrinkage of *eigenvalues* similarly requires enormous computation time as well as that of singular values. It is applied to improve the smoothing performance of image filters, e.g., bilateral filter [15–17]. It is realized by filtering eigenvalues of its filter matrix (eigenvalue filtering) [18–20]. Unfortunately, eigendecomposition (ED) needs much computation time for handling a large matrix. To overcome the difficulty, we proposed an approach using *Chebyshev polynomial approximation* (CPA) [21–24] in order to accelerate the eigenvalue filtering [25, 26]. CPA is a traditional and an effective function approximation method in the field of signal processing. CPA boils down the eigenvalue filtering to a multiplication of a filter matrix and an image vector, which significantly accelerates the eigenvalue filtering.

In this paper, we extend our previously proposed method to soft-thresholding of singular values (singular value thresholding) for image colorization. CPA is applied to our singular value thresholding, i.e., our method is represented as a multiplication of matrices, which does not require singular values/vectors explicitly. As a result, the computation time of our method becomes much less than that of the exact method [9]. In addition, we also apply the so-called *alternating direction method of multipliers* (ADMM) [27, 28] instead of ALM used in [9], which further accelerates the computation of the image colorization. Experimental results show that our method reduces the computation time up to 30% of that of the exact method while maintaining the accuracy of colorization.

This paper is organized as follows. Section 2 describes some notations and definitions. Section 3 presents our fast singular value thresholding. Firstly, CPA is briefly reviewed and then is extended to singular value thresholding. We apply the fast thresholding method to image colorization with ADMM in Section 4, and Section 5 shows some experimental results. Finally, Section 6 concludes the paper.

2. NOTATIONS AND PRELIMINARIES

2.1. Notations

Upper case bold-face and capital letters indicate a matrix and a vector, respectively. Superscript $\cdot^\top$ is the transpose of a matrix and a vector, and superscript $\cdot^{-1}$ is the inverse of a matrix. The matrix $\mathbf{I}$ is the identity matrix. The ℓ_p norm for $p \geq 1$ is defined as $\|\mathbf{x}\|_p = (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$ $(\forall \mathbf{x} \in \mathbb{R}^n)$. The operator $\mathrm{rank}(\cdot)$ computes the rank of a matrix. For a given matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, its i-th largest singular value is denoted by $\sigma_i(\mathbf{X})$, where $i \in \{1, 2, \ldots, K\}$ $(K := \min(m, n))$, and the nuclear norm of $\mathbf{X}$ is defined as

$$\|\mathbf{X}\|_* := \textstyle\sum_{i=1}^K \sigma_i(\mathbf{X}). \quad (1)$$

We also use some mathematical tools introduced as follows.

2.2. Proximity Operator [29]

Let $\mathbf{\Gamma}_0(\mathbb{R}^N)$ be the set of all proper lower semicontinuous convex[1] functions over $\mathbb{R}^N$. The *proximity operator* of a function $f \in \mathbf{\Gamma}_0(\mathbb{R}^N)$ of index $\gamma > 0$ is defined as follows:

$$\text{prox}_{\gamma f} : \mathbb{R}^N \to \mathbb{R}^N : \mathbf{x} \mapsto \underset{\mathbf{y}\in\mathbb{R}^N}{\arg\min}\ f(\mathbf{y}) + \tfrac{1}{2\gamma}\|\mathbf{x}-\mathbf{y}\|^2. \quad (2)$$

The proximity operator will play a central role in the optimization of image colorization in Sec. 4.

2.3. Alternating Direction Method of Multipliers [27, 28]

ADMM is an algorithm for solving a convex optimization problem represented as

$$\min_{\mathbf{x}\in\mathbb{R}^{n_1},\mathbf{y}\in\mathbb{R}^{n_2}} g_1(\mathbf{x}) + g_2(\mathbf{y}) \ \text{ s.t. } \mathbf{y} = \mathbf{K}\mathbf{x}, \quad (3)$$

where $g_1 \in \mathbf{\Gamma}_0(\mathbb{R}^{n_1})$, $g_2 \in \mathbf{\Gamma}_0(\mathbb{R}^{n_2})$ and $\mathbf{K} \in \mathbb{R}^{n_2\times n_1}$. For arbitrary $\mathbf{y}_0$, $\mathbf{p}_0 \in \mathbb{R}^{n_2}$, and $\gamma > 0$, ADMM algorithm is given by

$$\begin{cases} \mathbf{x}_{n+1} := \arg\min_{\mathbf{x}} g_1(\mathbf{x}) + \frac{\gamma}{2}\|\mathbf{y}_n - \mathbf{K}\mathbf{x} - \mathbf{p}_n\|_2^2 \\ \mathbf{y}_{n+1} := \text{prox}_{\frac{1}{\gamma}g_2}(\mathbf{K}\mathbf{x}_{n+1} + \mathbf{p}_n) \\ \mathbf{p}_{n+1} := \mathbf{p}_n + \mathbf{K}\mathbf{x}_{n+1} - \mathbf{y}_{n+1}. \end{cases} \quad (4)$$

We recall a convergence analysis of ADMM by Eskstein-Bertsekas [28].

Fact 1 (Convergence of ADMM [28]) *Consider Prob.* (3). *Assume that* $\mathbf{K}^\top\mathbf{K}$ *is invertible and that a saddle point of its unaugmented Lagrangian* $L_0(\mathbf{x},\mathbf{y},\mathbf{z}) := g_1(\mathbf{x}) + g_2(\mathbf{y}) - \langle \mathbf{z}, \mathbf{K}\mathbf{x} - \mathbf{y}\rangle$ *exists. Then the sequence* $(\mathbf{x}_n)_{(n\geq 1)}$ *generated by* (4) *converges to a solution of Prob.* (3).

3. SINGULAR VALUE THRESHOLDING USING CHEBYSHEV POLYNOMIAL APPROXIMATION

3.1. Chebyshev Polynomial Approximation for Scalar Functions

Let $h(y)$ and $h^*(y)$ be a real-valued function defined on the interval $y \in [-1, 1]$ and the function approximated by using CPA, respectively. CPA [21–24] gives an approximation of $h(y)$ by using its truncation:

$$h^*(y) := \tfrac{1}{2}c_0 + \textstyle\sum_{k=1}^{\alpha-1} c_k T_k(y), \quad (5)$$

where c_k and α denote a Chebyshev coefficient described later and an approximation order, respectively. Additionally, $T_k(\cdot)$ denotes the k-th order Chebyshev polynomial of first kind defined as

$$T_k(y) := \cos(k \arccos(y)). \quad (6)$$

It can also be computed by using the stable recurrence relation:

$$\begin{aligned} &T_k(y) = 2yT_{k-1}(y) - T_{k-2}(y), \\ &T_0(y) = 1, \quad T_1(y) = y. \end{aligned} \quad (7)$$

[1] A function $f : \mathbb{R}^N \to \mathbb{R}\cup\{\infty\}$ is called *proper lower semicontinuous convex* if $\text{dom}(f) := \{\mathbf{x}\in\mathbb{R}^N |\ f(\mathbf{x}) < \infty\} \neq \emptyset$, $\text{lev}_{\leq a}(f) := \{\mathbf{x}\in\mathbb{R}^N |\ f(\mathbf{x}) \leq a\}$ is closed in $\forall a \in \mathbb{R}$, and $f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda)f(\mathbf{y})$ in $\forall \mathbf{x},\mathbf{y}\in\mathbb{R}^N$ and $\forall\lambda\in(0,1)$, respectively.

The initial condition is defined by using $T_0(y)$ and $T_1(y)$. Since the polynomial consists of cosine functions, the value of $T_k(y)$ is bounded between -1 and 1 for $y \in [-1, 1]$. By using $T_k(y)$ and the orthogonality of sine waves, the Chebyshev coefficient c_k is calculated as

$$c_k := \tfrac{2}{\pi}\textstyle\int_{-1}^{1} \frac{T_k(y)h(y)}{\sqrt{1-y^2}}dy = \frac{2}{\pi}\int_0^\pi \cos(k\theta)h(\cos\theta)d\theta. \quad (8)$$

Practically, c_k is discretely calculated as in [23].

3.2. Chebyshev Polynomial Approximation for Eigenvalue Filtering Method

Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a symmetric full rank matrix, and $\mathbf{A} = \mathbf{X_A}\mathbf{D_A}\mathbf{X_A}^\top$ be its ED, where $\mathbf{X_A} \in \mathbb{R}^{n\times n}$ is the matrix composed of eigenvectors, and $\mathbf{D_A} = \text{diag}(d_1^\mathbf{A}, \ldots, d_i^\mathbf{A}, \ldots, d_n^\mathbf{A})$ is the diagonal matrix with the corresponding eigenvalues. We assume that the eigenvalues are bounded between -1 and 1, i.e., $d_i^\mathbf{A} \in [-1, 1]$, and this allows us to apply CPA to the eigenvalues. CPA of the matrix form gives an approximation of an eigenvalue filter function whose truncated Chebyshev series $\mathcal{H}^*(\mathbf{A})$ is defined as

$$\mathcal{H}^*(\mathbf{A}) := \tfrac{1}{2}c_0\mathbf{I} + \textstyle\sum_{k=1}^{\alpha-1} c_k\mathcal{T}_k(\mathbf{A}), \quad (9)$$

where the k-th order polynomial of the matrix form is defined as

$$\mathcal{T}_k(\mathbf{A}) := \mathbf{X_A}\text{diag}(\cos k\theta_1, \ldots, \cos k\theta_i, \ldots, \cos k\theta_n)\mathbf{X_A}^\top. \quad (10)$$

Similarly to (7), its Chebyshev polynomials are obtained by using the recurrence relation:

$$\begin{aligned} &\mathcal{T}_k(\mathbf{A}) = 2\mathbf{A}\mathcal{T}_{k-1}(\mathbf{A}) - \mathcal{T}_{k-2}(\mathbf{A}), \\ &\mathcal{T}_0(\mathbf{A}) = \mathbf{I}, \quad \mathcal{T}_1(\mathbf{A}) = \mathbf{A}. \end{aligned} \quad (11)$$

Furthermore, the eigenvalue filter function $\mathcal{H}^*(\mathbf{A})$ can also be represented with $h^*(d_i^\mathbf{A})$ as

$$\mathcal{H}^*(\mathbf{A}) = \mathbf{X_A}\text{diag}(h^*(d_1^\mathbf{A}), \ldots, h^*(d_i^\mathbf{A}), \ldots, h^*(d_n^\mathbf{A}))\mathbf{X_A}^\top. \quad (12)$$

Since c_k in (9) is calculated as (8), this resulting matrix is an approximation of the original filter $\mathcal{H}(\mathbf{A})$. The derivation realizes eigenvalue filtering without ED. It is usually faster than the exact method based on ED as long as $\mathbf{A}$ is sparse because the dominant computational complexity comes from a multiplication of two matrices.

3.3. Shifted Chebyshev Polynomial Approximation

CPA can only be defined in the interval of eigenvalues bounded -1 and 1. For a matrix whose eigenvalues are outside $[-1, 1]$, its eigenvalue distribution is adjusted by using the method introduced below.

Let $\mathbf{B} \in \mathbb{R}^{n\times n}$ be a full rank and diagonalizable matrix, and $\mathbf{B} = \mathbf{X_B}\mathbf{D_B}\mathbf{X_B}^{-1}$ be its ED. Additionally, we assume that the eigenvalues are bounded between 0 and $d_{\max}^\mathbf{B}$, where $d_{\max}^\mathbf{B} > 1$. The eigenvalues of $\mathbf{D_B}$ are firstly scaled to $[-1, 1]$ for utilizing the recurrence relation as (11). To shift the eigenvalues, the matrix $\widetilde{\mathbf{B}}$ is defined as

$$\widetilde{\mathbf{B}} := \tfrac{2}{d_{\max}^\mathbf{B}}\mathbf{B} - \mathbf{I}. \quad (13)$$

The k-th order polynomial of $\widetilde{\mathbf{B}}$ is defined as

$$\begin{aligned} \mathcal{T}_k(\widetilde{\mathbf{B}}) :=& \mathcal{T}_k\left(\tfrac{2}{d_{\max}^\mathbf{B}}\mathbf{B} - \mathbf{I}\right) \\ =& \mathbf{X_B}\left(\tfrac{2}{d_{\max}^\mathbf{B}}\mathbf{D_B} - \mathbf{I}\right)\mathbf{X_B}^{-1} \\ =& \mathbf{X_B}\text{diag}(\cos k\theta_1, \ldots, \cos k\theta_n)\mathbf{X_B}^{-1}. \end{aligned} \quad (14)$$

Similarly to (11), the Chebyshev polynomials are obtained by using the recurrence relation:

$$\begin{aligned}&\mathcal{T}_k(\widetilde{\mathbf{B}}) = 2\widetilde{\mathbf{B}}\mathcal{T}_{k-1}(\widetilde{\mathbf{B}}) - \mathcal{T}_{k-2}(\widetilde{\mathbf{B}}),\\ &\mathcal{T}_0(\widetilde{\mathbf{B}}) = \mathbf{I}, \quad \mathcal{T}_1(\widetilde{\mathbf{B}}) = \widetilde{\mathbf{B}}.\end{aligned} \tag{15}$$

As a result, CPA of $\mathbf{B}$ gives an approximation of an eigenvalue filter function $\mathcal{H}_{\mathrm{S}}(\mathbf{B})$ whose truncated Chebyshev series is defined as

$$\mathcal{H}_{\mathrm{S}}(\mathbf{B}) := \tfrac{1}{2}\widetilde{c}_0 + \textstyle\sum_{k=1}^{\alpha-1} \widetilde{c}_k \mathcal{T}_k(\widetilde{\mathbf{B}}). \tag{16}$$

Recall that the Chebyshev polynomial $\mathcal{T}_k(\widetilde{\mathbf{B}})$ is limited in $[-1, 1]$. Therefore, the range of eigenvalues is modified by deriving the Chebyshev coefficient $\widetilde{c}_k$ as

$$\widetilde{c}_k = \tfrac{2}{\pi}\textstyle\int_0^{\pi} \cos(k\theta) h\left(\frac{d^{\mathbf{B}}_{\max}}{2}(\cos\theta + 1)\right) d\theta. \tag{17}$$

The term $h\left(\frac{d^{\mathbf{B}}_{\max}}{2}(\cos\theta + 1)\right)$ returns the shifted range back to the original range. From (17), the eigenvalue filter function $\mathcal{H}_{\mathrm{S}}(\mathbf{B})$ can also be represented by using $h^*(d_i^{\mathbf{B}})$ as

$$\mathcal{H}_{\mathrm{S}}(\mathbf{B}) = \mathbf{X}_{\mathbf{B}}\mathrm{diag}(h^*(d_1^{\mathbf{B}}), \ldots, h^*(d_n^{\mathbf{B}}))\mathbf{X}_{\mathbf{B}}^{-1}. \tag{18}$$

3.4. Singular Value Thresholding Using Chebyshev Polynomial Approximation

Let a rectangular matrix be $\mathbf{C} \in \mathbb{R}^{m\times n}$ $(m > n)$ and its SVD be $\mathbf{C} = \mathbf{Y}_1\mathbf{\Sigma}\mathbf{Y}_2^{\top}$, where $\mathbf{Y}_1 \in \mathbb{R}^{m\times m}$ and $\mathbf{Y}_2 \in \mathbb{R}^{n\times n}$ are orthogonal matrices. $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ is the singular value matrix represented as

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \sigma_n \\ & \mathbf{O} & \end{bmatrix}, \tag{19}$$

where $\sigma_1 \geq \ldots \geq \sigma_n$. When the eigenvalue matrix of $\mathbf{C}^{\top}\mathbf{C}$ is defined as $\mathbf{D}_{\mathbf{C}_2} = \mathrm{diag}(d_1^{\mathbf{C}_2}, \ldots, d_i^{\mathbf{C}_2}, \ldots, d_n^{\mathbf{C}_2})$, $\mathbf{C}^{\top}\mathbf{C}$ is decomposed as

$$\mathbf{C}^{\top}\mathbf{C} = \mathbf{Y}_2\mathbf{\Sigma}^{\top}\mathbf{\Sigma}\mathbf{Y}_2^{\top} = \mathbf{Y}_2\mathbf{D}_{\mathbf{C}_2}\mathbf{Y}_2^{\top}, \tag{20}$$

where $\mathbf{D}_{\mathbf{C}_2} = \mathbf{\Sigma}^{\top}\mathbf{\Sigma}$. With the eigenvalue filter function $\mathcal{H}_{\mathrm{S}}(\cdot)$, the eigenvalue filtering of $\mathbf{C}^{\top}\mathbf{C}$ is approximately calculated by using (16) as

$$\mathcal{H}_{\mathrm{S}}(\mathbf{C}^{\top}\mathbf{C}) = \mathbf{Y}_2\mathrm{diag}(h^*(\sigma_1^2), \ldots, h^*(\sigma_n^2))\mathbf{Y}_2^{\top}. \tag{21}$$

To derive the coefficients in (17), a real-valued function $h(\cdot)$ is defined as

$$h(x) = \tfrac{g(x)}{\sqrt{x}}, \tag{22}$$

where $g(\cdot)$ is the desired filter response for $\mathbf{C}$ and x is an input value. Consequently, the singular value thresholding of $\mathbf{C}$ is represented as

$$\begin{aligned}&\mathbf{C}\mathcal{H}_{\mathrm{S}}(\mathbf{C}^{\top}\mathbf{C})\\ &\approx \mathbf{C}\mathbf{Y}_2\mathrm{diag}\left(\tfrac{g(\sigma_1^2)}{\sigma_1}, \ldots, \tfrac{g(\sigma_n^2)}{\sigma_n}\right)\mathbf{Y}_2^{\top}\\ &= \mathbf{Y}_1\begin{bmatrix} g(\sigma_1^2) & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & g(\sigma_n^2) \\ & \mathbf{O} & \end{bmatrix}\mathbf{Y}_2^{\top}.\end{aligned} \tag{23}$$

As can be seen in (23), the singular value thresholding of $\mathbf{C}$ is calculated by the truncated eigenvalue filter $\mathcal{H}_{\mathrm{S}}(\mathbf{C}^{\top}\mathbf{C})$.

4. APPLICATION TO IMAGE COLORIZATION

The image colorization method [9] consists of the local-color consistency method [2] and the optimization method based on NNM. In this section, the singular value thresholding using CPA is applied to the latter optimization part.

Let $\mathbf{R}_{\mathrm{c}}$, $\mathbf{G}_{\mathrm{c}}$, $\mathbf{B}_{\mathrm{c}} \in \mathbb{R}^{m\times n}$ be three color components of an image. The components are arranged into $\mathbf{L} := [\mathbf{R}_{\mathrm{c}}\ \mathbf{G}_{\mathrm{c}}\ \mathbf{B}_{\mathrm{c}}] \in \mathbb{R}^{m\times 3n}$. Given color labels are denoted by $\mathbf{Q} \in \mathbb{R}^{m\times 3n}$ (entries without color labels are set to 0), and some additive error is represented by $\mathbf{S} \in \mathbb{R}^{m\times 3n}$. An observed luminance image is defined as $\mathbf{W} \in \mathbb{R}^{m\times n}$. The linear operator extracting pixels with assigned color labels is denoted as $P_{\Omega}(\cdot)$. Then, the image colorization method [9] yields a color image by solving the following convex optimization problem:

$$\begin{aligned}&\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \lambda\|P_{\Omega}(\mathbf{S})\|_1\\ &\text{s.t. } P_{\Omega}(\mathbf{Q}) = P_{\Omega}(\mathbf{L}) + P_{\Omega}(\mathbf{S}),\ \mathbf{W} = \tfrac{1}{3}(\mathbf{R}_{\mathrm{c}} + \mathbf{G}_{\mathrm{c}} + \mathbf{B}_{\mathrm{c}}).\end{aligned} \tag{24}$$

In what follows, (24) is converted to the ADMM-applicable form in (3). Let $\mathrm{vec}(\cdot)$ be the operator vectorizing a matrix, and then, the vector forms of $\mathbf{L}$, $\mathbf{S}$, and $\mathbf{W}$ are defined as $\mathbf{l} := \mathrm{vec}(\mathbf{L})$, $\mathbf{s} := \mathrm{vec}(\mathbf{S}) \in \mathbb{R}^{3mn}$, and $\mathbf{w} := \mathrm{vec}(\mathbf{W}) \in \mathbb{R}^{mn}$, respectively. The matrix form of $P_{\Omega}(\cdot)$ is represented as $\mathbf{\Omega} \in \mathbb{R}^{3mn\times 3mn}$, which is a diagonal matrix whose diagonal entries are 1 w.r.t. assigned pixels and 0 otherwise. Additionally, indicator functions of the set Q and W are defined as ι_Q and ι_W, where $Q := \{\mathbf{x} \in \mathbb{R}^{3mn} |\ \mathbf{\Omega}\mathbf{x} = \mathbf{\Omega}\cdot\mathrm{vec}(\mathbf{Q})\}$ and $W := \{\mathbf{x} \in \mathbb{R}^{3mn} |\ \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)} \in \mathbb{R}^{mn}, \mathbf{x} := [(\mathbf{x}^{(1)})^{\top}, (\mathbf{x}^{(2)})^{\top}, (\mathbf{x}^{(3)})^{\top}]^{\top}, \frac{1}{3}(\mathbf{x}^{(1)} + \mathbf{x}^{(2)} + \mathbf{x}^{(3)}) = \mathbf{w}\}$. Then, (24) is rewritten as

$$\min_{\mathbf{l},\mathbf{s}} \|\mathbf{l}\|_* + \lambda\|\mathbf{\Omega}\mathbf{s}\|_1 + \iota_Q(\mathbf{l}+\mathbf{s}) + \iota_W(\mathbf{l}). \tag{25}$$

Next, by setting $\mathbf{u} := [\mathbf{l}^{\top}, \mathbf{s}^{\top}]^{\top}$, we define an auxiliary variable vector $\mathbf{z}$ as follows:

$$\mathbf{z} := \begin{bmatrix}\mathbf{z}^{(1)}\\ \mathbf{z}^{(2)}\\ \mathbf{z}^{(3)}\\ \mathbf{z}^{(4)}\end{bmatrix} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{\Omega}\\ \mathbf{I} & \mathbf{I}\\ \mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{u} = \mathbf{K}\mathbf{u}, \tag{26}$$

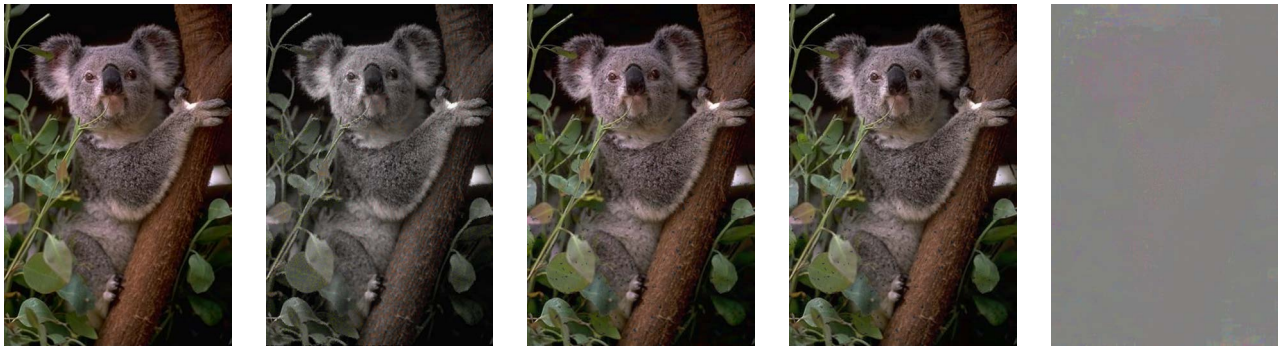
where $\mathbf{I} \in \mathbb{R}^{3mn\times 3mn}$. Finally, the colorization problem in (24) is reformulated into

$$\min_{\mathbf{z},\mathbf{u}} \|\mathbf{z}^{(1)}\|_* + \lambda\|\mathbf{z}^{(2)}\|_1 + \iota_Q(\mathbf{z}^{(3)}) + \iota_W(\mathbf{z}^{(4)}) \text{ s.t. } \mathbf{z} = \mathbf{K}\mathbf{u}. \tag{27}$$

Let $\mathbf{p}_0 := [(\mathbf{p}_0^{(1)})^{\top}, (\mathbf{p}_0^{(2)})^{\top}, (\mathbf{p}_0^{(3)})^{\top}, (\mathbf{p}_0^{(4)})^{\top}]^{\top}$ be an arbitrary vector, where $\mathbf{p}_0^{(i)} \in \mathbb{R}^{3mn}$ in $i = 1, \ldots, 4$. Applying ADMM to (27) yields the following algorithm:

$$\left|\begin{aligned}\mathbf{u}_{n+1} &:= (\mathbf{K}^{\top}\mathbf{K})^{-1}\mathbf{K}^{\top}(\mathbf{z}_n - \mathbf{p}_n)\\ \mathbf{z}_{n+1}^{(1)} &:= \mathrm{prox}_{\frac{1}{\rho}\|\cdot\|_*}(\mathbf{l}_{n+1} + \mathbf{p}_n^{(1)})\\ \mathbf{z}_{n+1}^{(2)} &:= \mathrm{prox}_{\frac{1}{\rho}\|\cdot\|_1}(\mathbf{\Omega}\mathbf{s}_{n+1} + \mathbf{p}_n^{(2)})\\ \mathbf{z}_{n+1}^{(3)} &:= P_Q(\mathbf{l}_{n+1} + \mathbf{s}_{n+1} + \mathbf{p}_n^{(3)})\\ \mathbf{z}_{n+1}^{(4)} &:= P_W(\mathbf{l}_{n+1} + \mathbf{p}_n^{(4)})\\ \mathbf{p}_{n+1} &:= \mathbf{p}_n + \mathbf{K}\mathbf{u}_{n+1} - \mathbf{z}_{n+1}.\end{aligned}\right. \tag{28}$$

The update of $\mathbf{u}_n$ is very easy since $\mathbf{K}$ is composed of simple diagonal matrices. The update of $\mathbf{z}_n^{(1)}$ can be computed by singular



(a) (b) (c) (d) (e)

Fig. 1. Image colorization results with the 10th-order approximation. (a) Original image *Koala*. (b) Gray scaled image + color labels. (c) Exact method [9] (ADMM ver.). (d) Proposed method. (e) The $2\times$ amplified difference between (c) and (d), where 0.5 is added to all the pixel values.

Table 1. The computation time of algorithms and PSNR. The 10th order approximation is good for our method.

Approximation order of $\mathcal{H}_{\mathrm{S}}(\mathbf{C}^\top\mathbf{C})$	5	**10**	15	20	Exact method
Total computation time (s)	21.99	**29.96**	36.15	44.67	41.57(s)
Computation time of singular value thresholding (s)	1.13	**1.72**	2.3	2.95	3.12(s)
PSNR (dB) compared with the exact method	40.08	**41.19**	41.73	42.15	-

value thresholding, which is performed by our CPA-based technique. The update of $\mathbf{z}_n^{(2)}$ is reduced to a well-known soft-thresholding, i.e., shrinking the absolute value of each entry. In the update of $\mathbf{z}_n^{(3)}$ and $\mathbf{z}_n^{(4)}$, $P_Q(\cdot)$ and $P_W(\cdot)$ are the metric projections onto Q and W, respectively. Practically, $P_Q(\cdot)$ is given by maintaining the assigned pixel values. Additionally, the auxiliary vector is calculated as a difference between $\mathbf{w}$ and the vector averaged with three color components of $\mathbf{l}_{n+1} + \mathbf{p}_n^{(4)}$. $P_W(\cdot)$ is derived by adding the auxiliary vector to each color components of $\mathbf{l}_{n+1} + \mathbf{p}_n^{(4)}$. As said before, the proximity operator $\mathrm{prox}_{\frac{1}{\rho}\|\cdot\|_*}$ is approximately calculated by using our technique. To filter the singular values, the filter function in (22) is emprically defined as

$$h(x) := \frac{\max(\sqrt{x}-200,0)}{\sqrt{x}}, \tag{29}$$

where $\max(x_1, x_2)$ is an operator choosing the greater one out of x_1 and x_2.

5. EXPERIMENTS

5.1. Experimental Condition

We examined the effectiveness of our method w.r.t the computation speed and the approximate precision. An eight-bit grayscale image *Koala*[2] was used as $\mathbf{W}$ in (24) whose size is 2400×1600. Additionally, 1% pixels in Fig. 1(a) were automatically chosen color labels, and then, the local-color consistency method was applied to the labeled image for pre-colorization which was used as $\mathbf{Q}$ in (24). The column vectors $\mathbf{l}_0$, $\mathbf{s}_0$, $\mathbf{p}_0$, and $\mathbf{z}_0$ were initialized by all one vectors. We used 5th, 10th, 15th, and 20th-order polynomial approximations to calculate $\mathcal{H}_{\mathrm{S}}(\mathbf{C}^\top\mathbf{C})$. Our method was compared with the exact method, i.e., SVD is performed for the singular value thresholding. In the results, PSNR was calculated by using the results of our method and the exact method to verify the approximate precision. All algorithms were implemented in MATLAB and run on a 2.9 GHz Intel Xeon E5-2690 processor.

[2]The image *Koala* was used in [9], and the image was provided in the author's website.

5.2. Experimental Results

Fig. 1 shows the results of image colorizations using the 10th-order approximation. Although our method approximates the exact method, the result of ours is perceptually equivalent to that of the exact method as shown in Fig. 1(c) and (d). Moreover, the difference between the exact method and ours is quite small even though the pixel values of Fig. 1(e) are amplified for visualization.

Table 1 indicates the computation time of the entire algorithm and that for singular value thresholding. PSNR between our method and the exact method is also shown to verify the approximation precision. Even when the singular value thresholding is approximated with low order, the proposed method indicates high PSNR. As can be seen, the singular value thresholding is computed quickly by using CPA. Moreover, our method reduces the total computation time up to 30% of that of the exact method.

6. CONCLUSION

In this paper, we proposed the fast image colorization method using Chebyshev polynomial approximation. The singular value thresholding can be represented as a multiplication of matrices derived from CPA. The approximation method was applied to the image colorization method with nuclear norm minimization. The experimental results showed that our colorization method is much faster than the conventional method and has the high approximation precision. As a future work, we will investigate to further reduce computation time.

7. REFERENCES

[1] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34-41, Sep, 2001.

[2] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 689-694, 2004.

[3] G. Qiu and J. Guan, "Color by linear neighborhood embedding," in Proc. *IEEE Int. Conf. Image Processing*, vol. 3, pp. 988-991, Sept., 2005.

[4] G. Sapiro, "Inpainting the colors," in Proc. *IEEE Int. Conf. Image Processing*, vol. 2, pp. 698-701, Sept., 2005.

[5] Y. Qu, T.-T. Wong, and P.-A. Heng, "Manga colorization," *ACM Transactions on Graphics (TOG)*, vol. 25, issue. 3, pp. 1214-1220, July, 2006.

[6] X. Liu, L. Wan, Y. Qu, T.-T. Wong, S. Lin, C.-S Leung, and P.-A. Heng, "Intrinsic colorization," *ACM Transactions on Graphics (TOG)*, vol. 27, issue. 5, pp. 1214-1220, July, 2006.

[7] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Trans. Image Processing*, vol. 15, no. 5, pp. 1120-1129, May, 2006.

[8] R. Szeliski *Computer Vision Algorithms and Applications*, USA: Springer, 2010.

[9] S. Wang and Z. Zhang, "Colorization by matrix completion," in Proc. *Twenty-Sixth AAAI Conference on Artificial Intelligence*, pp. 1169-1175, July, 2012.

[10] M. Fornasier, "Nonlinear projection recovery in digital inpainting for color image restoration," *Journal of Mathematical Imaging and Vision*, vol. 24, no. 3, pp. 359-373, 2006.

[11] W. Baatz, M. Fornasier, P. Markowich, C.-B. Schönlieb, "Inpainting of ancient austrian frescoes," Conference proceedings of Bridges 2008, Leeuwarden 2008, pp. 150-156.

[12] S. Ono, T. Miyata, and Y. Sakai, "Colorization-based coding by focusing on characteristics of colorization bases," in Proc. Picture Coding Symp. pp. 230-233, Dec. 2010.

[13] S. Lee, S. W. Park, P. Oh, and M. G. Kang, "Colorization-based compression using optimization," *IEEE Trans. Image Processing*, vol. 22, no. 7, pp. 2627-2636, 2013.

[14] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *UIUC Technical Report, UILU-ENG-09-2215*, Nov., 2009.

[15] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in Proc. *IEEE Int. Conf. Computer Vision (ICCV)*, pp. 839-846, Bombay, India, Jan., 1998.

[16] S. Paris, P. Jirboribst, J. Tumblin, and F. Durand, "Bilateral filtering: Theory and applications," *Foundations and Trends in Computer Graphics and Vision*, vol. 4, no. 12, pp. 2324-2333, Dec., 2009.

[17] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic range images," *ACM Trans. Graphics*, vol. 21, issue. 3, pp. 257-266, Jul., 2002.

[18] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 106-128, Jan., 2013.

[19] H. Talebi, X. Zhu and P. Milanfar, "How to SAIF-ly boost denoising performance," *IEEE Trans. Image Processing*, vol. 22, no. 4, pp. 1470-1485, Apr., 2013.

[20] H. Talebi and P. Milanfar, "Global image denoising," *IEEE Trans. Image Processing*, vol. 23, no. 2, pp. 755-768, Feb., 2014.

[21] G. M. Phillips, *Interpolation and approximation by polynomials*, CMS Books Math., Springer-Verlag, 2003.

[22] D. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129-150, 2011.

[23] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*, Chapman and Hall/CRC, 2003.

[24] D. I. Shuman, P. Vandergheynst, and P. Frossard, "Chebyshev polynomial approximation for distributed signal processing", *IEEE Int. Conf. Distributed Computing in Sensor Systems and Workshops*, Barcelona, Spain, June, 2011.

[25] M. Onuki, S. Ono, K. Shirai, and Y. Tanaka, "Non-local/local image filters using fast eigenvalue filtering," in Proc. *IEEE International Conference on Image Processing*, pp. 4659-4663, Québec City, Canada, Sep., 2015.

[26] K. Suwabe, M. Onuki, Y. Iizuka, and Y. Tanaka, "Globalized BM3D using fast eigenvalue filtering," in Proc. *IEEE Global Conference on Signal and Information Processing*, Orlando, Florida, USA, Dec. 2015.

[27] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite elements approximations," *Comput. Math. Appl.*, vol. 2, pp. 17-40, 1976.

[28] J. Eckstein and D. Bertsekas, "On the Douglas-Rachford splitting method and proximal point algorithm for maximal monotone operators," Math. Program., vol. 55, pp. 293-318, 1992.

[29] J. J. Moreau, "Fonctions convexes duales er points proximaux dans un espace hilbertien," *C. R. Acad. Sci. Prais Ser. A Math*, vol. 255, pp. 2897-2899, 1962.

[30] E. J. Candès and T. Tao, "The power of convex relaxation: near-optimal matrix completion," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2053-2080, 2010.