

Assignment 4 - Image Reconstruction with Tensor Completion

Mihaly Fey (5476747), Piotr Grabowski (6229913), Luka Caric(6179150)

I. INTRODUCTION

The project goal is to recover a complete image from a subset of its pixels. Without additional assumptions this inverse problem is ill-posed, as infinitely many matrices can fit the same subset of observed entries. A common assumption is that natural images are approximately *low-rank*, since pixel intensities exhibit strong spatial and chromatic correlations.

This assumption allows the problem to be formulated as a rank minimization task:

$$\min_X \text{rank}(X) \quad \text{s.t. } X_{ij} = M_{ij}, \quad (i, j) \in \Omega, \quad (1)$$

where Ω is the set of observed pixel indices. The non-convex rank function is replaced by its convex surrogate, the nuclear norm $\|X\|_* = \sum_i \sigma_i(X)$, leading to tractable algorithms such as SVT [1] and TNRR [2].

In a grayscale, an image can be represented by a matrix $M \in \mathbb{R}^{M \times N}$. However, as image normally (with RGB) uses a third-order tensor $T \in \mathbb{R}^{M \times N \times 3}$, a tensor completion approach is needed to model correlations between RGB channels.

For implemented methods, a low-rank is required. Therefore to verify it, a singular value decomposition (SVD) was applied to each color channel separately. As can be seen on ?? more than 95% of the total energy is cumulated in the first few singular values which positively verifies: $k \ll \min(512, 512)$. Also, to get >99% rank, the most around 20% of image dimension 512 is needed. It confirms that the images are low-rank.

II. MATRIX COMPLETION METHOD

For matrix completion, the Singular Value Thresholding Algorithm for was chosen - proposed in topic description. Version implemented in [3] was used. The SVT algorithm uses soft-thresholding operator \mathcal{D}_τ defined as:

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

where

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}, \quad \Sigma = \text{diag}(\{\sigma_i\}_{1 \leq i \leq r})$$

is the singular value decomposition of a matrix X of rank r , σ_i the i -th singular value and t_+ the positive part of t , which means that the soft thresholding operator keeps the singular values larger than a certain threshold τ .

The algorithm is initialized with $\mathbf{Y}^0 = 0$, \mathbf{Y} being the same size as \mathbf{X} , $\tau > 0$ is fixed as well as sequence $\{\delta_k\}_{k \geq 1}$ of scalar step sizes. The k -th iteration of the algorithm defines:

$$\begin{cases} \mathbf{X}^k = \mathcal{D}_\tau(\mathbf{Y}^{k-1}) \\ \mathbf{Y}^k = \mathbf{Y}^{k-1} + \delta_k (\mathbf{M}_\Omega - \mathbf{X}_\Omega^k) \end{cases} \quad (2)$$

The algorithm is stopped when

$$\frac{\|\mathbf{M}_\Omega - \mathbf{X}_\Omega^k\|_F}{\|\mathbf{M}_\Omega\|_F} \leq \epsilon \quad (3)$$

where ϵ is a fixed tolerance chosen in advance.

III. MOTIVATION FOR TENSOR METHOD

Although matrix completion is well-studied with mature algorithms and theory, directly applying these methods to tensors often fails to preserve their multi-way structure and inter-mode redundancy. Tensor completion, therefore, extends beyond simple matrix approaches, emphasizing techniques that maintain the inherent multidimensional relationships within data. Compared to matrix completion, tensor completion faces higher computational and storage complexity, especially with large-scale or high-order data. The main motivation for using tensor completion is to get a more accurate and faster reconstruction compared to using matrix completion of a tensor matricization.

IV. TENSOR CHOICE

Paper [4] gives us a good overview and comparison between tensor completion algorithms. From the families of tensor completion methods discussed there, the nuclear-norm-based (SNN) group was selected. In comparison to decomposition-based methods such as CP or Tucker, SNN does not require explicit rank estimation, is also robust and parameter-free. This makes it perfect choice for RGB image data where the true tensor rank is unknown. Similarly to the matrix case where we used the nuclear norm to get the lower bound of the matrix rank and got a convex problem, it can be generalized to the tensor case. Once again, the goal is to minimize the tensor rank, a non-convex function. As in the matrix case, the trace norm was used to approximate the rank, a generalization of the trace norm to a higher order is used in the tensor case. This is a weighted sum of the trace norms of the n-mode matricizations of the tensor

as can be seen in Equation 4. $\sum_{n=1}^N \alpha_n = 1$ is to maintain consistency with the matrix trace norm, where (and in the following paragraphs) n is the order of the tensor to be reconstructed.

$$\min_{\mathcal{X}} : \sum_{i=1}^n \alpha_i \|\mathcal{X}_{(i)}\|_* \quad (4)$$

s.t. $\mathcal{X}_\Omega = \mathcal{T}_\Omega$

What is very convenient when using the SNN (sum of the nuclear norm) model is that it allows us to solve the problem without predefining the tensor rank. Among the convex tensor completion algorithms introduced by [5], (SiLRTC, FaLRTC, and HaLRTC)

- High-accuracy Low-Rank Tensor Completion was selected. It provides the best trade-off between accuracy and computational efficiency - as reported in [5]. Therefore, this method is particularly well-suited for tasks like image reconstruction. Using ADMM requires the definition of the augmented Lagrangian and a decomposition of the problem into two subproblems. The augmented Lagrangian can be found in Equation 5. The primal variables are \mathcal{X} and $\mathcal{M}_i (i = 1..n)$ and the dual variables are $\mathcal{Y}_i (i = 1..n)$, all are the same dimension and order as the original tensor to be reconstructed.

Each optimization step of ADMM consists of three steps of updating the first and second set of primal, and the dual variables. For minimizing \mathcal{L}_ρ with respect to \mathcal{M}_i , Equation 6 is to be solved for which the solution is given in Equation 7. Then, to minimize \mathcal{L}_ρ with respect to \mathcal{X} , the closed form solution is given by Equation 8 and the update step of \mathcal{Y}_i is given by Equation 9. Note that only the unconstrained part of \mathcal{X} , $\mathcal{X}_{\bar{\Omega}}$ is being updated every time, the rest is initialized with the constrained values \mathcal{T}_Ω . The algorithm could be stopped by a convergence threshold, but in the case of this implementation it was stopped after a fixed number of iterations, as runtime is not a concern.

$$L_\rho(\mathcal{X}, \mathcal{M}_1, \dots, \mathcal{M}_n, \mathcal{Y}_1, \dots, \mathcal{Y}_n) = \sum_{i=1}^n \alpha_i \|\mathcal{M}_{i(i)}\|_* + \langle \mathcal{X} - \mathcal{M}_i, \mathcal{Y}_i \rangle + \frac{\rho}{2} \|\mathcal{M}_i - \mathcal{X}\|_F^2 \quad (5)$$

$$\begin{aligned} \min_{\mathcal{M}_i} : & \frac{\beta_i}{2} \|\mathcal{M}_i - \mathcal{X}_{(i)}\|_F^2 + \alpha_i \|\mathcal{M}_i\|_* \\ & \equiv \frac{1}{2} \|\mathcal{M}_i - \mathcal{X}_{(i)}\|_F^2 + \frac{\alpha_i}{\beta_i} \|\mathcal{M}_i\|_* \end{aligned} \quad (6)$$

$$M_i = \text{fold}_i \left[\mathcal{D}_{\frac{\alpha_i}{\rho}} (\mathcal{X}_{(i)} + \frac{1}{\rho} \mathcal{Y}_{i(i)}) \right] \quad (7)$$

$$\mathcal{X}_{\bar{\Omega}} = \frac{1}{n} \left(\sum_{i=1}^n \mathcal{M}_i - \frac{1}{\rho} \mathcal{Y}_i \right)_{\bar{\Omega}} \quad (8)$$

$$\mathcal{Y}_i = \mathcal{Y}_i - \rho(\mathcal{M}_i - \mathcal{X}) \quad (9)$$

V. IMPLEMENTATION

VI. RESULTS

A. Experimental Setup

Three random observation ratios were tested: 50%, 75%, and 85%.

For all experiments, both matrix and tensor, methods were tested on same images. 3 RGB images with resolution 512x512 were taken from database [6]. For each method same pixels were removed to ensure a fair comparison. Performance was evaluated using the relative Frobenius error (RelErr) and the peak signal-to-noise ratio (PSNR):

$$\text{RelErr} = \frac{\|A - A_{\text{recon}}\|_F}{\|A\|_F}, \quad \text{PSNR} = 10 \log_{10} \left(\frac{1}{\text{MSE}} \right). \quad (10)$$

First, the low-rank assumption was checked for all 3 images. Results are plotted below and they validate the low-rank:

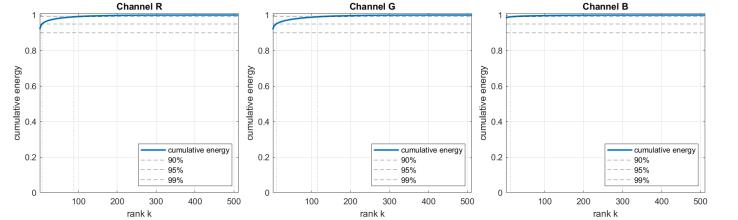


Fig. 1: Verifying low-rank assumption for image 1

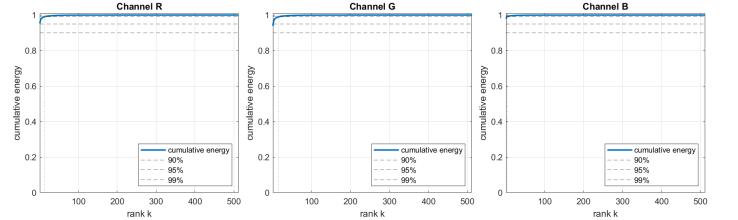


Fig. 2: Verifying low-rank assumption for image 2

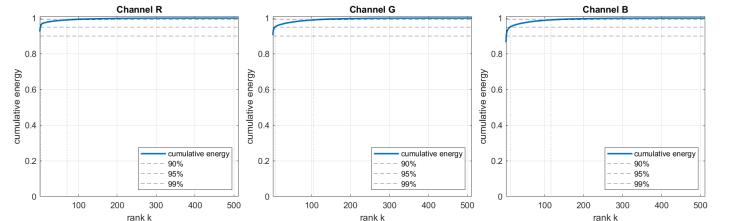


Fig. 3: Verifying low-rank assumption for image 3

Next, the reconstruction for three different images is shown, each corresponding to an observation ratio of 50%, 75%, or 85% with original images and error maps are presented. For fair visual comparison, the same images with same % of observation ratio are shown for different methods.

Matrix results:



Fig. 4: Matrix completion method at 50% observations

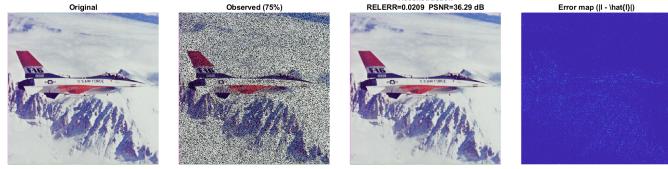


Fig. 5: Matrix completion method at 75% observations

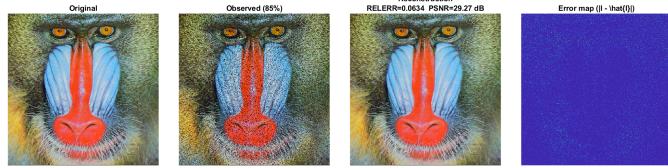


Fig. 6: Matrix completion method at 85% observations

Tensor results:



Fig. 7: Tensor completion reconstruction at 50% observations



Fig. 8: Tensor completion reconstruction at 75% observations

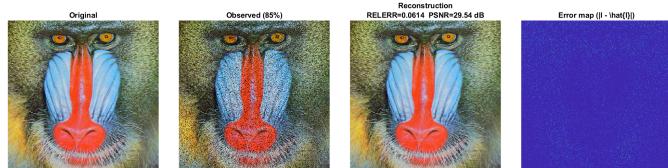


Fig. 9: Tensor completion reconstruction at 85% observations

B. Quantitative Comparison

Figures below present the quantitative results for both matrix and tensor completion methods. Each image

was tested for three observation ratios (50%, 75%, and 85%), and the metrics include the Peak Signal-to-Noise Ratio (PSNR), relative reconstruction error (RelErr) and computation time.

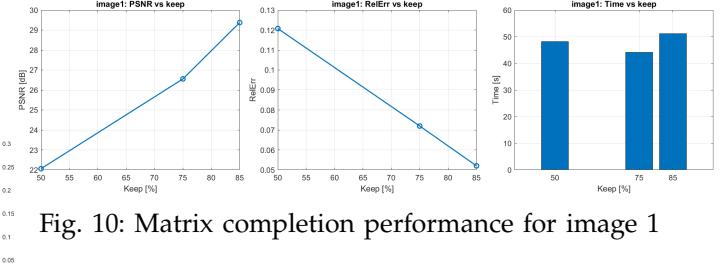


Fig. 10: Matrix completion performance for image 1

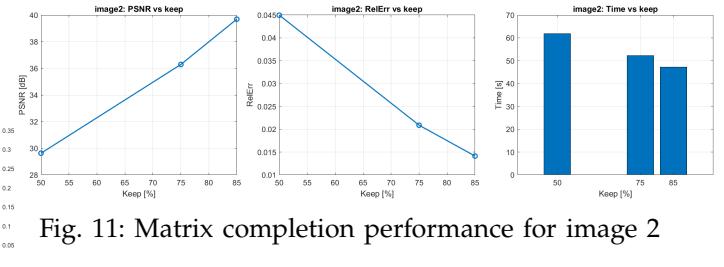


Fig. 11: Matrix completion performance for image 2

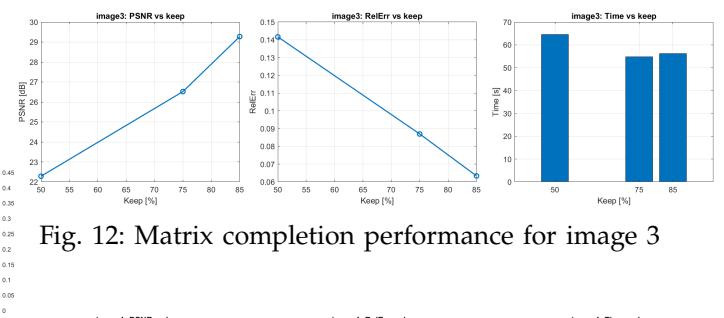


Fig. 12: Matrix completion performance for image 3

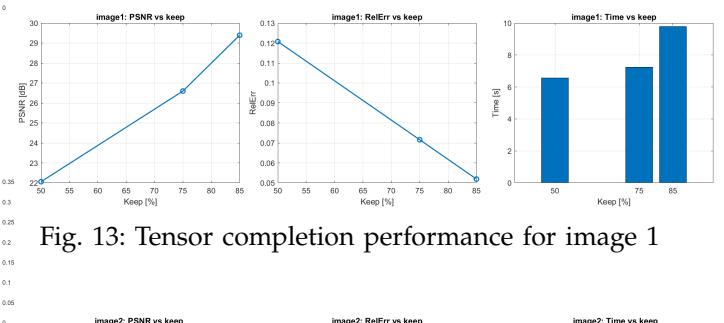


Fig. 13: Tensor completion performance for image 1

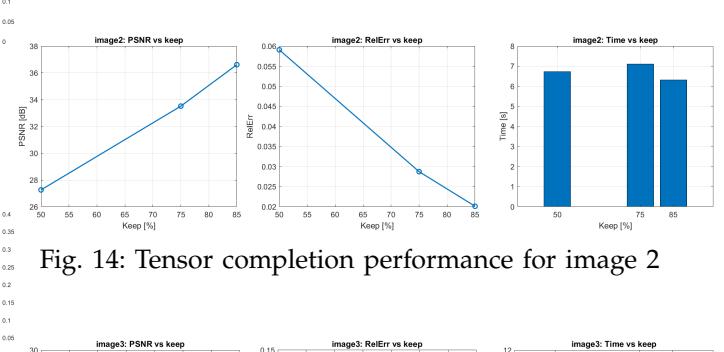


Fig. 14: Tensor completion performance for image 2

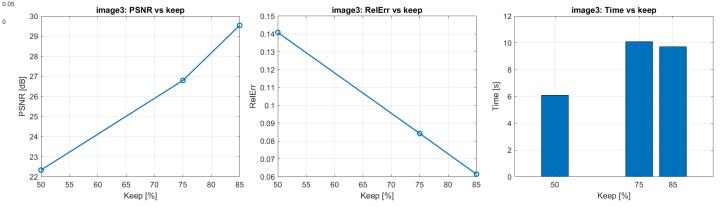


Fig. 15: Tensor completion performance for image 3

As seen from below comparison of performance, both PSNR increases and relative error decreases with the observation ratio - which is expected as problem is easier with more observations. However there are slightly differences between images. For Image2, the matrix method achieves noticeably higher PSNR and lower relative error than Image1 and Image3 - those are very similar. The main reason is probably due to the spatial structure of the images and the random distribution of missing pixels - some regions may more difficult to recover.

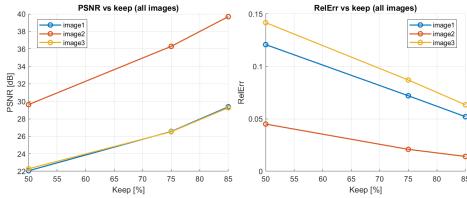


Fig. 16: Comparison of performance for all images - matrix completion

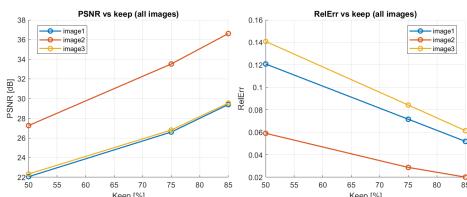


Fig. 17: Comparison of performance for all images - tensor completion

C. Results discussion

From all above figures and table I it can be concluded that for the reconstruction quality (in terms of the relative error and the peak signal-to-noise ratio) for methods demonstrate expected trend -reconstruction quality improves as the observation ratio increases. Higher observation percentages consistently lead to lower RelErr and higher PSNR values which confirms that both algorithms works well as images have low-rank structure.

From table II it can be concluded that matrix completion method achieves slightly better reconstruction accuracy across all images (average PSNR of 29.07 dB and RelErr of 0.0685, compared to 28.23 dB and 0.0710 for tensor method). Probably, this small difference can be explained by the fact that matrix completion processes each color channel independently, potentially achieving more aggressive low-rank approximations. However, results are very similar and hard to spot on images - maybe matrix one provides a bit smoother images.

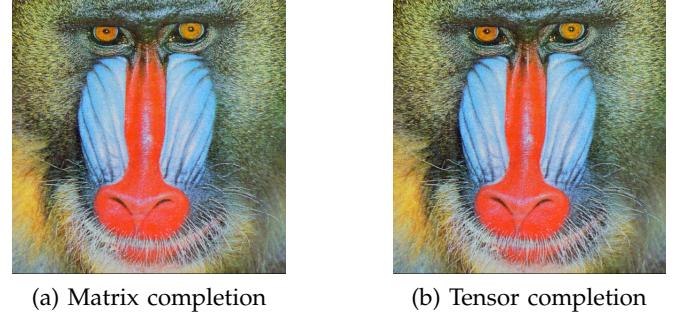
Nevertheless, from table of time measurements it can be concluded that the tensor completion (HalRTC) is significantly faster computationally. The average runtime per image is 7.74 s and for matrix one 53.38 s. That results in an average speed-up of approximately 7× while achieving very similar results. The main explanation is most likely that in tensor method the update

was done for all modes jointly within each ADMM iteration and for matrix method three separate optimization problems had to be solved and then solutions were combined.

D. visual comparison

Below, also reconstructed images were put together for graphical comparison only. As said before, the difference is hard to spot. For some images, matrix completion gives slightly smoother picture upon closer inspection.

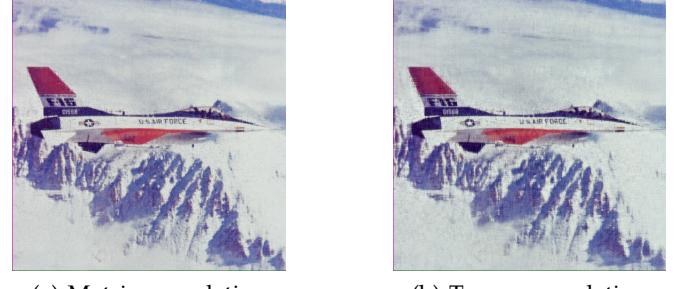
Fig. 18: Image 3, 75% observations



(a) Matrix completion

(b) Tensor completion

Fig. 19: Image 2, 50% observations



(a) Matrix completion

(b) Tensor completion

REFERENCES

- [1] J.-F. Cai, E. J. Candès, and Z. Shen, *A singular value thresholding algorithm for matrix completion*, 2008. arXiv: 0810.3286 [math.OC].
- [2] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, “Fast and accurate matrix completion via truncated nuclear norm regularization,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, pp. 2117–30, Sep. 2013. doi: 10.1109/TPAMI.2012.271.
- [3] A. Majumdar, *Matrix completion via thresholding*, <https://nl.mathworks.com/matlabcentral/fileexchange/26395-matrix-completion-via-thresholding>, 2025.
- [4] Q. Song, H. Ge, J. Caverlee, and X. Hu, “Tensor completion algorithms in big data analytics,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 13, no. 1, pp. 1–48, 2019.
- [5] J. Liu, P. Musialski, P. Wonka, and J. Ye, “Tensor completion for estimating missing values in visual data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2013. doi: 10.1109/TPAMI.2012.39.

TABLE I: Table with results of reconstruction quality in RelErr and PSNR for Matrix vs Tensor

Image	Keep [%]	RelErr (Matrix)	PSNR (Matrix)	RelErr (Tensor)	PSNR (Tensor)
image1	50	0.1207	22.06	0.1208	22.06
image1	75	0.0720	26.55	0.0716	26.60
image1	85	0.0520	29.37	0.0519	29.39
image2	50	0.0450	29.63	0.0591	27.26
image2	75	0.0209	36.29	0.0287	33.52
image2	85	0.0141	39.69	0.0201	36.61
image3	50	0.1417	22.28	0.1408	22.34
image3	75	0.0870	26.52	0.0842	26.80
image3	85	0.0634	29.27	0.0614	29.54
Avg over images	50	0.1025	24.66	0.1069	23.88
Avg over images	75	0.0599	29.79	0.0615	28.97
Avg over images	85	0.0432	32.78	0.0445	31.85
Overall avg	—	0.0685	29.07	0.0710	28.23

TABLE II: Table with results of runtime of is seconds and speed up for Matrix vs Tensor

Image	Keep [%]	Time Matrix [s]	Time Tensor [s]	Speed-up
image1	50	48.19	6.55	7.35
image1	75	44.17	7.25	6.09
image1	85	51.20	9.80	5.23
image2	50	61.88	6.73	9.20
image2	75	52.26	7.11	7.35
image2	85	47.28	6.32	7.48
image3	50	64.54	6.08	10.62
image3	75	54.76	10.10	5.42
image3	85	56.19	9.71	5.79
Avg over images	50	58.20	6.46	9.05
Avg over images	75	50.40	8.15	6.29
Avg over images	85	51.56	8.61	6.17
Overall avg	—	53.38	7.74	7.17

- [6] *The usc-sipi image database*, <http://sipi.usc.edu/database/>, 1977.