Fast Smooth Rank Approximation for Tensor Completion

Mohammed Al-Qizwini
Department of Electrical and Computer Engineering
Michigan State University
East Lansing, Michigan, USA
alqizwin@egr.msu.edu

Hayder Radha, IEEE Fellow
Department of Electrical and Computer Engineering
Michigan State University
East Lansing, Michigan, USA
radha@egr.msu.edu

Abstract— **In this paper we consider the problem of recovering an *N*-dimensional data from a subset of its observed entries. We provide a generalization for the smooth Shcatten-p rank approximation function in [1] to the *N*-dimensional space. In addition, we derive an optimization algorithm using the Augmented Lagrangian Multiplier in the *N*-dimensional space to solve the tensor completion problem. We compare the performance of our algorithm to state-of-the-art tensor completion algorithms using different color images and video sequences. Our experimental results showed that the proposed algorithm converges faster (approximately half the execution time), and at the same time it achieves comparable performance to state-of-the-art tensor completion algorithms.**

Keywords*— *tensor completion; smooth rank function; augmented lagrange multiplier; nuclear norm minimization

I. Introduction and Related Work

Recovering signals from few observed samples is emerging as a powerful tool in a wide range of applications [2]; some examples are: computer vision [3], control [4] and wireless sensor networks [5]. Even though in several applications, such as computer vision, we deal with a higher dimensional space, we find few studies considered the *N*-dimensional tensor completion case because of the difficulties associated with higher dimensional space computations [6], [7]. Some researchers, for example in [8], applied their matrix completion algorithms to color images by considering each color channel as a separate matrix. Such algorithms don't consider the correlation among the different channels. Previous algorithms solved the matrix and tensor completion problems based on minimizing the nuclear norm, which has been shown [9] to be the tightest approximation to the rank function [2], [6], [10]–[13]. Some other algorithms also used a reweighted nuclear norm, for example in [14], to achieve better results than using only the truncated nuclear norm. However, minimizing the nuclear norm requires evaluating the computationally expensive SVD in each iteration. The authors in [15] proposed using reweighted least squares function to recover sparse signals for the compressed sensing problem. In [1] the authors proposed an extension of the work in [15] into the matrix completion problem and proposed computationally efficient algorithms for matrix completion. In this paper, we extend these efficient matrix completion algorithms to the *N*-dimensional tensor completion problem by first providing a general definition for the Schatten-P function into the *N*-dimensional space. We also derive a multi-dimensional Augmented Lagrangian Multiplier (ALM) optimization algorithm to solve it.

The rest of the paper is organized as follows: Section II defines the notations and briefly reviews the matrix and tensor completion problems. Section III provides a detailed description of our contributions and the derivation of the ALM algorithm for the smooth rank approximation tensor completion problem. Section IV presents the experimental results to evaluate the performance of our method compared to the previous well known algorithms. Finally in section V we conclude our work and the experimental results.

II. Notations and Background

Throughout this paper, matrices are denoted using capital letters, e.g. M, and tensors by cursive capital letters, e.g. $\mathcal{M}$. Also, M^T is the transpose of M. $M_{i_1 i_2}$ is an entry of the matrix $M \in \mathbb{R}^{m\times n}$ at row i_1 and column i_2; and $\mathcal{M}_{i_1 \dots i_N}$ is an entry of the tensor $\mathcal{M} \in \mathbb{R}^{m_1\times m_2 \dots m_N}$ at indices i_1 through i_N. For a matrix $M \in \mathbb{R}^{m\times n}$, $P_\Omega(M)$ is the sampling operator which is defined as:

$$P_\Omega(M) = \begin{cases} M_{i_1 i_2} & if\ i_1, i_2 \in \Omega \\ 0 & elsewhere \end{cases} \qquad (1)$$

where Ω represents the set of observed entries.

For $\mathcal{M} \in \mathbb{R}^{m_1\times m_2 \dots m_N}$, the $unfold(.)_{(i)}$ operation over dimension i unfolds the N-dimensional tensor into a matrix, $fold(.)_{(i)}$ is the inverse of $unfold(.)_{(i)}$ and it can be defined as:

$$\mathcal{M}{=}fold(unfold(\mathcal{M})_{(i)})_{(i)} \qquad (2)$$

The inner product between two matrices is defined by:

$$\langle C, E\rangle = Tr(CE) \qquad (3)$$

where $Tr(.)$ is the trace function.

If $M = U\Sigma V^T$ is the SVD decomposition of M, then the soft thresholding operator over M is defined as [6], [10]:

$$D_{Th}(M) = U\Sigma_{Th}V^T \qquad (4)$$

where $\Sigma_{Th} = max(\sigma_i - Th, 0)$ is the diagonal soft thresholded singular values matrix.

The convex approximation to the matrix completion problem can be represented by:

$$\min_{X} \|X\|_* \quad s.t.\ P_\Omega(X) = P_\Omega(M) \tag{5}$$

where X is the low rank matrix to be estimated, M is the original matrix and $P_\Omega(.)$ is defined in (1).

Since the observed samples are usually contaminated with noise, we need to relax the equality constraint into an inequality constraint in order to make (5) more robust to noise [12].

$$\min_{X} \|X\|_* \quad s.t.\ \|P_\Omega(X) - P_\Omega(M)\|_F^2 \le \varepsilon \tag{6}$$

Tensor completion is a generalization to the matrix completion problem into the N-dimensional space, it can be represented by [7], [16]:

$$\min_{\mathcal{X}} \|\mathcal{X}\|_* \quad s.t.\ \|P_\Omega(\mathcal{X}) - P_\Omega(\mathcal{M})\|_F^2 \le \varepsilon \tag{7}$$

where $\|\mathcal{X}\|_*$ is the nuclear norm of the tensor which is defined by [7], [16]:

$$\|\mathcal{X}\|_* = \frac{1}{N}\sum_{i=1}^{N} \left\|unfold(\mathcal{X})_{(i)}\right\|_* \tag{8}$$

Instead of minimizing the convex nuclear norm function in (5), the authors in [1] suggested to use a smooth approximation to the rank function and they showed that it converges faster than minimizing the nuclear norm function; they considered the smooth Schatten-p function:

$$f_p(X) = Tr(X^T X + \gamma I)^{\frac{p}{2}} \tag{9}$$

Here, I is the identity matrix and $\gamma > 0$. The function (9) is convex for $p \ge 1$ [1].

So the matrix rank minimization approximation problem can be written as [1]:

$$min_X\, Tr(X^T X + \gamma I)^{\frac{p}{2}} \quad s.t.\ P_\Omega(X) = P_\Omega(M) \tag{10}$$

III. Smooth Rank Approximation Tensor Completion

In this paper we are interested in using a generalized version of the smooth Schatten-p function to solve the tensor completion problem, and hence we need to redefine (9) to the higher dimensional space.

Definition 3.1: The smooth Schatten-p function for an N-dimensional signal is given by:

$$f_p(\mathcal{X}) = \frac{1}{N}\sum_{i=1}^{N} Tr(unfold(\mathcal{X})_{(i)}^T unfold(\mathcal{X})_{(i)} + \gamma I)^{\frac{p}{2}} \tag{11}$$

One can verify the validity of Defintion 3.1 for the 2-dimensional Schatten-p case ($N = 2$) by observing that the trace of the matrix and its transpose are the same; and hence, (11) reduces to (9).

Now the tensor completion problem based on a smooth rank approximation function can be casted as:

$$\min_{\mathcal{X}} \frac{1}{N}\sum_{i=1}^{N} Tr(unfold(\mathcal{X})_{(i)}^T unfold(\mathcal{X})_{(i)} + \gamma I)^{\frac{p}{2}} \quad s.t.\ P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{M}) \tag{12}$$

Next we derive the ALM optimization method to solve the tensor completion problem in (12). Using similar idea for the matrix case in [13], we can rewrite (12) in the tensor domain as:

$$\min_{\mathcal{X}} \frac{1}{N}\sum_{i=1}^{N} Tr(unfold(\mathcal{X})_{(i)}^T unfold(\mathcal{X})_{(i)} + \gamma I)^{\frac{p}{2}} \quad s.t.\ \mathcal{X} + \mathcal{E} = P_\Omega(\mathcal{M}), P_\Omega(\mathcal{E}) = 0 \tag{13}$$

Here, $\mathcal{X}$ represents the non-observed entries of the original tensor $\mathcal{M}$. The partial augmented Lagrange function for (13) is given by:

$$\begin{aligned} L(\mathcal{X}, \mathcal{E}, \mathcal{Y}, \mu) = \sum_{i=1}^{N} & \alpha_i Tr(unfold(\mathcal{X})_{(i)}^T unfold(\mathcal{X})_{(i)} \\ & + \gamma I)^{\frac{p}{2}} + \langle \mathcal{Y}, P_\Omega(\mathcal{M}) - \mathcal{X} - \mathcal{E} \rangle \\ & + \frac{\mu}{2}\|P_\Omega(\mathcal{M}) - \mathcal{X} - \mathcal{E}\|_F^2 \end{aligned} \tag{14}$$

where $\mathcal{Y}$ is the Lagrange multiplier tensor.

In this section we state the solution for (14) with respect to each variable ($\mathcal{X}, \mathcal{E}$ and $\mathcal{Y}$) as three lemmas; the proofs are presented separately in appendix A.

Lemma 3.1: The optimal solution for (14) with respect to $\mathcal{X}$ is given by:

$$\mathcal{X}^{k+1} = P_\Omega(\mathcal{M}) + \frac{1}{\mu}\mathcal{Y}^k - \left[\mathcal{E}^k + \frac{\frac{1}{\mu}\sum_{i=1}^{N} \beta_i \mathcal{Z}_i^{k+1}}{\sum_{i=1}^{N} \beta_i}\right] \tag{15}$$

where β is a control parameter and $\mathcal{Z}_i^{k+1}$ is given by:

$$\mathcal{Z}_i^{k+1} = fold\left(\alpha_i unfold(\mathcal{X}^k)_{(i)} W_i\right)_{(i)} \tag{16}$$

W_i is defined as:

$$W_i = (unfold(\mathcal{X}^k)_{(i)}^T unfold(\mathcal{X}^k)_{(i)} + \gamma I)^{1-\frac{p}{2}} \tag{17}$$

Lemma 3.2: The optimal solution for (14) with respect to $\mathcal{E}$ is given by:

$$P_{\bar{\Omega}}(\mathcal{E}^{k+1}) = P_{\bar{\Omega}}(\mathcal{M}+\frac{1}{\mu}\mathcal{Y}^{k} - \mathcal{X}^{k+1}) \tag{18}$$
$$P_{\Omega}(\mathcal{E}^{k+1})=0$$

Lemma 3.3: The optimal solution for (14) with respect to $\mathcal{Y}$ is given by:

$$\mathcal{Y}^{k+1} = \mathcal{Y}^{k} + \mu(P_{\Omega}(\mathcal{M}) - \mathcal{X}^{k+1} - \mathcal{E}^{k+1}) \tag{19}$$

Benefitting from the smooth Schatten-P function formula we used a code optimization idea in order to reduce the computation complexity. Since the trace of the matrix $unfold(\mathcal{X}^{k})_{(i)} \in \mathbb{R}^{p\times q}$ and its transpose are the same; then while computing W_i we used the transpose of $unfold(\mathcal{X}^{k})_{(i)}$ if $q > p$ and we also used the term $(unfold(\mathcal{X}^{k})_{(i)}W_i)^T$ in order to maintain the original dimensions of the unfolded tensor. The proposed Smooth Rank Approximation Tensor Completion (SRATC) algorithm is presented in algorithm 1.

```
Algorithm 1:        SRATC
Input:      $\mu > 0, \beta \in \mathbb{R}^N, P_{\Omega}(\mathcal{M})$
Steps:   1: initialize $\mathcal{X}^1 = P_{\Omega}(\mathcal{M})$, $\mathcal{Y}^1 = \mathcal{E}^1 = 0$
         2: while  Not converged
             Do:  For each dimension $i$
                      Evaluate $W_i$ using (17)
                      Evaluate $\mathcal{Z}_i^{k+1}$ using (16)
                  End For
                  Evaluate $\mathcal{X}^{k+1}$ using (15)
                  Evaluate $\mathcal{E}^{k+1}$ using (18)
                  Evaluate $\mathcal{Y}^{k+1}$ using (19)

             End While
Output:     $\mathcal{X}$, $\mathcal{E}$
```

IV. Experimental Results

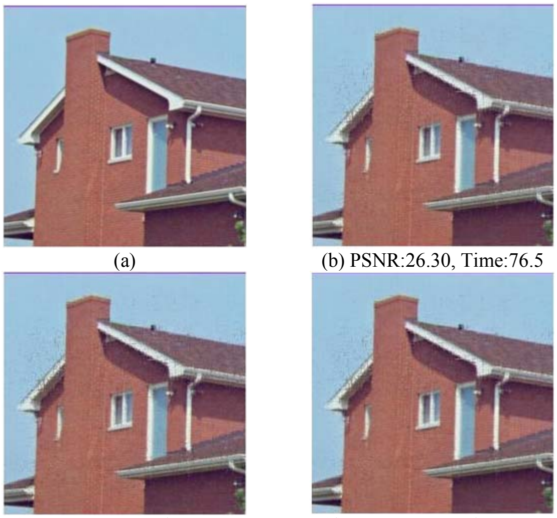
In this section we present the results of the experiments that we performed to compare the performance of our SRATC algorithm to the Low Rank Tensor Completion (LRTC) algorithm in [6] and the Accelerated Proximal Gradient (APG) tensor completion in [16].

The quality of the recovered data is evaluated using the Peak Signal to Noise Ratio (PSNR) and the convergence rate is evaluated based on the time needed by the algorithm to converge to the final results. We ran the algorithms with MATLAB 2012b on the same desktop computer with a 3.3 GHz i-5 CPU and a 4GB Memory.

In the first application we applied the tensor completion algorithms to recover the house color image (size 256×256) [6] with rank 80 in each color channel from random 50% observed samples; the results are shown in Fig.1.

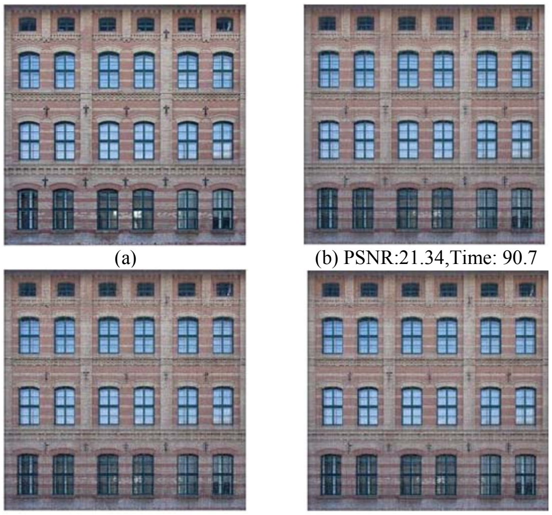
From Fig.1 we see that the APG takes less execution time than LRTC but it converges to the lowest PSNR, while the LRTC takes more time to execute than APG but it achieves higher PSNR. Our algorithm achieves comparative PSNR to LRTC, but it needs the lowest execution time.

To test the performance of the tensor completion algorithms under high number of missing entries, we used the algorithms to recover the façade image (size 256×256 and rank 80 for each color channel) in [6] and using only 30% observed samples; the results are presented in Fig.2.


(a) (b) PSNR:26.30, Time:76.5

(c) PSNR:25.79, Time:64.1 (d) PSNR:26.34, Time:37.8

Fig.1. (a) The original house image, (b-d) Recovered images with the corresponding PSNR (dB) and execution time (Sec.) using (b) LRTC, (c) APG, (d) SRATC


(a) (b) PSNR:21.34,Time: 90.7

(c) PSNR:20.41,Time:75.5 (d) PSNR: 21.27,Time:49.5

Fig.2. (a) The original façade image, (b-d) Recovered images with the corresponding PSNR (dB) and execution time (Sec.) using (b) LRTC, (c) APG, (d) SRATC

Again, Fig.2 shows that APG achieves the lowest PSNR measure but converges faster than LRTC, while the proposed SRATC algorithm converges the fastest with almost the same performance as LRTC. LRTC converges the slowest but it achieves a good PSNR value.

Fig.3 shows the effect of changing the rank on the performance of the tensor completion algorithms. In this experiment we used the color lena image (size 256×256) with 50% observed samples.

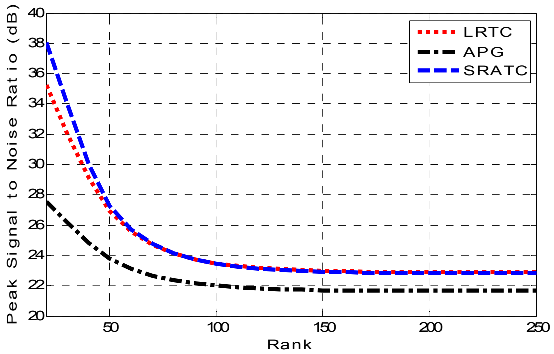


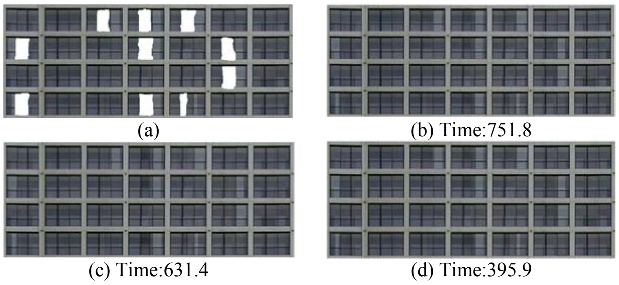
Fig.3. Peak Signal to Noise Ratio versus the rank of the matrix using lena colored image with 50% observed samples.

From Fig.3 we see that when the rank of the data is very low, SRATC achieves the highest PSNR. While when the rank starts to increase, the SRATC and LRTC algorithms both have approximately the same PSNR measure. From this experiment, we also noticed that the computation time remains approximately constant as we change the rank.

To give an accurate performance evaluation for the tensor completion algorithms, we apply the algorithms to recover a set of 10 different color images[1] (size 256×256 and rank 60 for each color channel) from (70, 50 and 30)% observed entries; then we average the PSNR and the execution time; the results are shown in Table 1. We notice that regardless of the number of observed entries, the SRATC algorithm achieves the lowest execution time while LRTC and APG require significantly longer time to converge because of using the SVD decomposition in each iteration.

Missing (%)	**Method**	**PSNR (dB)**	**Time (Second)**
30	LRTC	32.84	63.5
	APG	30.51	57.6
	SRATC	**33.85**	**28.7**
50	LRTC	26.59	77.3
	APG	24.93	57.1
	SRATC	**26.74**	**31.5**
70	LRTC	**21.54**	77.3
	APG	19.39	70.0
	SRATC	21.46	**40.3**

Table 1: Average execution time and PSNR comparison for tensor completion algorithms using a set of 10 colored images.



(a)

(b) Time:751.8

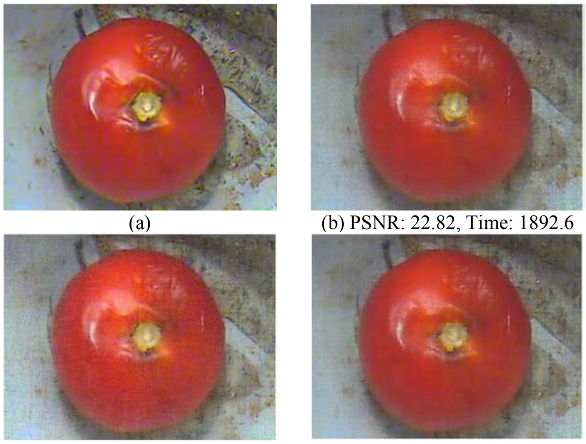
(c) Time:631.4

(d) Time:395.9

Fig.4. (a) The masked façade image, (b-d) Recovered images with the corresponding PSNR (dB) and execution time (Sec.) using (b) LRTC, (c) APG, (d) SRATC

Another application for testing the tensor completion algorithms is blocks image inpainting. We used the façade image of size 318×861 [6]. In this application we are trying to recover the missing parts of the façade image texture by inpainting missing blocks of the image. The results are shown in Fig. 4. The results presented in Fig. 4 show that our algorithm also converges faster than state-of-the-art tensor completion algorithms. For this particular simulation, we don't have the original image for evaluating the PSNR values; however, by observing the visual quality of the different results, one can observe that the proposed SRATC algorithm provides a visual quality similar to the LRTC algorithm while converging with about one-half of the time required by LRTC.

To show the applicability of our SRATC algorithm for higher dimensional data and non-square tensors, we show in Fig. 5 the result of recovering a video sequence of the tomato video [6]; we used 20 frames with each frame of size 242×320 pixels. We also compared the SRATC result to LRTC and APG. As shown in Fig. 5, our SRATC algorithm saves about 4 minutes when compared to the APG algorithm in addition to the higher PSNR and the visually better recovered frames. Compared to LRTC, our algorithm saves approximately 10 minutes execution time and achieves the same PSNR value.



(a)

(b) PSNR: 22.82, Time: 1892.6

(c) PSNR:21.01, Time:1530.8

(d) PSNR: 22.86, Time: 1311.4

Fig.5. (a) The original tomato frame (number 6), (b-d) Recovered frames with the corresponding PSNR (dB) and execution time (Sec.) using (b) LRTC, (c) APG, (d) SRATC

V. Conclusions

In this paper, we developed a generalization to the Schatten-P function into the *N*-dimensional space to provide a less computational complexity algorithm for tensor completion. The experimental results that we presented showed that the proposed SRATC algorithm achieves similar or slightly higher PSNR values than what can be achieved using the powerful tensor completion algorithms that are based on the computational complex SVD decomposition; meanwhile SRATC requires significantly less computation time to converge to the same PSNR value.

[1] The images set is downloaded from http://web.eecs.utk.edu/~gonzalez/

APPINDIX A

Proof of Lemma 3.1:

From (14) we complete the sum of squares:

$$L(\mathcal{X},\mathcal{E},\mathcal{Y},\mu) = \sum_{i=1}^{N} \alpha_i Tr(unfold(\mathcal{X})_{(i)}^{T} unfold(\mathcal{X})_{(i)} + \gamma I)^{\frac{p}{2}} + \frac{\mu}{2}\left\| P_{\Omega}(\mathcal{M}) - \mathcal{X} - \mathcal{E} + \frac{1}{\mu}\mathcal{Y}\right\|_F^2 - \frac{1}{\mu}\|\mathcal{Y}\|_F^2 \tag{A1}$$

Deriving and solving (A1) for $\mathcal{X}$:

$$\sum_{i=1}^{N} \frac{\alpha_i (unfold(\mathcal{X})_{(i)}^{T})}{(unfold(\mathcal{X})_{(i)}^{T} unfold(\mathcal{X})_{(i)} + \gamma I)^{1-\frac{p}{2}}} - \mu(P_{\Omega}(\mathcal{M}) - \mathcal{X} - \mathcal{E} + \frac{1}{\mu}\mathcal{Y}) = 0 \tag{A2}$$

For simplicity, we assume W_i is given by (17). Then the solution for $\mathcal{X}$ is:

$$\mathcal{X}^{k+1} = P_{\Omega}(\mathcal{M}) - \mathcal{E} + \frac{1}{\mu}\mathcal{Y} - \frac{\frac{1}{\mu}\sum_{i=1}^{N} \beta_i \, fold\,(\alpha_i \, unfold(\mathcal{X})_{(i)}^{T} W_i)_{(i)}}{\sum_{i=1}^{N} \beta_i} \tag{A3}$$

For further simplification, we assume $\mathcal{Z}_i{}^{k+1}$ as presented in (16) and hence $\mathcal{X}^{k+1}$ is reduced to (15).

Proof of Lemma 3.2:

We start from (A1) and derive for $\mathcal{E}$. But here we need to include another constraint to force the entries of $\mathcal{E}$ at the observed samples entries to be zero.

$$\mu(P_{\Omega}(\mathcal{M}) - \mathcal{X} - \mathcal{E} + \frac{1}{\mu}\mathcal{Y}) = 0 \quad s.t.\, P_{\Omega}(\mathcal{E}){=}0 \tag{A4}$$

Now solving for $\mathcal{E}$:

$$\mathcal{E}^{k+1} = P_{\Omega}(\mathcal{M}) - \mathcal{X} + \frac{1}{\mu}\mathcal{Y} \quad P_{\Omega}(\mathcal{E}^{k+1}){=}0 \tag{A5}$$

Or we can rewrite (A4) as in (18).

Proof of Lemma 3.3:

Also starting from (A1), we derive for $\mathcal{Y}$:

$$P_{\Omega}(\mathcal{M}) - \mathcal{X} - \mathcal{E} + \frac{1}{\mu}\mathcal{Y} - \frac{1}{\mu}\mathcal{Y} = 0 \tag{A6}$$

Solving for $\mathcal{Y}$:

$$\mathcal{Y}^{k+1} = \mathcal{Y}^{k} + \mu(P_{\Omega}(\mathcal{M}) - \mathcal{X} - \mathcal{E}) \tag{A7}$$

Which is the same result shown in (19).

REFERENCES

[1] K. Mohan and M. Fazel, “Iterative reweighted algorithms for matrix rank minimization,” J. Mach. Learn. Res., vol. 13, pp. 3441–3473, 2012.

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

[3] H. Ji, C. Liu, Z. Shen, and Y. Xu, “Robust video denoising using low rank matrix completion,” Comput. Vis. Pattern …, pp. 1791–1798, Jun. 2010.

[4] Z. Liu and L. Vandenberghe, “Interior-point method for nuclear norm approximation with application to system identification,” SIAM J. Matrix Anal. Appl., vol. 31, no. 3, pp. 1235–1256, Jan. 2009.

[5] F. Fazel, M. Fazel, and M. Stojanovic, “Random access sensor networks: Field reconstruction from incomplete data,” 2012 Inf. Theory Appl. Work., pp. 300–305, Feb. 2012.

[6] J. Liu, P. Musialski, P. Wonka, and J. Ye, “Tensor completion for estimating missing values in visual data,” Comput. Vision, 2009 IEEE 12th Int. Conf., pp. 2114–2121, Jan. 2009.

[7] M. Signoretto, V. de P. Raf, D. M. Bart, and J. A. Suykens, “Tensor versus matrix completion: a comparison with application to spectral data,” Signal Process. Lett., vol. 18, no. 7, pp. 403–406, Jul. 2011.

[8] D. Zhang, Y. Hu, Jj. Ye, X. Li, and X. He, “Matrix completion by truncated nuclear norm regularization,” Comput. Vis. Pattern Recognit., pp. 2192–2199, Jun. 2012.

[9] M. Fazel, “Matrix rank minimization with applications,” Stanford University, 2002.

[10] S. Ma, D. Goldfarb, and L. Chen, “Fixed point and Bregman iterative methods for matrix rank minimization,” Math. Program., vol. 128, no. 1–2, pp. 321–353, 2011.

[11] R. S. Cabral, F. Torre, J. P. Costeira, and A. Bernardino, “Matrix completion for multi-label image classification,” Adv. Neural Inf. Process. Syst., pp. 1–9, 2011.

[12] R. H. Keshavan, A. Montanari, and S. Oh, “Matrix completion from noisy entries,” J. Mach. Learn. Res., vol. 99, pp. 2057–2078, 2010.

[13] Z. Lin, M. Chen, and Y. Ma, “The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices,” arXiv Prepr. arXiv1009.5055, 2010.

[14] K. Mohan and M. Fazel, “Reweighted nuclear norm minimization with application to system identification,” Am. Control Conf. (ACC), 2010, no. 5, 2010.

[15] I. Daubechies and R. DeVore, “Iteratively reweighted least squares minimization for sparse recovery,” Commun. Pure Appl. Math., vol. 63, no. 1, pp. 1–38, 2010.

[16] Z. Shi, T. Zheng, and J. Han, “Trace norm regularized tensor classification and its online learning approaches,” arXiv Prepr. arXiv1109.1342, pp. 1–11, 2011.