

Reproduction of “Fast and Accurate Matrix Completion via Truncated Nuclear Norm Regularization”

Zhao-Yang Liu & Xin-Hua Zheng

Dec. 09, 2022

Contests

- 1 || Background
- 2 || Related Work (Baseline)
- 3 || Truncated Nuclear Norm Regularization
- 4 || Optimization ways -- ADMM & APGL & ADMMAP
- 5 || Reproductions' Results

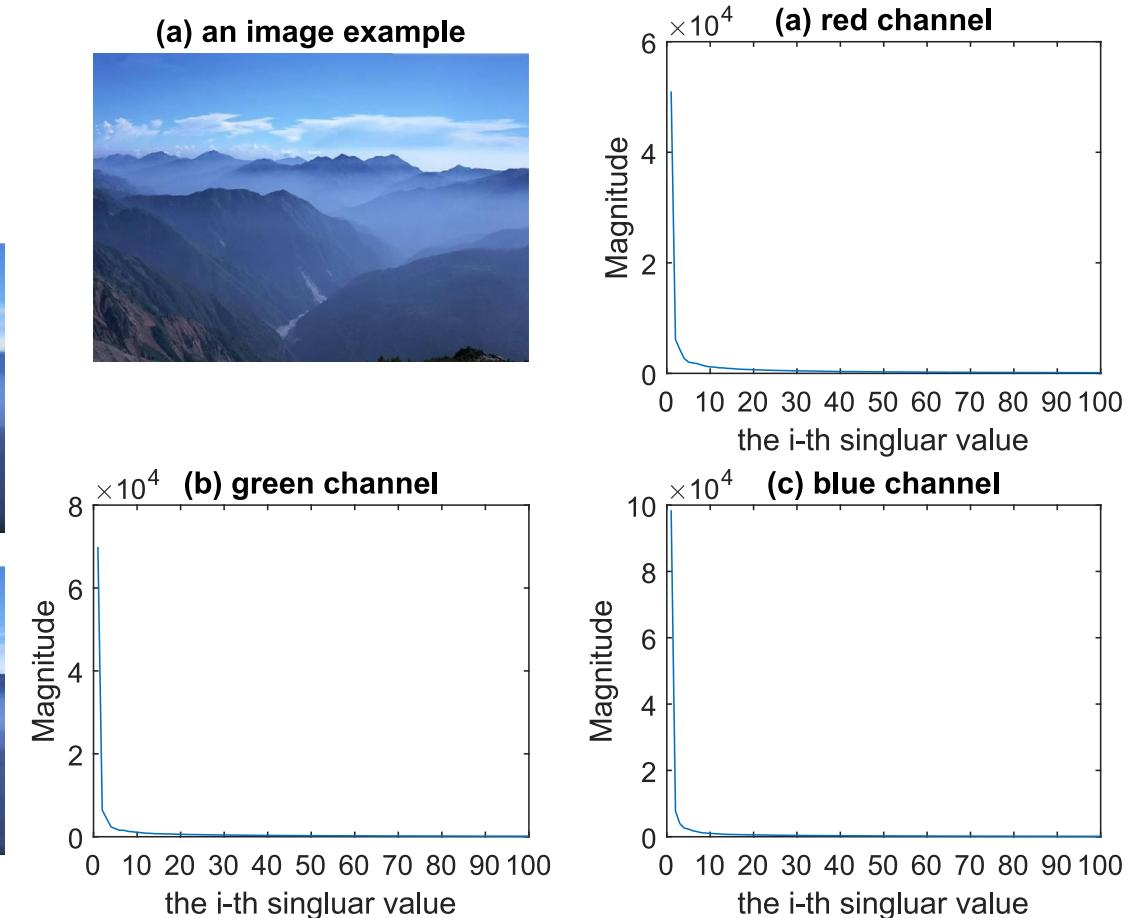
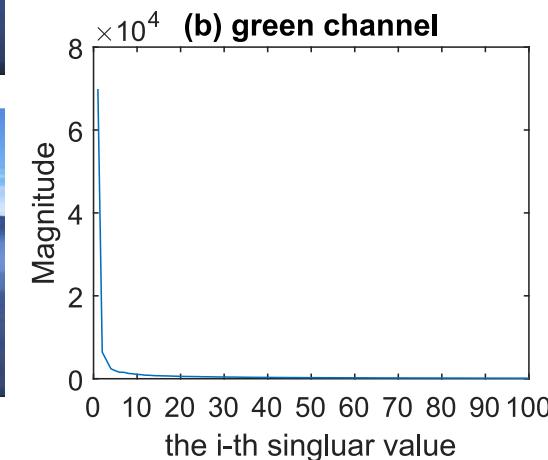
• 1. Background – rank

Using `svd` function decompose the image (Channels separately)

- The first few singular values are much larger than others
- For $r > 20$, r -th singular value σ_r close to 0



(a) an image example

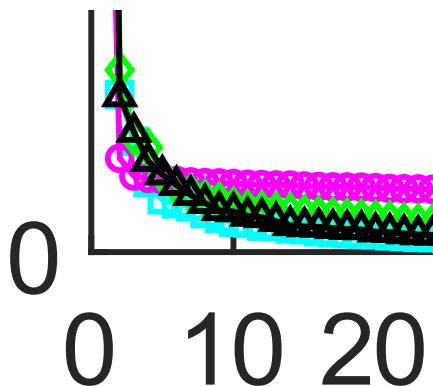


First Few values contain main information → Low rank approximate is reasonable → Truncated svd (svds)

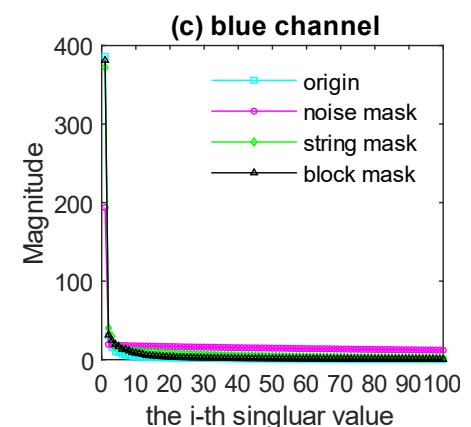
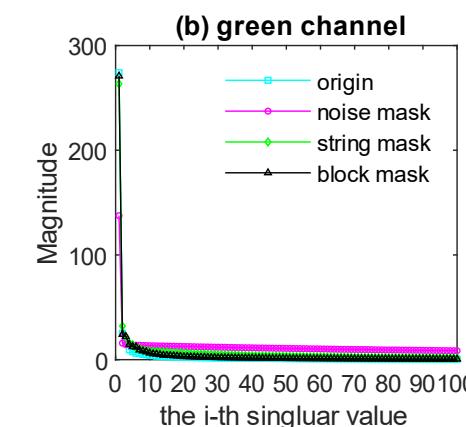
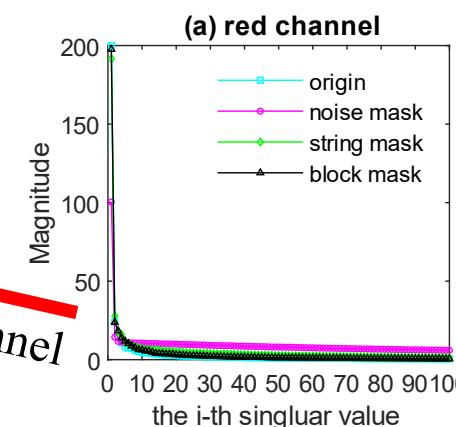
• 1. Background – with mask

We mask the figure with three ways:

- Random mask (50% missing)
- String mask (add string)
- Block mask (add block)



Red Channel



$$\min_{\mathbf{X}} \text{rank}(\mathbf{X})$$

$$s.t. \quad \mathcal{P}_{\Omega}(\mathbf{X}) = \mathcal{P}_{\Omega}(\mathbf{M})$$

$$\Rightarrow \min_{\mathbf{X}} \|\mathbf{X}\|_* \quad [1]$$

$$s.t. \quad \mathcal{P}_{\Omega}(\mathbf{X}) = \mathcal{P}_{\Omega}(\mathbf{M})$$

$\mathbf{M} \in \mathbb{R}^{m \times n}$, $\mathbf{X} \in \mathbb{R}^{m \times n}$, Ω observed entries, Orthogonal projection operator \mathcal{P}_{Ω}

[1] M. Fazel, "Matrix Rank Minimization with Applications," PhD thesis, Stanford Univ., 2002.

• 2. Related Work

- Singular Value Thresholding (SVT)^[1]: Time complexity $\mathcal{O}(\frac{1}{N})$

$$\begin{aligned} \min_{\mathbf{X}} \quad & \|\mathbf{X}\|_* + \alpha \|\mathbf{X}\|_F^2 \\ \text{s.t.} \quad & \mathcal{P}_{\Omega}(\mathbf{X}) = \mathcal{P}_{\Omega}(\mathbf{M}) \end{aligned} \Rightarrow L(\mathbf{X}, \mathbf{Y}) = \|\mathbf{X}\|_* + \alpha \|\mathbf{X}\|_F^2 + \langle \mathbf{Y}, \mathcal{P}_{\Omega}(\mathbf{M} - \mathbf{X}) \rangle$$

- Singular Value Projection (SVP)^[2]: Solve the rank minimization problem

$$\begin{aligned} \min_{\mathbf{X}} \quad & \|\mathbf{P}_{\Omega}(\mathbf{X}) - P_{\Omega}(\mathbf{M})\|_F^2 \\ \text{s.t.} \quad & \text{rank}(\mathbf{X}) \leq r \end{aligned} \Rightarrow L(\mathbf{X}, \mathbf{Y}) = \min_{\mathbf{S} \in \mathbb{R}^{r \times r}} \|\mathcal{A}(\mathbf{X}) - \mathbf{y}\|_F^2 \Rightarrow \begin{aligned} \mathbf{Y}_{k+1} &= \mathbf{X}_k - \gamma_k \mathcal{A}^*(\mathcal{A}(\mathbf{X}_k) - \mathbf{y}) \\ \mathbf{X}_{k+1} &= \text{Trancated SVD}_r(\mathbf{Y}_{k+1}) \end{aligned}$$

- OptSpace^[3]:

$$\begin{aligned} \min_{\mathbf{X}} \quad & \|\mathbf{P}_{\Omega}(\mathbf{X}) - P_{\Omega}(\mathbf{M})\|_F \\ \text{s.t.} \quad & \text{rank}(\mathbf{X}) \leq r \end{aligned} \Rightarrow L(\mathbf{X}, \mathbf{Y}) = \min_{\mathbf{S} \in \mathbb{R}^{r \times r}} L(\mathbf{X}, \mathbf{Y}, \mathbf{S}) = \frac{1}{2} \|\mathcal{P}_{\Omega}(\mathbf{M} - \mathbf{X} \mathbf{S} \mathbf{Y}^T)\|_F^2 + \frac{\lambda}{2} \|\mathcal{P}_{\Omega^c}(\mathbf{M} - \mathbf{X} \mathbf{S} \mathbf{Y}^T)\|_F^2$$

$\mathbf{Y} \in \mathbb{R}^{m \times n}$ is the Lagrange multiplier matrix, inner produce $\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i,j} X_{ij} Y_{ij}$

[1] J.F. Cai, E.J. Candès, and Z. Shen, “A Singular Value Thresholding Algorithm for Matrix Completion,” SIAM J. Optimization, vol. 20, pp. 1956–1982, 2010.

[2] P. Jain, R. Meka, and I. Dhillon, “Guaranteed Rank Minimization via Singular Value Projection,” in Advances in Neural Information Processing Systems, 2010, vol. 23

[3] R. H. Keshavan and S. Oh, “A Gradient Descent Algorithm on the Grassmann Manifold for Matrix Completion,” Transportation Research Part C: Emerging Technologies, vol. 28, pp. 15–27, Mar. 2013.

• 3. Truncated Nuclear Norm Regularization (TNNR)

- Truncated nuclear norm: $\|\mathbf{X}\|_r = \sum_{i=r+1}^{\min(m,n)} \sigma_i(\mathbf{X})$ ($\mathbf{X} \in \mathbb{R}^{m \times n}$)

- By Von Neumann's trace inequality^[1], we get $\text{Tr}(\mathbf{AXB}^T) = \text{Tr}(\mathbf{XB}^T \mathbf{A}) \leq \sum_{i=1}^r \sigma_i(\mathbf{X}).$

$$\text{Tr}(\mathbf{AXB}^T) = \text{Tr}(\mathbf{XB}^T \mathbf{A}) \leq \sum_{i=1}^{\min(m,n)} \sigma_i(\mathbf{X}) \sigma_i(\mathbf{B}^T \mathbf{A}).$$

$$\begin{aligned} & \sum_{i=1}^{\min(m,n)} \sigma_i(\mathbf{X}) \sigma_i(\mathbf{B}^T \mathbf{A}) \\ &= \sum_{i=1}^s \sigma_i(\mathbf{X}) \sigma_i(\mathbf{B}^T \mathbf{A}) + \sum_{i=s+1}^{\min(m,n)} \sigma_i(\mathbf{X}) \sigma_i(\mathbf{B}^T \mathbf{A}) \\ &= \sum_{i=1}^s \sigma_i(\mathbf{X}) \cdot 1 + \sum_{i=s+1}^{\min(m,n)} \sigma_i(\mathbf{X}) \cdot 0 \\ &= \sum_{i=1}^s \sigma_i(\mathbf{X}) \leq \sum_{i=1}^r \sigma_i(\mathbf{X}) \quad (s \leq r \text{ and } \sigma_i(\mathbf{X}) \geq 0) \end{aligned}$$

Suppose, $\mathbf{U}\Sigma\mathbf{V}^T$ is the singular value decomposition of \mathbf{X} , where $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_m) \in \mathbb{R}^{m \times m}$, $\Sigma \in \mathbb{R}^{m \times n}$, and $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_m) \in \mathbb{R}^{n \times n}$,

$$\mathbf{A} = (\mathbf{u}_1, \dots, \mathbf{u}_m)^T, \mathbf{B} = (\mathbf{v}_1, \dots, \mathbf{v}_m)^T.$$

$$\begin{aligned} & \text{Tr}(\mathbf{AXB}^T) \\ &= \text{Tr}((\mathbf{u}_1, \dots, \mathbf{u}_m)^T \mathbf{X} (\mathbf{v}_1, \dots, \mathbf{v}_m)) \\ &= \text{Tr}(((\mathbf{u}_1, \dots, \mathbf{u}_m)^T \mathbf{U}) \Sigma (\mathbf{V}^T (\mathbf{v}_1, \dots, \mathbf{v}_m))) \\ &= \text{Tr} \left(\begin{bmatrix} \mathbf{I}_r & 0 \\ 0 & 0 \end{bmatrix} \Sigma \begin{bmatrix} \mathbf{I}_r & 0 \\ 0 & 0 \end{bmatrix} \right) \\ &= \text{Tr}(diag(\sigma_1(\mathbf{X}), \dots, \sigma_r(\mathbf{X}), 0, \dots, 0)) \\ &= \sum_{i=1}^r \sigma_i(\mathbf{X}) \end{aligned}$$

[1] Von Neumann, J, "Some matrix-inequalities and metrization of metric-space," Tomsk Univ. Rev. 1, 286–300, 1937.

• 3. Truncated Nuclear Norm Regularization (TNNR)

$$\max_{\mathbf{A}\mathbf{A}^T = \mathbf{I}, \mathbf{B}\mathbf{B}^T = \mathbf{I}} \text{Tr}(\mathbf{AXB}^T) \leq \sum_{i=1}^r \sigma_i(\mathbf{X}) \quad \|\mathbf{X}\|_* - \max_{\mathbf{A}\mathbf{A}^T = \mathbf{I}, \mathbf{B}\mathbf{B}^T = \mathbf{I}} \text{Tr}(\mathbf{AXB}^T) = \sum_{i=1}^{\min(m,n)} \sigma_i(\mathbf{X}) - \sum_{i=1}^r \sigma_i(\mathbf{X}) = \|\mathbf{X}\|_r$$

$$\begin{array}{lll} \min_{\mathbf{X}} \|\mathbf{X}\|_r & \Rightarrow & \min_{\mathbf{X}} \|\mathbf{X}\|_* - \max_{\mathbf{A}\mathbf{A}^T = \mathbf{I}, \mathbf{B}\mathbf{B}^T = \mathbf{I}} \text{Tr}(\mathbf{AXB}^T) \\ s.t. \quad \mathcal{P}_{\Omega}(\mathbf{X}) = \mathcal{P}_{\Omega}(\mathbf{M}) & & \Rightarrow \min_{\mathbf{X}} \|\mathbf{X}\|_* - \text{Tr}(\mathbf{AXB}^T) \\ & & s.t. \quad \mathcal{P}_{\Omega}(\mathbf{X}) = \mathcal{P}_{\Omega}(\mathbf{M}) \end{array}$$

Algorithm 1 The Proposed Two-Step Approach for Sovling (6)

Input: original incomplete data matrix \mathbf{M}_{Ω} , where Ω is the indices of the observed entries, tolerance ϵ_0

Initialization: $\mathbf{X}_1 = \mathcal{P}_{\Omega}(\mathbf{M})$.

repeat

step 1. Given \mathbf{X}_l , compute $\mathbf{U}_l, \Sigma_l, \mathbf{V}_l = \text{svd}(\mathbf{X}_l)$,

where $\mathbf{U}_l = (\mathbf{u}_1, \dots, \mathbf{u}_m) \in \mathbb{R}^{m \times m}$, $\mathbf{V}_l = (\mathbf{v}_1, \dots, \mathbf{v}_m) \in \mathbb{R}^{n \times n}$.

Compute \mathbf{A}_l and \mathbf{B}_l , as $\mathbf{A}_l = (\mathbf{u}_1, \dots, \mathbf{u}_m)^T$, $\mathbf{B}_l = (\mathbf{v}_1, \dots, \mathbf{v}_m)^T$

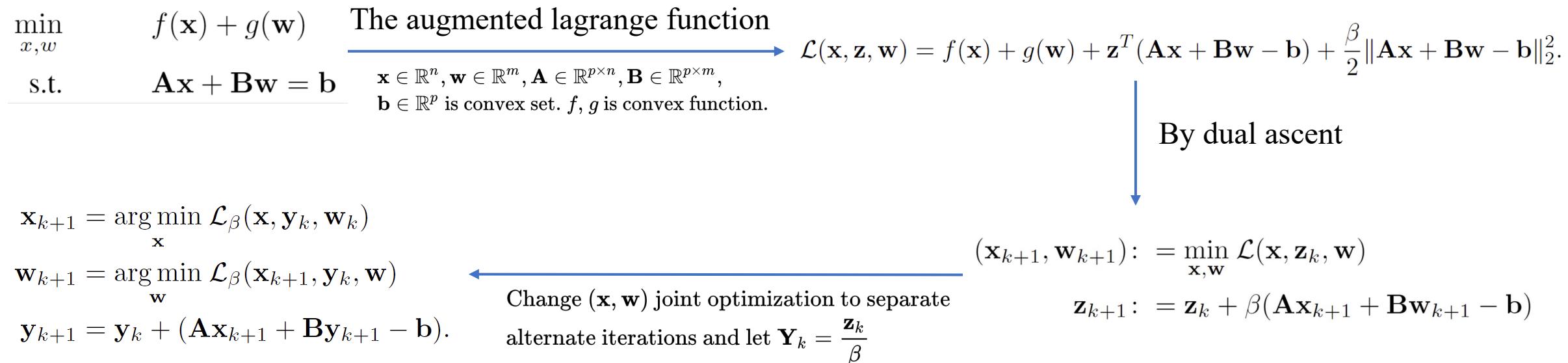
step 2. Solve $\mathbf{X}_{l+1} = \arg \min_{\mathbf{X}} \|\mathbf{X}\|_* - \text{Tr}(\mathbf{AXB}^T)$ s.t. $\mathcal{P}_{\Omega}(\mathbf{X}) = \mathcal{P}_{\Omega}(\mathbf{M})$

until $\|\mathbf{X}_{l+1} - \mathbf{X}_l\|_F \leq \epsilon_0$.

return the recovered matrix.

• 4. Optimization ways – ADMM

- Alternating Direction Method of Multipliers (ADMM)^[1] is an algorithm that is intended to blend the decomposability of dual ascent with the superior convergence properties of the method of multipliers.
- If the optimization problem is separable, we can use ADMM to solve. The ADMM algorithm is suitable for solving the following 2-block (or N-block) convex optimization problems.



[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers,” Foundations and Trends in Machine Learning, vol. 3, 2010.

• 4. Optimization ways – ADMM

- Rewrite the objective function:

$$\begin{array}{lll} \min_{\mathbf{X}} & \|\mathbf{X}\|_* - \text{Tr}(\mathbf{AXB}^T) \\ \text{s.t.} & \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{M}). \end{array} \Rightarrow \begin{array}{lll} \min_{\mathbf{X}, \mathbf{W}} & \|\mathbf{X}\|_* - \text{Tr}(\mathbf{A}_l \mathbf{XB}_l^T) \\ \text{s.t.} & \mathbf{X} = \mathbf{W}, \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{M}). \end{array}$$

- The augmented Lagrange function: $L(\mathbf{X}, \mathbf{Y}, \mathbf{W}, \beta) = \|\mathbf{X}\|_* - \text{Tr}(\mathbf{A}_l \mathbf{XB}_l^T) + \frac{\beta}{2} \|\mathbf{X} - \mathbf{W}\|_F^2 + \text{Tr}(\mathbf{Y}^T (\mathbf{X} - \mathbf{W}))$

- Iteration function:

$$\begin{aligned} \mathbf{X}_{k+1} &= \arg \min_{\mathbf{X}} L(\mathbf{X}, \mathbf{Y}_k, \mathbf{W}_k, \beta) \\ &= \arg \min_{\mathbf{X}} \|\mathbf{X}\|_* - \text{Tr}(\mathbf{A}_l \mathbf{W}_k \mathbf{B}_l^T) + \frac{\beta}{2} \|\mathbf{X} - \mathbf{W}_k\|_F^2 + \text{Tr}(\mathbf{Y}_k^T (\mathbf{X} - \mathbf{W}_k)) \end{aligned}$$

- Singular Value Shrinkage operator^[1]:

$$\mathcal{D}_\tau : \quad \mathcal{D}_\tau(\mathbf{X}) = \mathbf{U} \mathcal{D}_\tau(\Sigma) \mathbf{V}^T, \quad \mathcal{D}_\tau(\Sigma) = \text{diag}(\max\{\sigma_i - \tau, 0\}).$$

For each $\tau \geq 0$ and $\mathbf{Y} \in \mathbb{R}^{m \times n}$, we have

$$\mathcal{D}_\tau(\mathbf{X}) = \arg \min_{\mathbf{X}} \frac{1}{2} \|\mathbf{X} - \mathbf{Y}\|_F^2 + \tau \|\mathbf{X}\|_*.$$

Algorithm 2 The Optimization using TNNR-ADMM

Input: $\mathbf{A}_l, \mathbf{B}_l, \mathbf{M}_\Omega$, and tolerance ϵ are given.

Initialization: $\mathbf{X}_1 = \mathbf{M}_\Omega$, $\mathbf{W}_1 = \mathbf{X}_1$, $\mathbf{Y}_1 = \mathbf{X}_1$, and $\beta = 1$.

repeat

step 1. $\mathbf{X}_{k+1} = \mathcal{D}_{\frac{1}{\rho}}(\mathbf{W}_k - \frac{1}{\rho} \mathbf{Y}_k).$

step 2. $\mathbf{W}_{k+1} = \mathbf{X}_{k+1} + \frac{1}{\beta} (\mathbf{A}_l^T \mathbf{B}_l + \mathbf{Y}_k).$

Fix values at observed entries, $\mathbf{W}_{k+1} = \mathbf{X}_{k+1} + \frac{1}{\beta} (\mathbf{A}_l^T \mathbf{B}_l + \mathbf{Y}_k).$

step 3. $\mathbf{Y}_{k+1} = \mathbf{Y}_k + \beta(\mathbf{X}_{k+1} - \mathbf{W}_{k+1}).$

until $\|\mathbf{X}_{k+1} - \mathbf{X}_k\|_F \leq \epsilon$.

[1] J.F. Cai, E.J. Candès, and Z. Shen, “A Singular Value Thresholding Algorithm for Matrix Completion,” SIAM J. Optimization, vol. 20, pp. 1956-1982, 2010.

• 4. Optimization ways -- APGL

- Accelerated Proximal Gradient Line (APGL), also called as iterative shrinkage-thresholding algorithm (FISTA)^[1]. It is also a common approach to solve the 2-block convex optimization problems.

$$\min_{\mathbf{X}} g(\mathbf{X}) + f(\mathbf{X})$$

g is a closed, convex, possibly nondifferentiable function
 f is a convex and differentiable function

- The APGL method constructs an approximation of $f(\mathbf{X})$ at a given point \mathbf{Y} as

$$Q(\mathbf{X}, \mathbf{Y}) = f(\mathbf{Y}) + \langle \mathbf{X} - \mathbf{Y}, \nabla f(\mathbf{Y}) \rangle + \frac{1}{2t} \|\mathbf{X} - \mathbf{Y}\|_F^2 + g(\mathbf{X}).$$

- Above optimization problem by iteratively updating \mathbf{X} , \mathbf{Y} , and t .

$$\begin{aligned} \mathbf{X}_{k+1} &= \arg \min_{\mathbf{X}} Q(\mathbf{X}, \mathbf{Y}_k) \\ &= \arg \min_{\mathbf{X}} g(\mathbf{X}) + \frac{1}{2t} \|\mathbf{X} - (\mathbf{Y}_k - t_k \nabla f(\mathbf{Y}_k))\|_F^2 \end{aligned}$$

[1] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," SIAM J. Imaging Sciences, vol. 2, no. 1, pp. 183-202, 2009.

• 4. Optimization ways -- APGL

- Relax the constrained problem:

$$\min \|\mathbf{X}\|_* - \text{Tr}(\mathbf{A}_l \mathbf{X} \mathbf{B}_l^T) + \frac{\lambda}{2} \|\mathbf{X}_{\Omega} - \mathbf{M}_{\Omega}\|_F^2$$

$$g(\mathbf{X}) = \|\mathbf{X}\|_* \quad f(\mathbf{X}) = -\text{Tr}(\mathbf{A}_l \mathbf{X} \mathbf{B}_l^T) + \frac{\lambda}{2} \|\mathbf{X}_{\Omega} - \mathbf{M}_{\Omega}\|_F^2$$

- Iteration function:

$$\begin{aligned}\mathbf{X}_{k+1} &= \arg \min_{\mathbf{X}} \|\mathbf{X}\|_* + \frac{1}{2t} \|\mathbf{X} - (\mathbf{Y}_k - t_k \nabla f(\mathbf{Y}_k))\|_F^2 \\ &= \mathcal{D}_{t_k}(\mathbf{Y}_k - t_k \nabla f(\mathbf{Y}_k)) \\ &= \mathcal{D}_{t_k}(\mathbf{Y}_k - t_k(\mathbf{A}_l^T \mathbf{B}_l - \lambda(\mathcal{P}_{\Omega}(\mathbf{Y}_k) - \mathcal{P}_{\Omega}(\mathbf{M}))))\end{aligned}$$

$$\begin{aligned}t_{k+1} &= \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \\ \mathbf{Y}_{k+1} &= \mathbf{X}_{k+1} + \frac{t_k - 1}{t_{k+1}}(\mathbf{X}_{k+1} - \mathbf{X}_k)\end{aligned}$$

Algorithm 3 The Optimization using TNNR-APGL

Input: \mathbf{A}_l , \mathbf{B}_l , \mathbf{M}_{Ω} , and tolerance ϵ are given.

Initialization: $t_1 = 1$, $\mathbf{X}_1 = \mathbf{M}_{\Omega}$, $\mathbf{Y}_1 = \mathbf{X}_1$.

repeat

step 1. $\mathbf{X}_{k+1} = \mathcal{D}_{t_k}(\mathbf{Y}_k - t_k(\mathbf{A}_l^T \mathbf{B}_l - \lambda(\mathcal{P}_{\Omega}(\mathbf{Y}_k) - \mathcal{P}_{\Omega}(\mathbf{M}))))$.

step 2. $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$.

step 3. $\mathbf{Y}_{k+1} = \mathbf{X}_{k+1} + \frac{t_k - 1}{t_{k+1}}(\mathbf{X}_{k+1} - \mathbf{X}_k)$.

until $\|\mathbf{X}_{k+1} - \mathbf{X}_k\|_F \leq \epsilon$.

• 4. Optimization ways -- ADMMAP

- Problem of ADMM
 - The convergence of ADMM becomes slower with more constraints^[1].
 - It is hard to choose an optimal penalty parameter in advance (improper penalty may lead to divergence).
- The authors purpose the approach named ADMM with adaptive penalty (TNNR-ADMMAP) to solve the problem.
 - New transform variant of ADMM: Deal with all the constraints together to accelerate the convergence
 - Adaptive update strategy for the penalty parameter^[2]: further speed up the convergence
- Reformulate the objective function:

$$\begin{array}{ll} \min_{\mathbf{X}} & \|\mathbf{X}\|_* - \text{Tr}(\mathbf{A}_l \mathbf{X} \mathbf{B}_l^T) \\ \text{s.t.} & \mathcal{A}(\mathbf{X}) + \mathcal{B}(\mathbf{M}) = \mathcal{C}, \end{array} \quad \mathcal{A}(\mathbf{X}) = \begin{pmatrix} \mathbf{X} & 0 \\ 0 & 0 \end{pmatrix}, \quad \mathcal{B}(\mathbf{X}) = \begin{pmatrix} -\mathbf{W} & 0 \\ 0 & \mathcal{P}_{\Omega}(\mathbf{X}) \end{pmatrix}, \quad \mathcal{C} = \begin{pmatrix} 0 & 0 \\ 0 & \mathcal{P}_{\Omega}(\mathbf{M}) \end{pmatrix}$$

[1] B. He, M. Tao, and X. Yuan, “Alternating Direction Method with Gaussian Back Substitution for Separable Convex Programming,” SIAM J. Optimization, vol. 22, no. 2, pp. 313-340, 2012.

[2] Z. Lin, R. Liu, and Z. Su, “Linearized Alternating Direction Method with Adaptive Penalty for Low-Rank Representation,” Proc. Advances in Neural Information Processing Systems, 2011.

• 4. Optimization ways -- ADMMAP

- The augmented Lagrange function:

$$\mathcal{L}_{AP}(\mathbf{X}, \mathbf{W}, \mathbf{Y}, \beta) = -\text{Tr}(\mathbf{A}_l \mathbf{W} \mathbf{B}_l^T) + \langle \mathbf{Y}, \mathcal{A}(\mathbf{X}) + \mathcal{B}(\mathbf{W}) - \mathcal{C} \rangle + \|\mathbf{X}\|_* + \frac{\beta}{2} \|\mathcal{A}(\mathbf{X}) + \mathcal{B}(\mathbf{W}) - \mathcal{C}\|_F^2$$

- Iteration function:

$$\mathbf{X}_{k+1} = \arg \min_{\mathbf{X}} \mathcal{L}_{AP}(\mathbf{X}, \mathbf{W}_k, \mathbf{Y}_k, \beta)$$

$$= \arg \min_{\mathbf{X}} \frac{\beta}{2} \|\mathcal{A}(\mathbf{X}) + \mathcal{B}(\mathbf{W}_k) - \mathcal{C} + \frac{1}{\beta} \mathbf{Y}_k\|_F^2 + \|\mathbf{X}\|_*$$

$$\mathbf{W}_{k+1} = \arg \min_{\mathbf{W}} \mathcal{L}_{AP}(\mathbf{X}_{k+1}, \mathbf{W}, \mathbf{Y}_k, \beta)$$

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k + \beta[\mathcal{A}(\mathbf{X}_{k+1}) + \mathcal{B}(\mathbf{W}) - \mathcal{C}]$$

$$\beta_{k+1} = \min(\beta_{\max}, \rho \beta_k) \quad (54)$$

$$\rho = \begin{cases} \rho_0, & \text{if } \frac{\beta_k \max\{\|\mathbf{X}_{k+1} - \mathbf{X}_k\|_F\}}{\|\mathcal{C}\|_F} < \kappa \\ 1, & \text{otherwise.} \end{cases} \quad (55)$$

Algorithm 4 Inner Optimization by ADMMAP

Input: $\mathbf{A}_l, \mathbf{B}_l, \mathbf{M}_\Omega$, and tolerance ϵ are given.

Initialization: $\mathbf{X}_1 = \mathbf{M}_\Omega, \mathbf{Y}_1 = \mathbf{X}_1$.

repeat

step 1. $\mathbf{X}_{k+1} = \mathcal{D} \left(\mathbf{W}_k - \frac{1}{\beta} (\mathbf{Y}_k)_{11} \right)$.

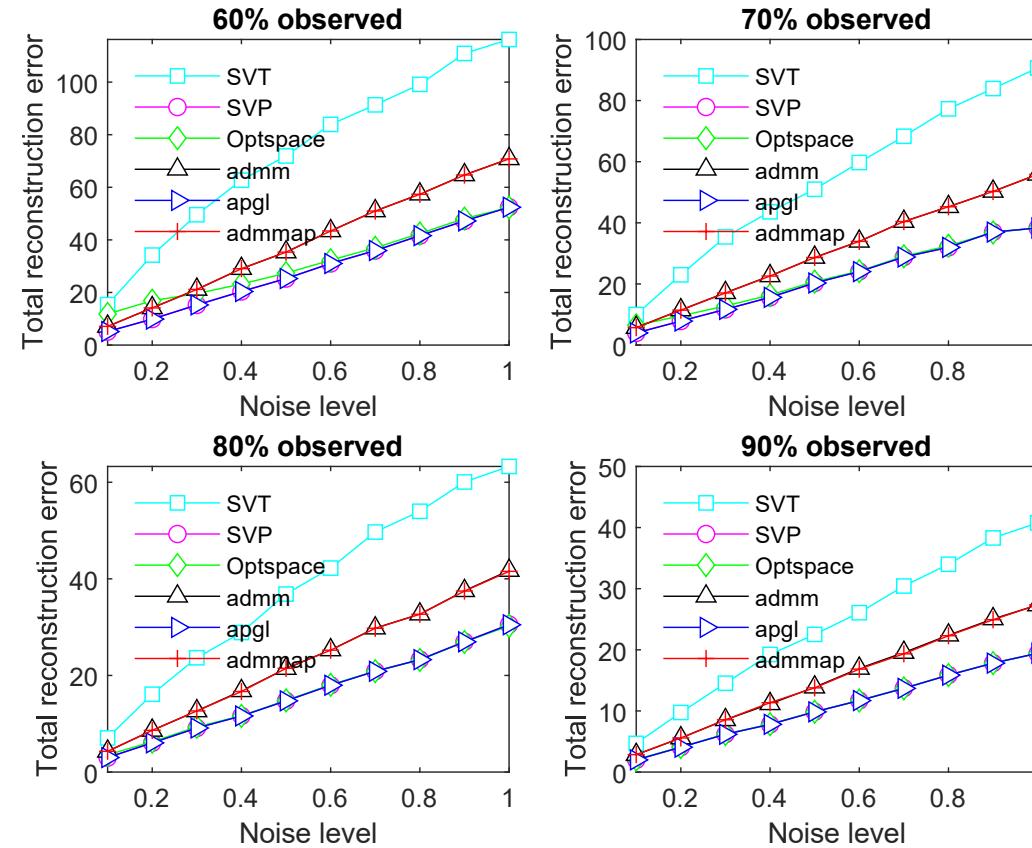
step 2. $\mathbf{W}_{k+1} = \frac{1}{2\beta} \mathcal{P}_\Omega [\beta(\mathbf{M} - \mathbf{X}_{k+1}) - (\mathbf{A}_l^T \mathbf{B}_l + (\mathbf{Y}_k)_{11} + (\mathbf{Y}_k)_{22})] + \mathbf{X}_{k+1} + \frac{1}{\beta} (\mathbf{A}_l^T \mathbf{B}_l + (\mathbf{Y}_k)_{11})$.

step 3. $\mathbf{Y}_{k+1} = \mathbf{Y}_k + \beta[\mathcal{A}(\mathbf{X}_{k+1}) + \mathcal{B}(\mathbf{W}) - \mathcal{C}]$.

step 4. Update β_k by (54) and (55)

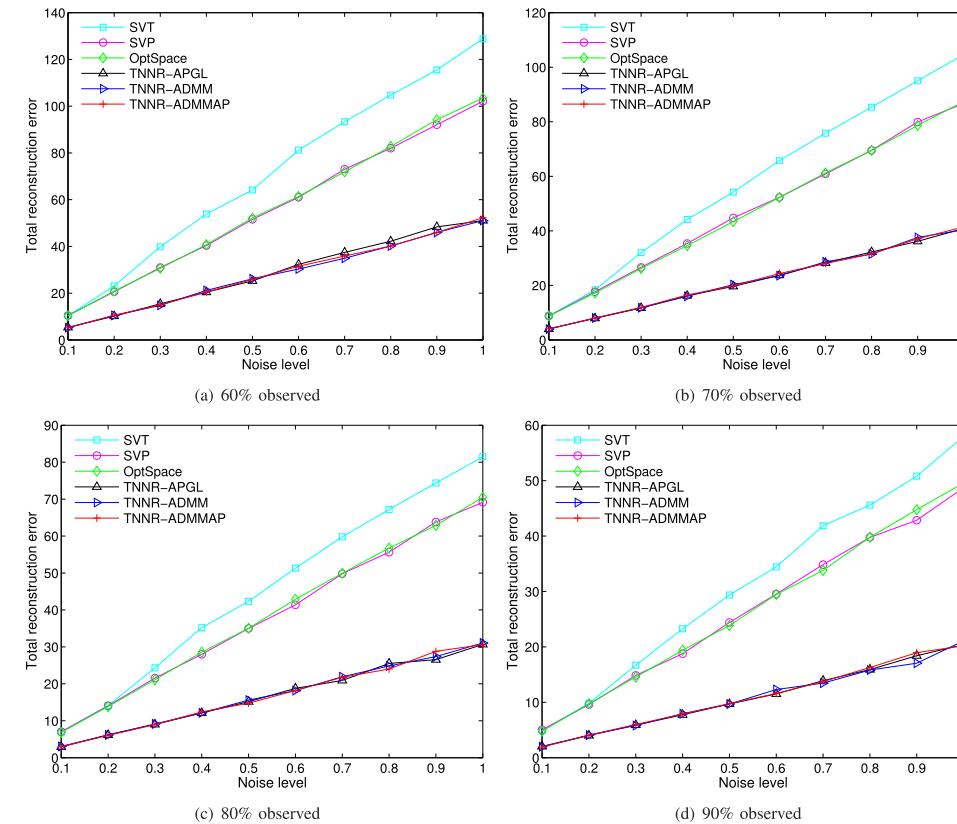
until $\|\mathbf{X}_{k+1} - \mathbf{X}_k\|_F \leq \epsilon$.

• 5. Results – Reconstruction error



The reproduction of Fig. 2.

Data Generation: $\mathbf{B} = \mathbf{M} + \sigma \mathbf{Z}$, $B_{ij} = M_{ij} + \sigma Z_{ij}$, $(i, j) \in \Omega$
 $\mathbf{M} = \mathbf{M}_L \mathbf{M}_R$, $\mathbf{M}_L \in \mathbb{R}^{m \times r_0}$, $\mathbf{M}_R \in \mathbb{R}^{r_0 \times n}$
 $\mathbf{M}_L, \mathbf{M}_R, \mathbf{Z}$ is gaussian.



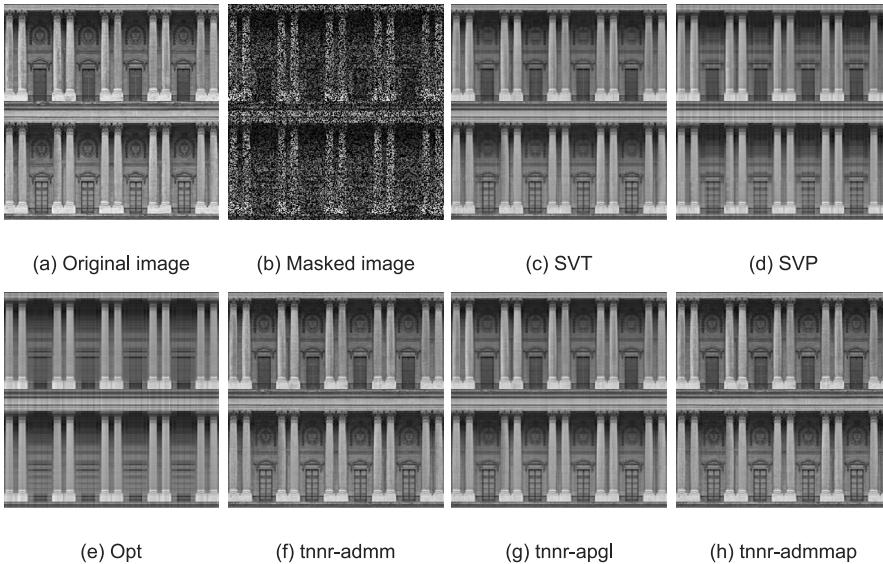
The original of Fig. 2.

Metric:

Total reconstruction error: $\|\mathcal{P}_{\Omega^c}(\mathbf{X}_{sol} - \mathbf{X}_{full})\|_F$

• 5. Results – completion, PSNR, convergence

β	1e-3	r	5
β_{\max}	1e10	ϵ_0	1e-3
ρ_0	1.1	ϵ	1e-4
κ	1e-3	λ	1e-2
p_{SVT}	0.87	τ_{opt}	1e-3
δ_{2k}	0.2	$iter_{\max}$	100

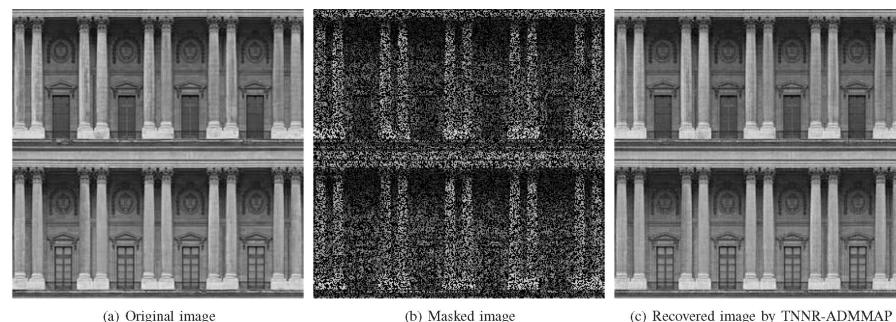


Total number of missing pixels: T

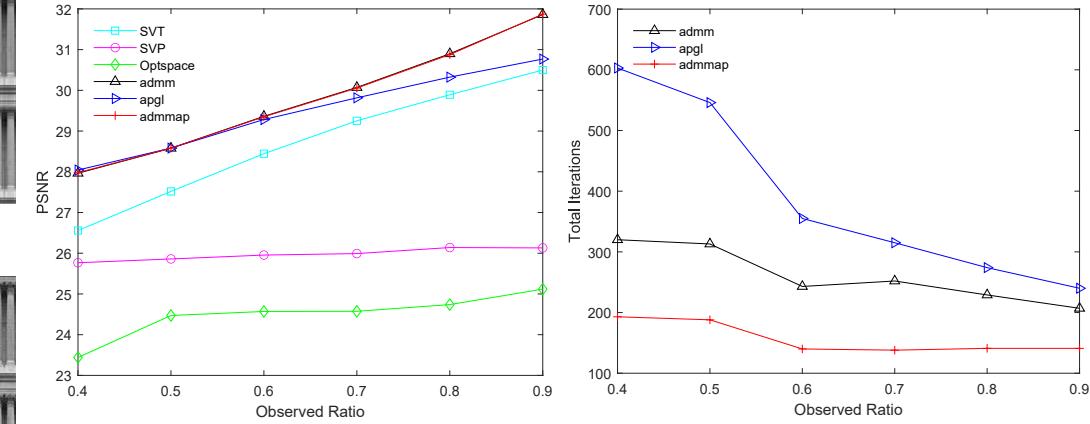
$$MSE = \frac{error_r^2 + error_g^2 + error_b^2}{3T}$$

$$PSNR = 10 \times \log_{10} \left(\frac{255^2}{MSE} \right)$$

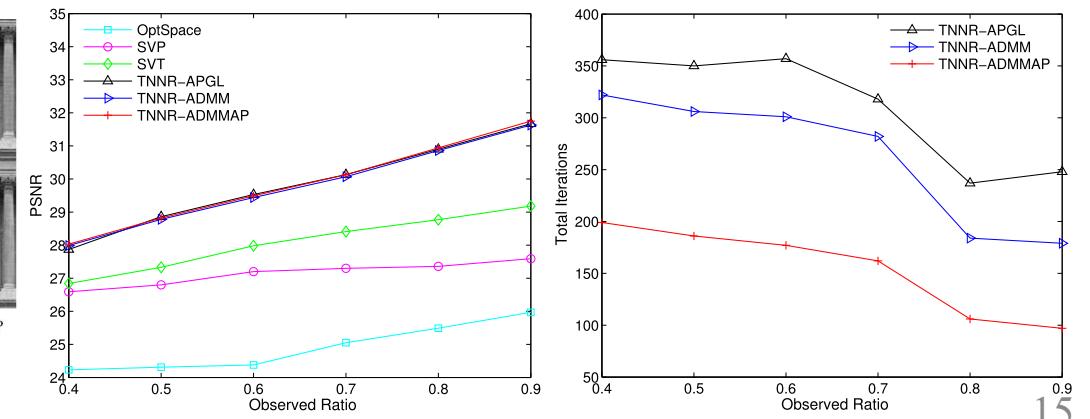
β	1e-3	r	1~20
β_{\max}	1e10	ϵ_0	1e-3
ρ_0	1.9	ϵ	1e-4
κ	1e-3	λ	0.06



The original of Fig. 4, Fig. 5, Fig. 6.



The reproduction of Fig. 4 (completion), Fig. 5 (PSNR), Fig. 6 (convergence).



• 5. Results – Eliminate String

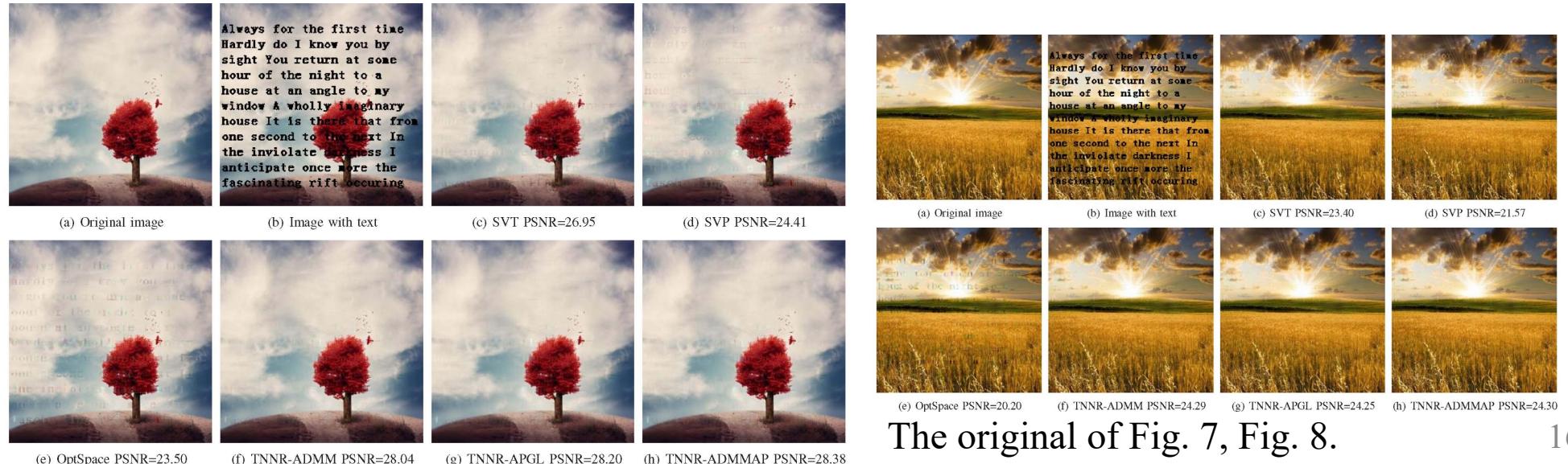
β	1e-3	r	3~15
β_{\max}	1e10	ϵ_0	1e-3
ρ_0	1.01	ϵ	1e-4
κ	1e-3	λ	1e-3
p_{SVT}	0.8	τ_{opt}	1e-3
δ_{2k}	0.2	$iter_{\max}$	1e2/1e3



Total number of missing pixels: T

$$MSE = \frac{error_r^2 + error_g^2 + error_b^2}{3T}$$

$$PSNR = 10 \times \log_{10} \left(\frac{255^2}{MSE} \right)$$

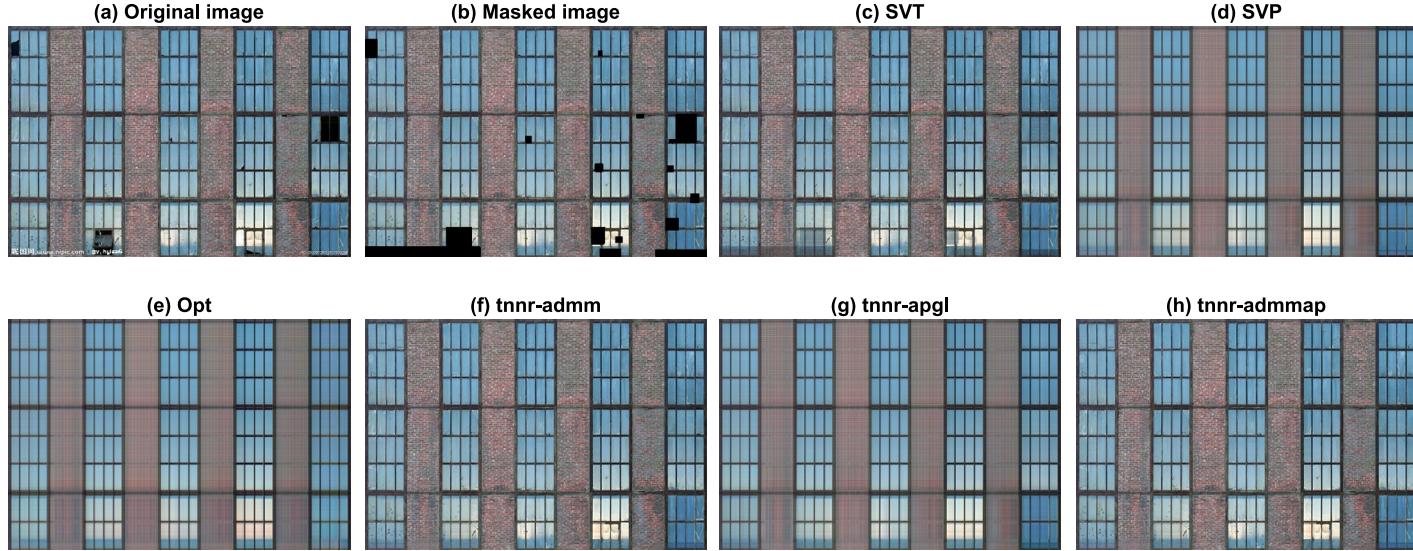


The original Parameter Setting is equal to Fig. 4-6.

The reproduction of Fig. 7, Fig. 8.

The original of Fig. 7, Fig. 8.

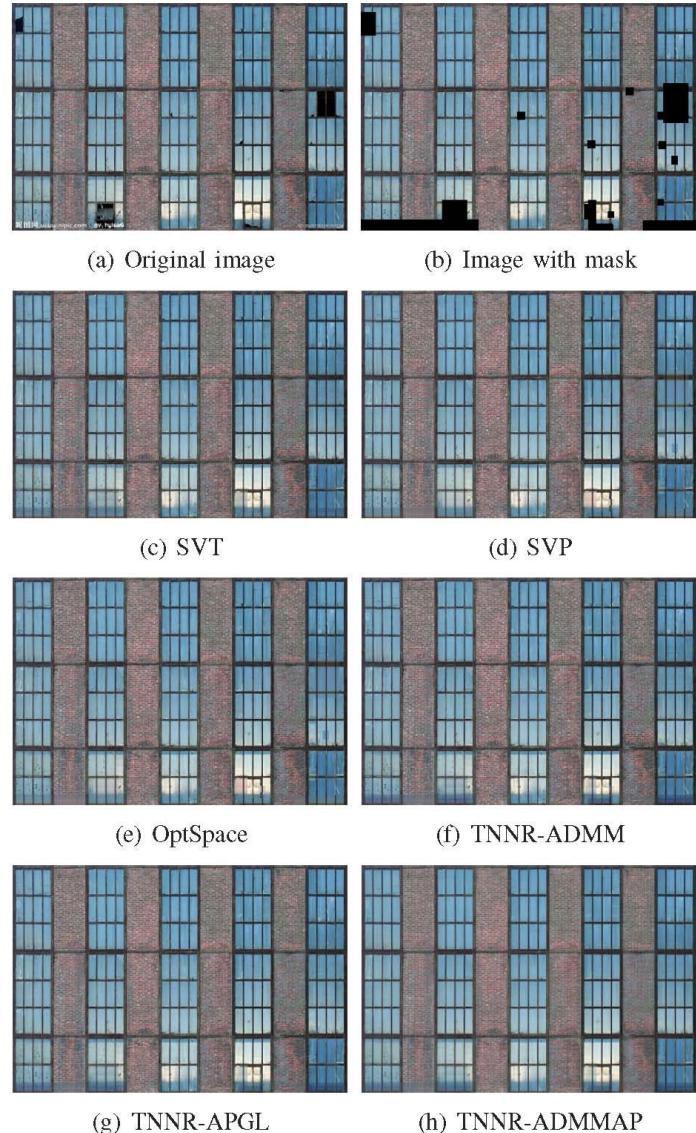
• 5. Results – Eliminate Blocks



The reproduction of Fig. 9.

β	1e-3	r	3	p_{SVT}	0.8
β_{\max}	1e10	ϵ_0	1e-3	τ_{opt}	1e-2
ρ_0	1.01	ϵ	1e-4	δ_{2k}	0.2
κ	1e-3	λ	1e-3	$iter_{\max}$	1e2/1e3

The original Parameter Setting is equal to Fig. 4-6.



The original of Fig. 9.

Thanks for
listening