Tensor Cross Approximations for Image and Video Reconstruction

by Salman Ahmadi Asl

11.12.2023

Signal Processing
Volume 213, December 2023, 109121

Fast cross tensor approximation for image and video completion

Salman Ahmadi-Asl [a], Maame Gyamfua Asante-Mensah [a], Andrzej Cichocki [a b], Anh Huy Phan [a], Ivan Oseledets [a], Jun Wang [a]

Show more

Add to Mendeley   Share   Cite

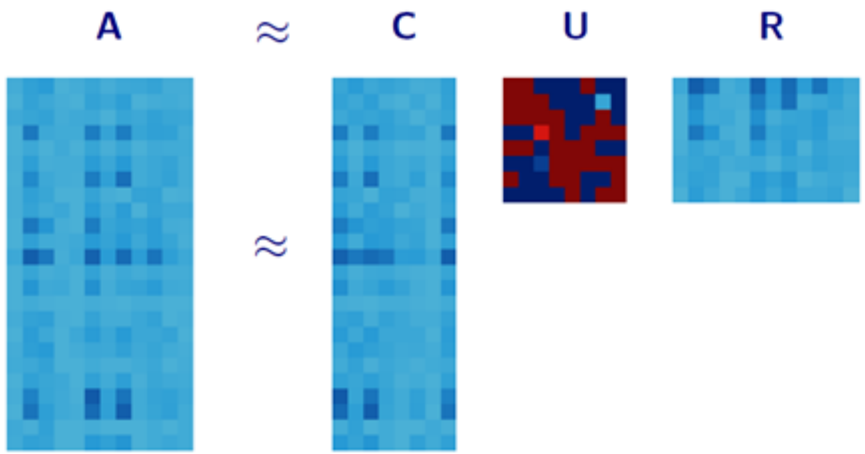
https://doi.org/10.1016/j.sigpro.2023.109121

Outline

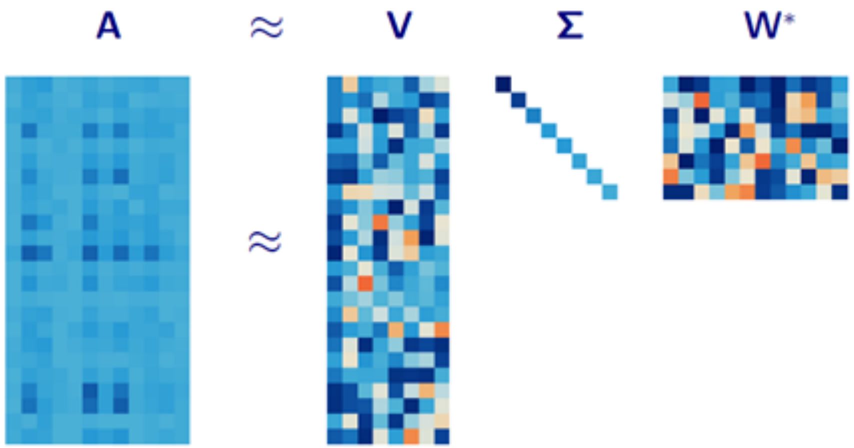
- **Cross matrix/tensor approximations**
- **Tensor completion using cross tensor approximations**
- **Simulations**

Cross Matrix Approximation

Cross (Skeleton) or equivalently CUR matrix approximation is used for computing a low-rank matrix approximation.



vs.



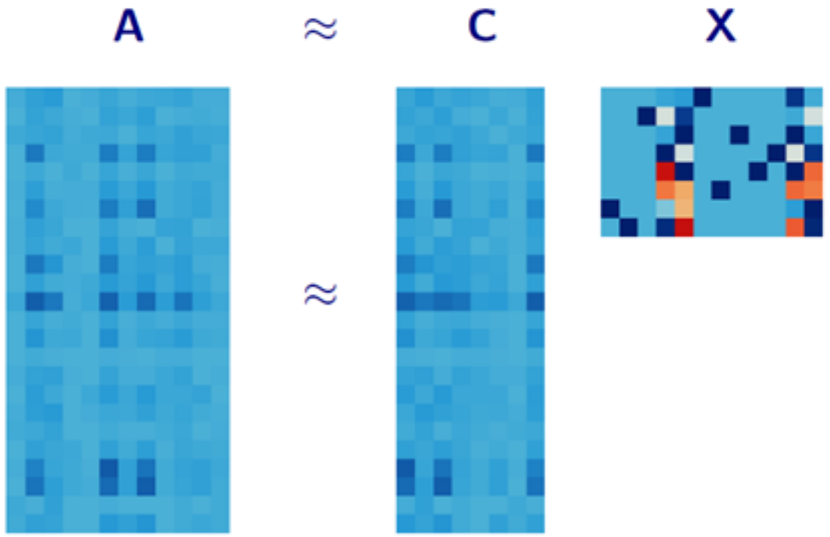
The main benefits of the CUR approximation are:

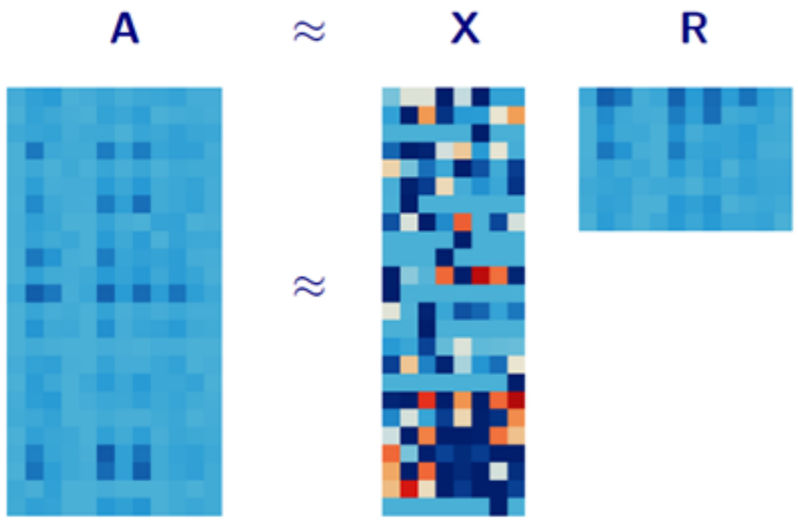
- Fast low-rank matrix computation
- Interpretability issues
- More compact matrix representation

Images were taken from https://personal.math.vt.edu/embree/cur_talk.pdf

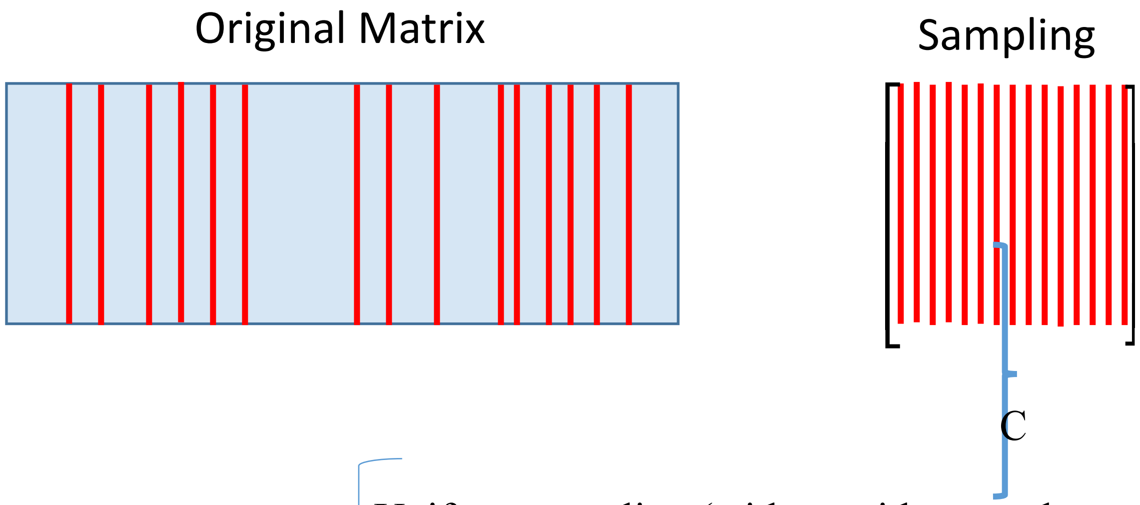
Cross Matrix Approximation

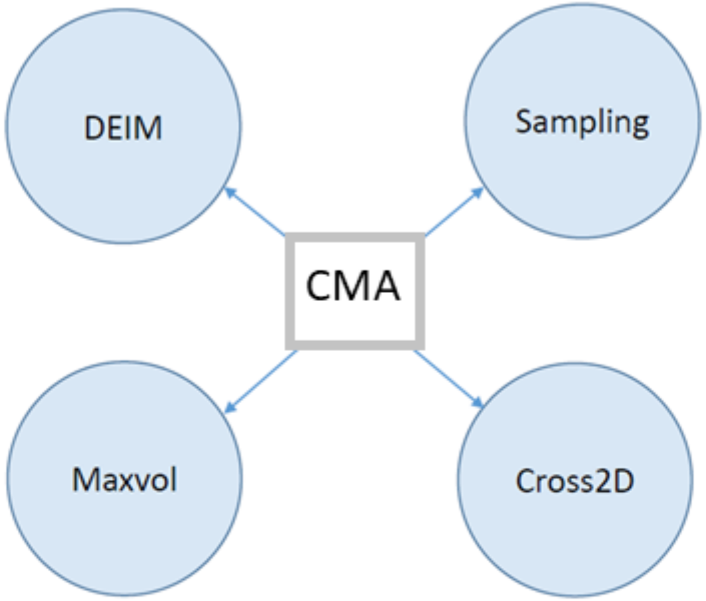
CX and XR are special cases of the CUR approximation in which we only select columns or rows, respectively, not both of them simultaneously.





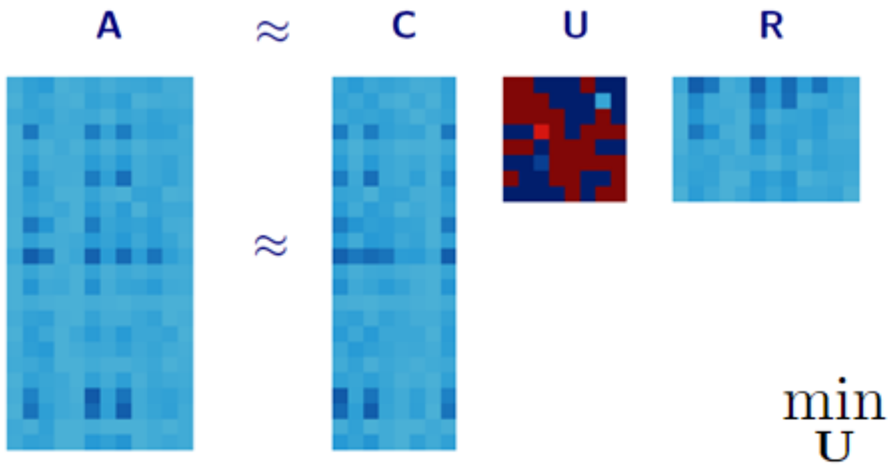
Sampling Based on Different Probability Distributions





Sampling columns
- Uniform sampling (with or without replacement)
- Length-squared distribution
- Leverage scores distribution

Selection of the Middle Matrix U in the CUR Approximation

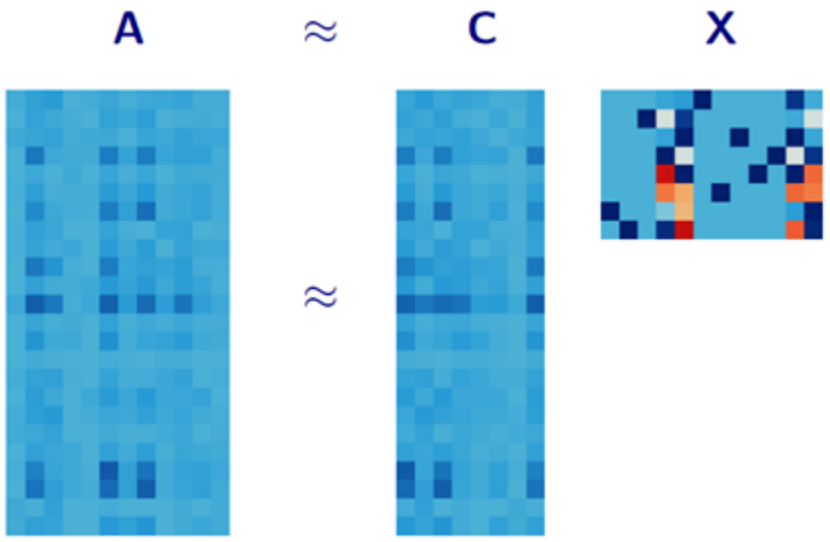


$$\min_{\mathbf{U}} \|\mathbf{A} - \mathbf{CUR}\|_F$$

$$\mathbf{U} = \mathbf{C}^{\dagger}\mathbf{A}\mathbf{R}^{\dagger}$$

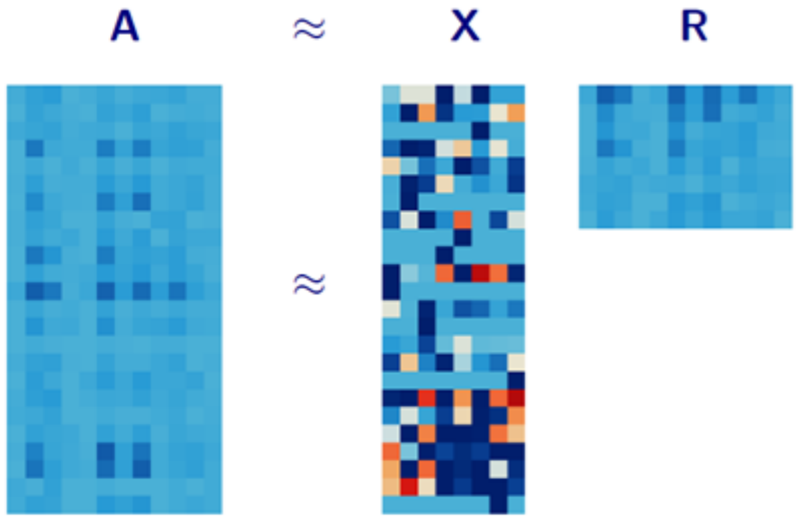
- One-Pass Algorithm

Selection of the Matrix X in the CX and XR Approximations



$$\min_{\mathbf{X}} \|\mathbf{A} - \mathbf{C}\mathbf{X}\|_F$$

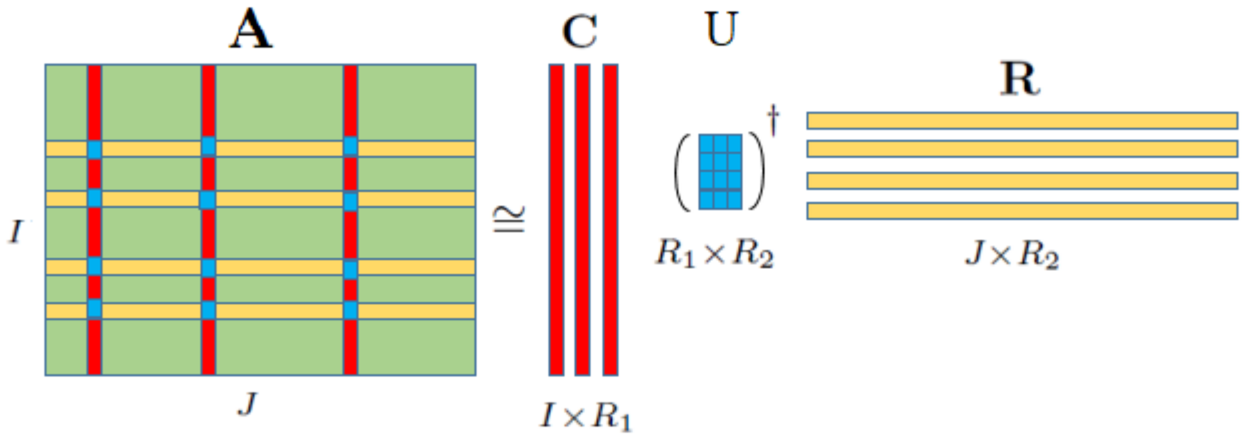
$$\mathbf{X} = \mathbf{C}^{\dagger}\mathbf{A}$$



$$\min_{\mathbf{X}} \|\mathbf{A} - \mathbf{X}\mathbf{R}\|_F$$

$$\mathbf{X} = \mathbf{A}\mathbf{R}^{\dagger}$$

Selection of the Middle Matrix U in the CUR Approximation



$$\mathbf{U} = \mathbf{W}^{\dagger} \quad \longrightarrow \quad \mathbf{A} = \mathbf{C}\mathbf{W}^{\dagger}\mathbf{R}$$

Rank($\mathbf{W}$)=Rank($\mathbf{A}$)

Sampling Based on Different Probability Distributions

- Uniform distribution

$$p_j = \frac{1}{J},\ j = 1, 2, \ldots, J$$

$$p_i = \frac{1}{I},\ \ i = 1, 2, \ldots, I$$

- Length-squared distribution

$$p_j = \frac{\|\mathbf{A}(:,j)\|_2^2}{\|\mathbf{A}\|_F^2},\ j = 1, 2, \ldots, J$$

$$p_i = \frac{\|\mathbf{A}(i,:)\|_2^2}{\|\mathbf{A}\|_F^2},\ \ i = 1, 2, \ldots, I$$

Sampling Based on Different Probability Distributions

- Leverage Scores are a popular technique for computing the CUR factorization, based on identifying the key elements of the singular vectors

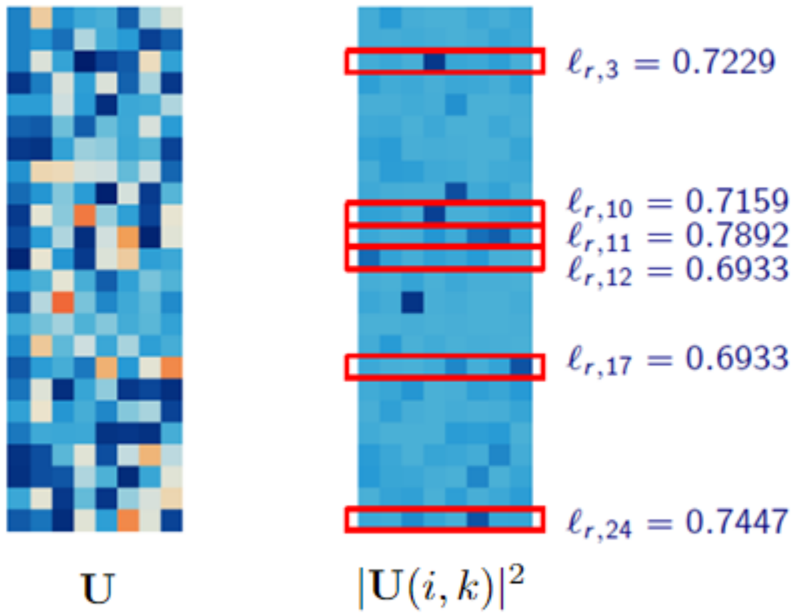
Suppose we have $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$, $\mathbf{U} \in \mathbb{R}^{I \times R}$ and $\mathbf{V} \in \mathbb{R}^{J \times R}$. To rank the importance of the rows, take the 2-norm of each row of $\mathbf{U}$:

Row leverage score= $l_{R,i} = \|\mathbf{U}(i,:)\|_2^2$

Column leverage score= $\hat{l}_{R,j} = \|\mathbf{V}(j,:)\|_2^2$

$$p_i = \frac{l_{R,i}}{R}, \; i = 1, 2, \ldots, I.$$

$$p_j = \frac{\hat{l}_{R,j}}{R}, \; j = 1, 2, \ldots, J.$$



Column/Row Coherence is the maximum of column/row leverage scores.

The Image was taken from https://personal.math.vt.edu/embree/cur_talk.pdf

Sampling Based on Different Probability Distributions

Depending on the probability distribution used for sampling columns/rows, either the additive or Multiplicative/relative error estimation is achieved:

$$\|\mathbf{A} - \mathbf{CUR}\|_\alpha^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_\alpha^2 + \epsilon\|\mathbf{A}\|_\alpha^2$$ Additive error norm

$$\|\mathbf{A} - \mathbf{CUR}\|_\alpha^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_\alpha^2$$ Multiplicative (Relative) error norm

$$\alpha = 2,\ F$$

Approximation with relative error norm is of more interest than the additive error norm.

Error Bound of the CUR Approximation

Mahoney and Drineas

CUR in time $\mathcal{O}(IJ)$ achieves

- $\|\mathbf{A} - \mathbf{CUR}\|_F \leq \|\mathbf{A} - \mathbf{A}_R\|_F + \epsilon\|\mathbf{A}\|_F$ by probability at least $1-\delta$, by picking
- $\mathcal{O}(R\log(1/\delta)/\epsilon^2)$ columns
- $\mathcal{O}(R^2\log^3(1/\delta)/\epsilon^6)$ rows

Maxvol approach

$$\|\mathbf{X} - \mathbf{CUR}\|_F \leq \sqrt{1 + R(J-R)}\sigma_{R+1}$$

$$\|\mathbf{X} - \mathbf{CUR}\|_\infty \leq (R+1)\sigma_{R+1}$$

Error Bound of the CUR Approximation

- *If* $p_j = \frac{\|A(:,j)\|_2^2}{\|A\|_F^2}$ *and sampling is performed with replacement*

$$\left\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^T\mathbf{A}\right\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_R\|_F^2 + \sqrt{\frac{4R}{\delta c}}\,\|\mathbf{A}\|_F^2 .$$

This sampling minimizes the variance for the error of the approximation.

- *If* $p_j = \frac{1}{J}$ *and sampling is performed with replacement,*

$$\left\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^T\mathbf{A}\right\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_R\|_F^2 + \sqrt{\frac{4R}{\delta}\frac{J}{c}\sum_{j=1}^{J}\|\mathbf{A}(:,j)\|_2^4}.$$

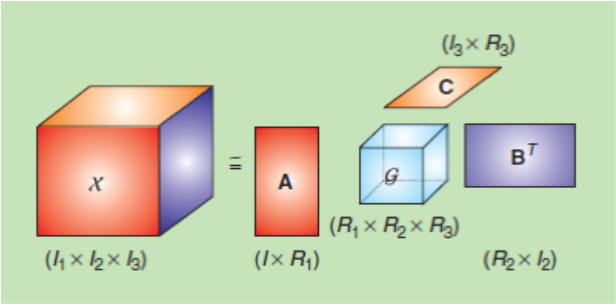
- *If* $p_j = \frac{1}{J}$ *and sampling is performed without replacement,*

$$\left\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^T\mathbf{A}^T\right\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_R\|_F^2 + \sqrt{\frac{4R}{\delta}\left(\frac{J}{c} - 1\right)\sum_{j=1}^{J}\|\mathbf{A}(:,j)\|_2^4}.$$

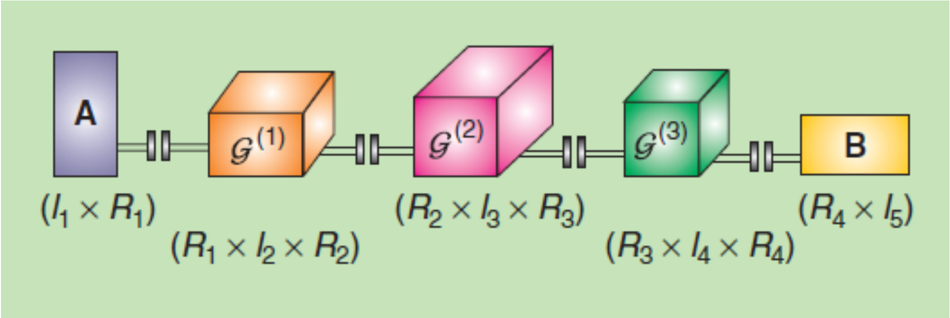
$\mathbf{A}_R$ *is the best rank-*R *approximation of matrix* $\mathbf{A}$ *in least-squares sense, for any* $c \leq J$*, with probability at least* $1 - \delta$

Different tensor decompositions are important data analysis tools

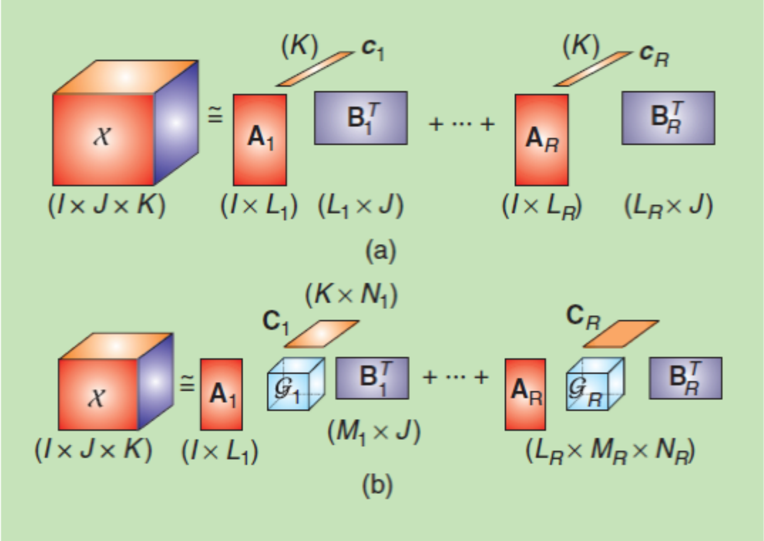
Tucker decomposition



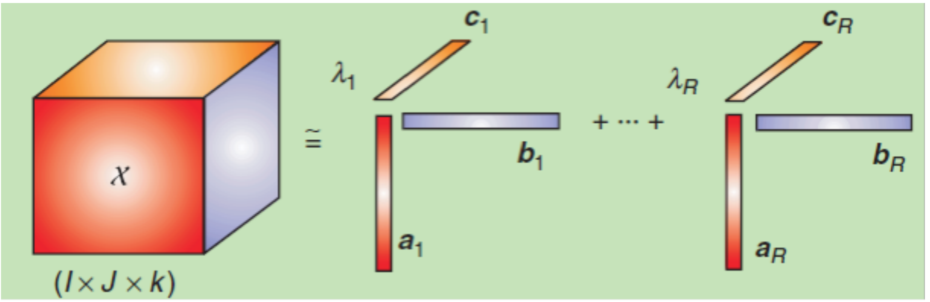
Tensor Train Decomposition



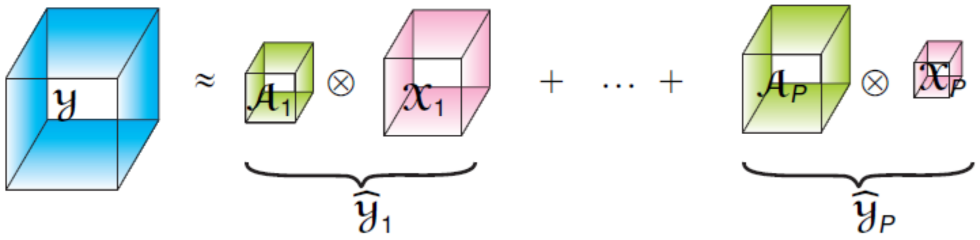
Block Tensor Decomposition



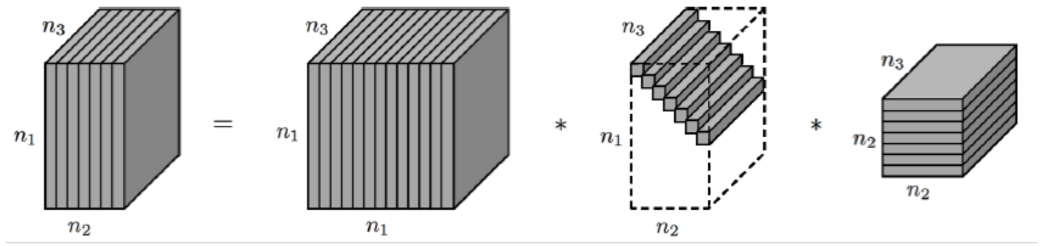
CP decomposition



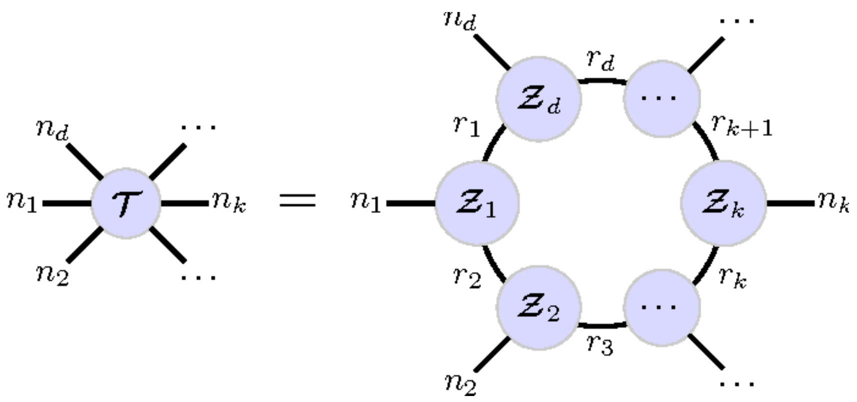
Different tensor decompositions



Kronecker Tensor Decomposition (KTD)



Tubal Tensor Decomposition

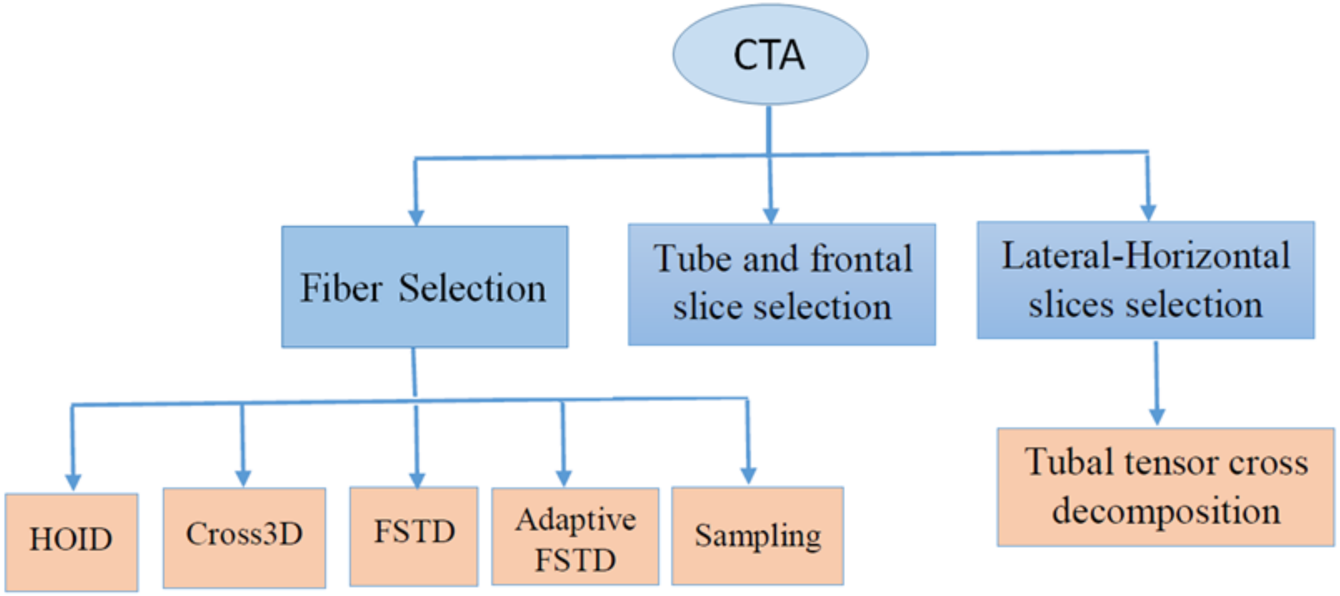


Tensor Ring Decomposition

Tensor CUR Approximation: A Generalization of the Matrix CUR

Motivated by the application of the matrix CUR, it has been generalized to tensors in the following ways:

- Fiber selection
- Slice/Fiber Selection
- Slice/Slice Selection



Ahmadi-Asl, et. al, Cross Tensor Approximation Methods for Compression and Dimensionality Reduction, IEEE ACCESS, 2021.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 939–956

TUCKER DIMENSIONALITY REDUCTION OF THREE-DIMENSIONAL ARRAYS IN LINEAR TIME*

I. V. OSELEDETS†, D. V. SAVOSTIANOV†, AND E. E. TYRTYSHNIKOV†

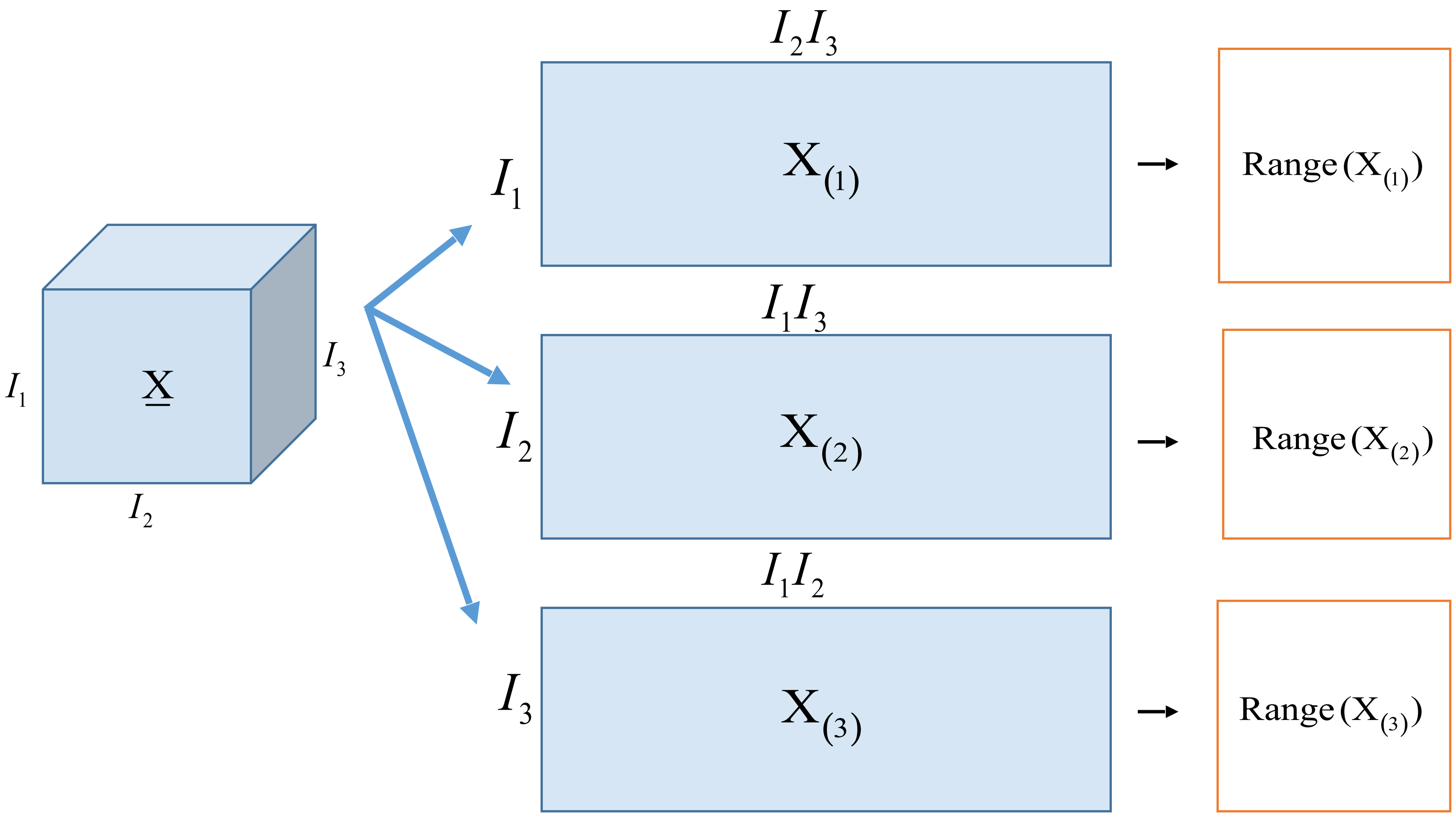
Abstract. We consider Tucker-like approximations with an $r \times r \times r$ core tensor for three-dimensional $n \times n \times n$ arrays in the case of $r \ll n$ and possibly very large n (up to 10^4–10^6). As the approximation contains only $\mathcal{O}(rn + r^3)$ parameters, it is natural to ask if it can be computed using only a small amount of entries of the given array. A similar question for matrices (two-dimensional tensors) was asked and positively answered in [S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, *A theory of pseudo-skeleton approximations*, Linear Algebra Appl., 261 (1997), pp. 1–21]. In the present paper we extend the positive answer to the case of three-dimensional tensors. More specifically, it is shown that if the tensor admits a good Tucker approximation for some (small) rank r, then this approximation can be computed using only $\mathcal{O}(nr)$ entries with $\mathcal{O}(nr^3)$ complexity.

Key words. multidimensional arrays, Tucker decomposition, tensor approximations, low-rank approximations, skeleton decompositions, dimensionality reduction, data compression, large-scale matrices, data-sparse methods

AMS subject classifications. 15A69, 15A18, 65F30

DOI. 10.1137/060655894

Tensor CUR Approximation Based on Fiber Selection



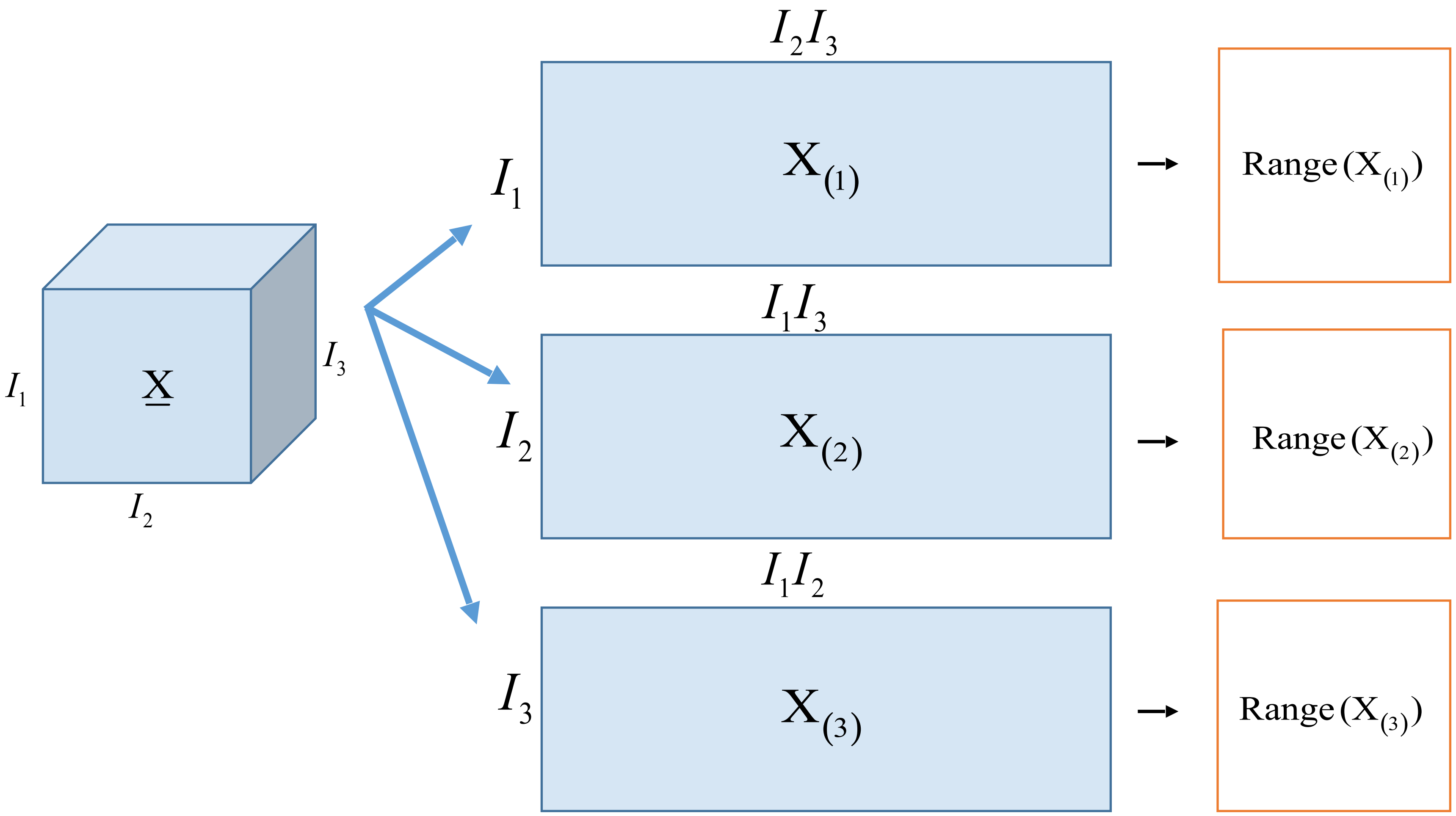
How to Find a Good Submatrix*

S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, and N. L. Zamarashkin

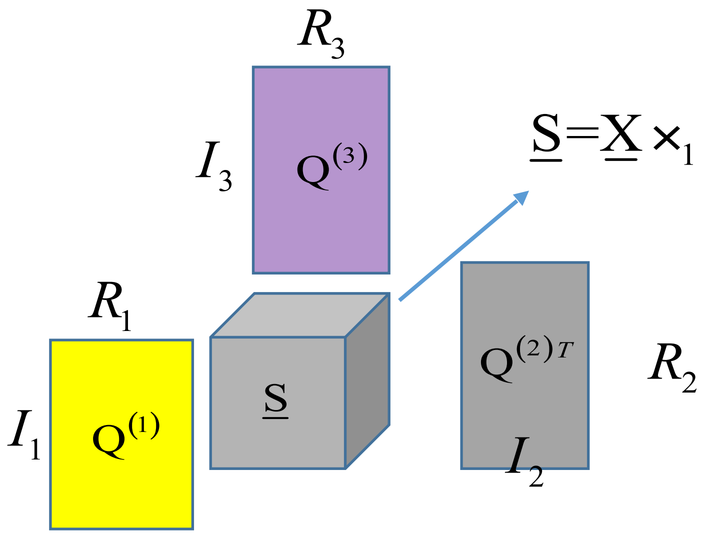
Institute of Numerical Mathematics of Russian Academy of Sciences,
Gubkina 8, 119333 Moscow, Russia
{sergei,ivan,draug,tee,kolya}@bach.inm.ras.ru

Tensor CUR Approximation Based on Fiber Selection

Drineas & Mahoney , A randomized algorithm for a tensor based generalization of the singular value decomposition, *Linear Algebra and its Applications*, 2006.



Tensor CUR Approximation Based on Fiber Selection



$$\underline{S} = \underline{X} \times_1 Q^{(1)T} \times_2 Q^{(2)T} \times_3 Q^{(3)T}$$

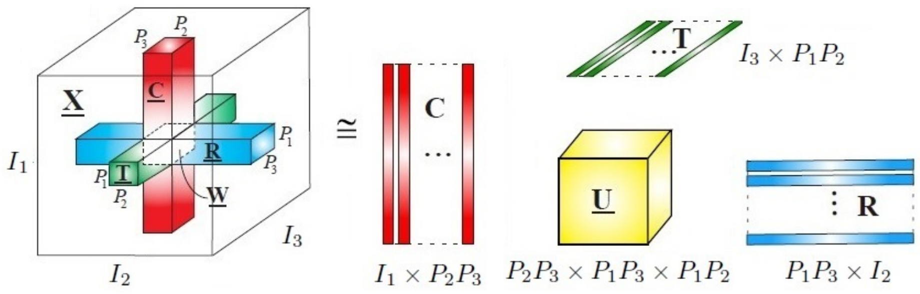
Tensor CUR Approximation Based on Fiber Selection

Matrix case $\quad \mathbf{U} = \mathbf{W} \times_1 \mathbf{W}^+_{(1)} \times_2 \mathbf{W}^+_{(2)} = \mathbf{W}^+ \mathbf{W} \mathbf{W}^+ = \mathbf{W}^+$

Tensor case

$$\begin{aligned} \underline{\mathbf{U}} &= \underline{\mathbf{W}} \times_1 \mathbf{W}^+_{(1)} \times_2 \mathbf{W}^+_{(2)} \times_3 \mathbf{W}^+_{(3)} \\ &\equiv \left[\!\left[\underline{\mathbf{W}}; \mathbf{W}^+_{(1)}, \mathbf{W}^+_{(2)}, \mathbf{W}^+_{(3)} \right]\!\right] \end{aligned}$$

$$\begin{aligned} \underline{\mathbf{X}} &\cong [\![\underline{\mathbf{U}}; \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3]\!] \\ &\equiv \left[\!\left[\underline{\mathbf{W}}; \underbrace{\mathbf{A}_1 \mathbf{W}^+_{(1)}}_{\widetilde{\mathbf{C}}_1}, \underbrace{\mathbf{A}_2 \mathbf{W}^+_{(2)}}_{\widetilde{\mathbf{C}}_2}, \underbrace{\mathbf{A}_3 \mathbf{W}^+_{(3)}}_{\widetilde{\mathbf{C}}_3} \right]\!\right] \end{aligned}$$

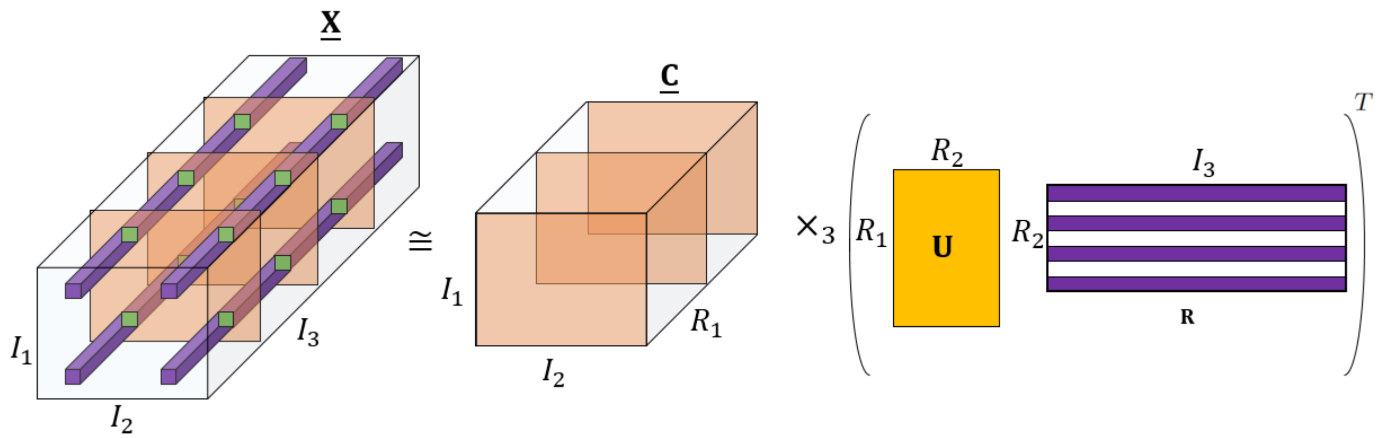


C. F. Caiafa, A. Cichocki, Generalizing the column–row matrix decomposition to multi-way arrays, Linear Algebra and its Applications, 2010.

FSTD

Tensor CUR Approximation Based on Slice and Tube Selection

The Tensor CUR approximation can be model as follows



Michel Mahoney et.al, Tensor-CUR decompositions for tensor-based data, SIAM Journal on Matrix Analysis and Applications, 2008.

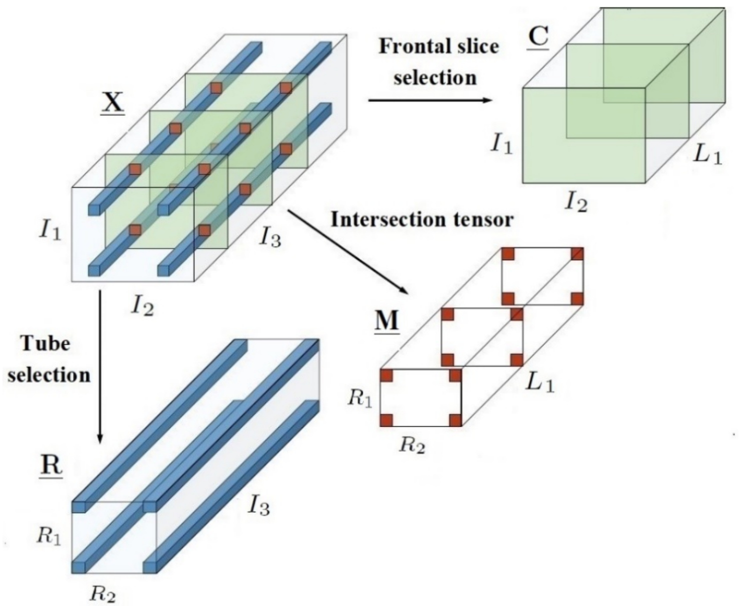
$$p_i = \frac{\|\underline{\mathbf{X}}(:,:,i_3)\|_F^2}{\|\underline{\mathbf{X}}\|_F^2}, \; i_3 = 1, 2, \ldots, I_3,$$

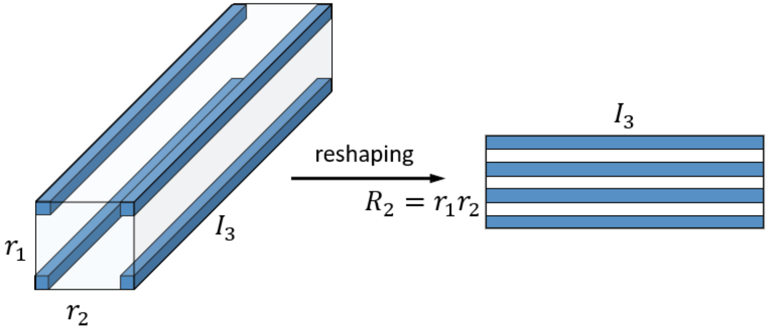
$$q_j = \frac{\underline{\mathbf{X}}(j_1, j_2, :)}{\|\underline{\mathbf{X}}\|_F^2}, \quad j_1, j_2 \in J_1, J_2$$

Tensor CUR Approximation Based on Slice and Tube Selection

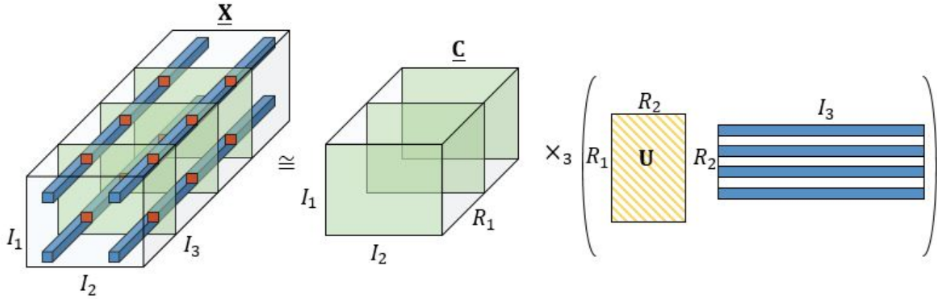
The procedure of frontal slice and tube selections is as follows:

Step 1



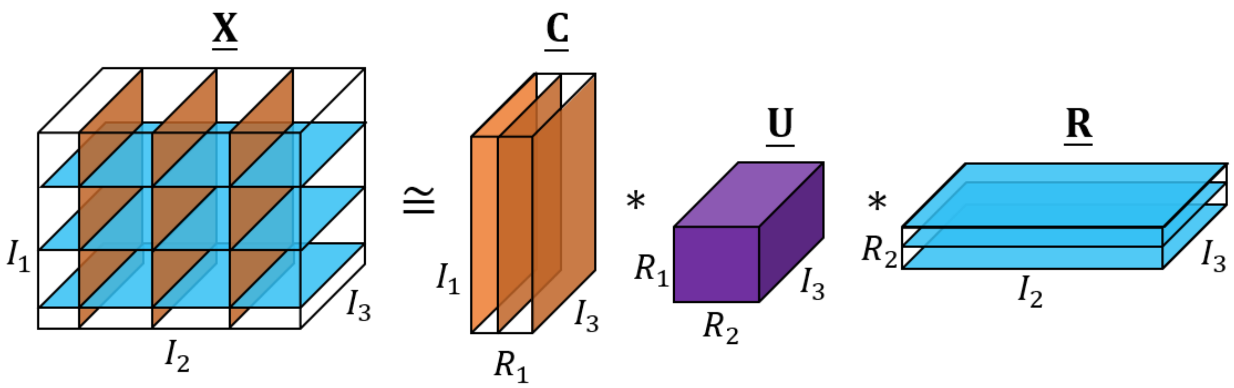


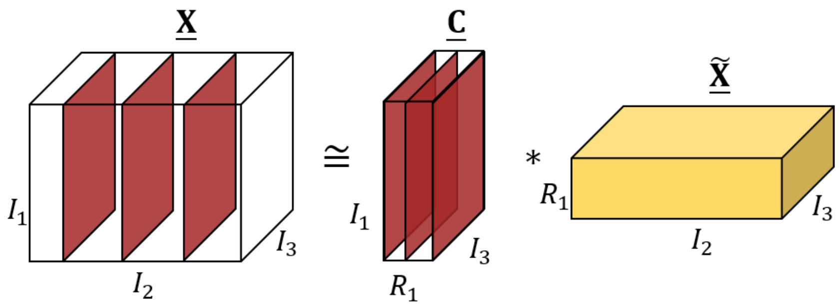
Step 2



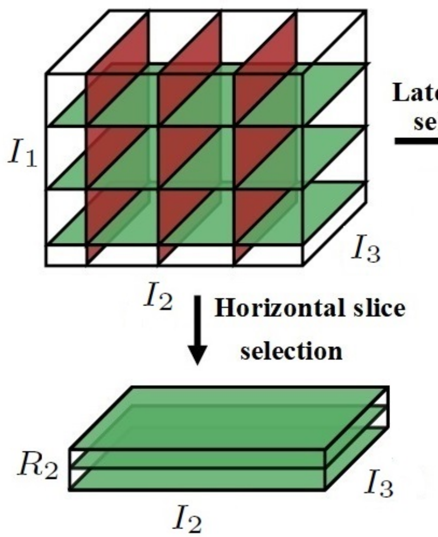
Tensor CUR Approximation Based on Tubal Approximation

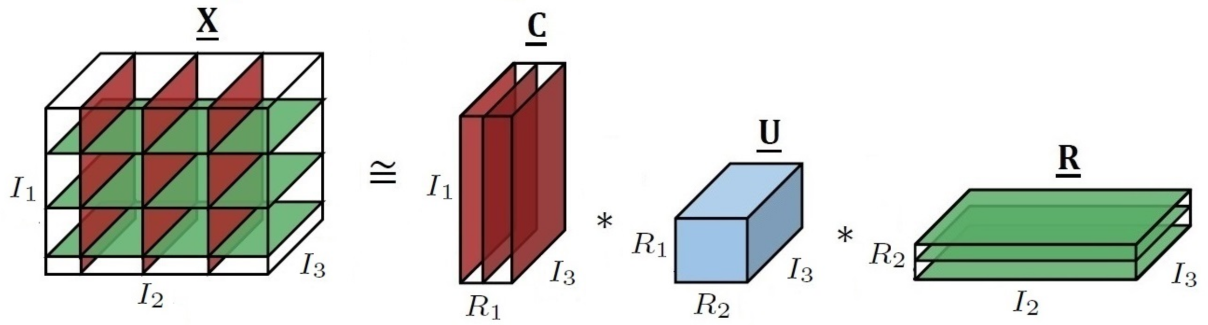
The CUR approximation can be generalized to tensors using the tubal product





Tensor CUR Approximation Based on Tubal Product

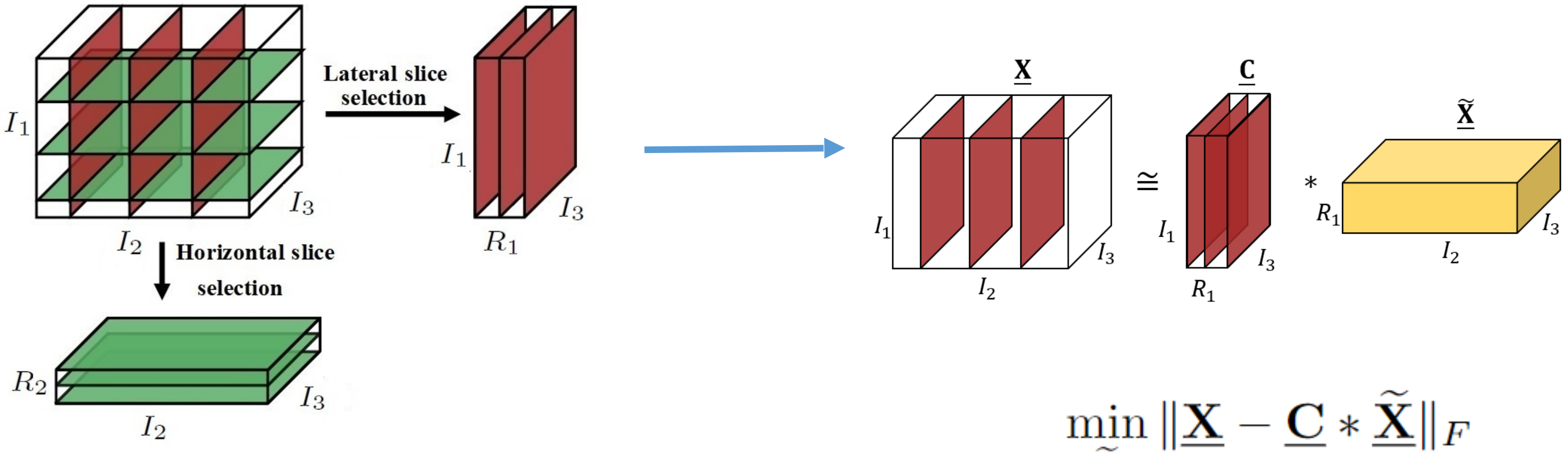




$$\underline{\mathbf{U}} = \underline{\mathbf{C}}^{\dagger} * \underline{\mathbf{X}} * \underline{\mathbf{R}}^{\dagger}$$

$$\underline{\mathbf{U}} = \underline{\mathbf{W}}^{\dagger}$$

Tensor CUR Approximation Based on Tubal Product

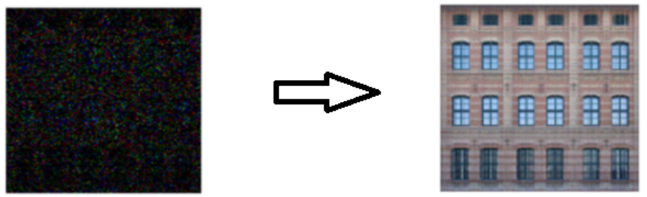


$$\min_{\underline{\widetilde{\mathbf{X}}}} \|\underline{\mathbf{X}} - \underline{\mathbf{C}} * \underline{\widetilde{\mathbf{X}}}\|_F$$

$$\underline{\widetilde{\mathbf{X}}} = \underline{\mathbf{C}}^{\dagger} * \underline{\mathbf{X}}$$

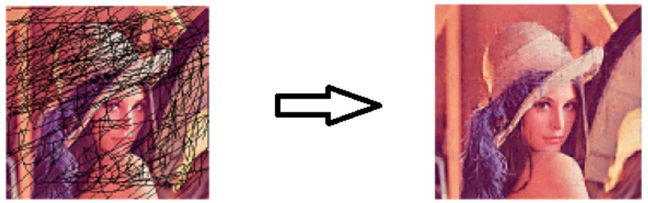
Tensor Completion

- Tensor completion problem is an extension of matrix completion problem
- The problem is to estimate the missing or uncertain elements of a tensor
  - (RGB) Image inpainting
  - Video completion
  - Time series completion
  - Image supperresolution
  - …
  - In Tensor completion problem
    - Some of the elements are not available
    - Some of the elements are noisy or destroyed by outliers



PSNR= 6.2029
SSIM= 0.0258

PSNR= 25.5538
SSIM= 0.8072
Time= 1.5988 Sec



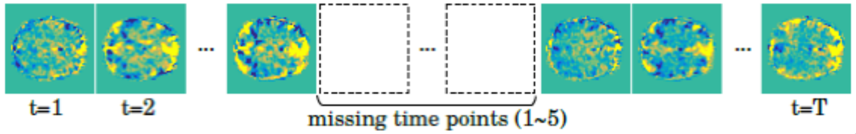
PSNR=10.4236
SSIM=0.2285

PSNR= 28.2562
SSIM= 0.8096
Time= 2.317 Sec

Tensor Completion

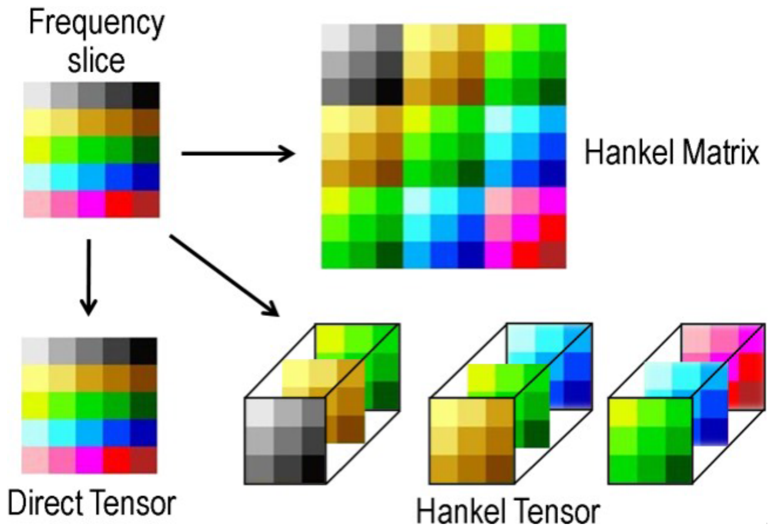
Two main challenges in the tensor completion problem

- Reconstructing images/videos with structural missing components or high missing ratio

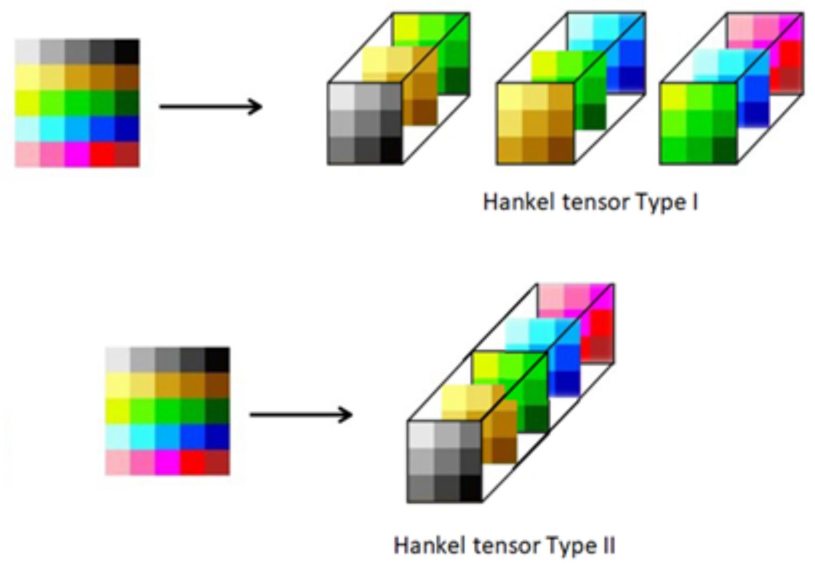


- High computational complexity of the algorithms

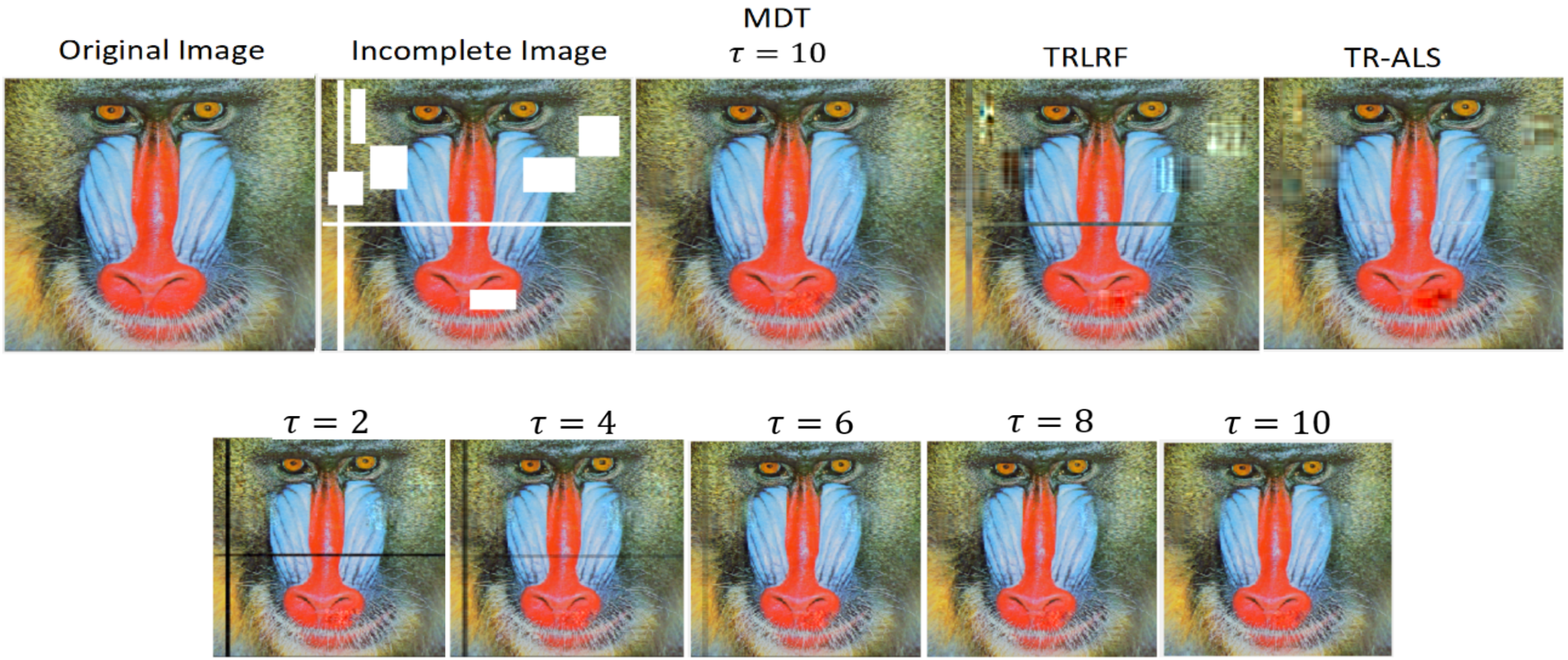
Hankelization procedure



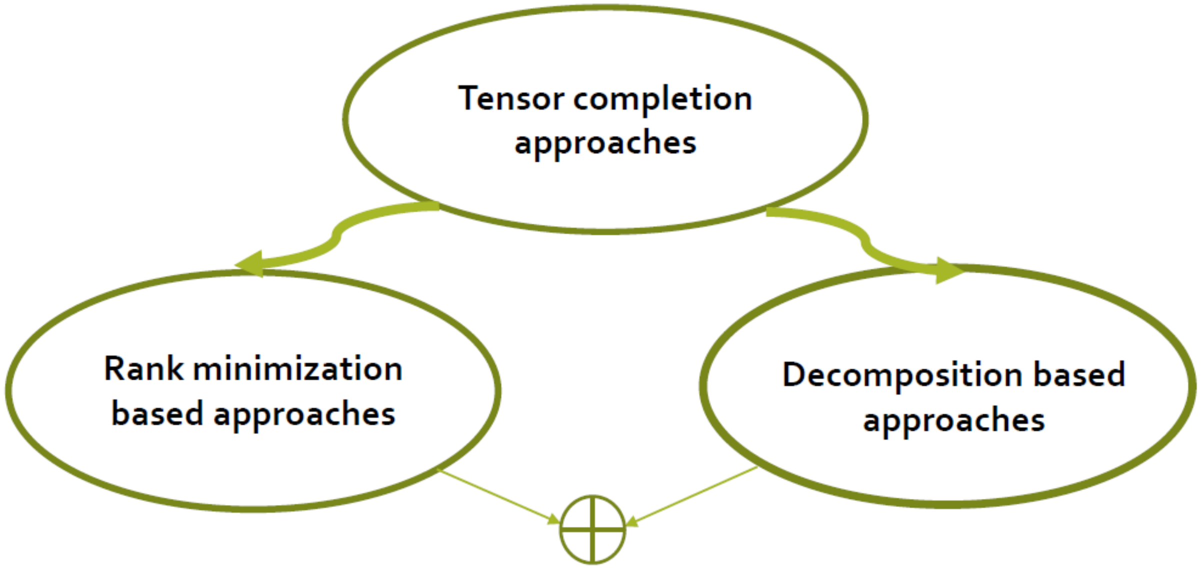
Different Hankelization Strategies



Hankelization Results



Two Main Categorizes for Tensor Completion



Tensor Completion Problem Based on the Cross Tensor Approximation

The rank minimization can efficiently recover data from only a part of its components

$$\begin{aligned} \min_{\underline{\mathbf{X}}} \quad & \text{Rank}\left(\underline{\mathbf{X}}\right) \\ \text{s.t.} \quad & \mathbf{P}_{\underline{\Omega}}(\underline{\mathbf{X}}) = \mathbf{P}_{\underline{\Omega}}(\underline{\mathbf{M}}), \end{aligned}$$

The notion of rank is not unique for tensors.

$\Big\downarrow$ Tucker rank

$$\begin{aligned} \min_{\underline{\mathbf{X}}} \quad & \textstyle\sum_{n=1}^{N} R_n \\ \text{s.t.} \quad & \mathbf{P}_{\underline{\Omega}}(\underline{\mathbf{X}}) = \mathbf{P}_{\underline{\Omega}}(\underline{\mathbf{M}}), \end{aligned}$$

Tensor Completion Problem Based on the Cross Tensor Approximation

We use tensor factorization approach in which for a fixed rank, the following optimization problem should be solved:

$$\begin{aligned} \min_{\underline{\mathbf{X}},\underline{\mathbf{C}}} \quad & \|\underline{\mathbf{X}} - \underline{\mathbf{C}}\|_F^2, \\ \text{s.t.} \quad & \mathrm{Rank}(\underline{\mathbf{X}}) = R, \\ & \mathbf{P}_{\underline{\Omega}}(\underline{\mathbf{C}}) = \mathbf{P}_{\underline{\Omega}}(\underline{\mathbf{M}}) \end{aligned}$$

ALS (Alternating Least Squares) Iterations

$$\underline{\mathbf{X}}^{(n)} \leftarrow \mathcal{L}(\underline{\mathbf{C}}^{(n)}),$$ Low-rank Approximation

$$\underline{\mathbf{C}}^{(n+1)} \leftarrow \underline{\mathbf{\Omega}} \circledast \underline{\mathbf{M}} + (\underline{\mathbf{1}} - \underline{\mathbf{\Omega}}) \circledast \underline{\mathbf{X}}^{(n)}$$ Masking procedure

The Completion Algorithm Based on the Tensor CUR

Algorithm 1: Tensor CUR algorithm for Nth-order tensor completion.

Input : An incomplete data tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, Tensor Rank $\mathbf{R}$, the set of observed components $\mathbf{\Omega}$, error bound ε and $\mathrm{MaxIter}$.

Output : Completed data tensor $\underline{\mathbf{X}}^*$

$\underline{\mathbf{X}}^{(0)} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is the original data tensor with missing pixels;

$\underline{\mathbf{Y}}^{(0)} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is a zero tensor;

for $n = 0, 1, 2, \ldots$ **do**

  $\underline{\mathbf{Y}}^{(n+1)} \leftarrow$Compute CUR approximation of the data tensor $\underline{\mathbf{X}}^{(n)}$ using selected fibers/slices and smoothing them,

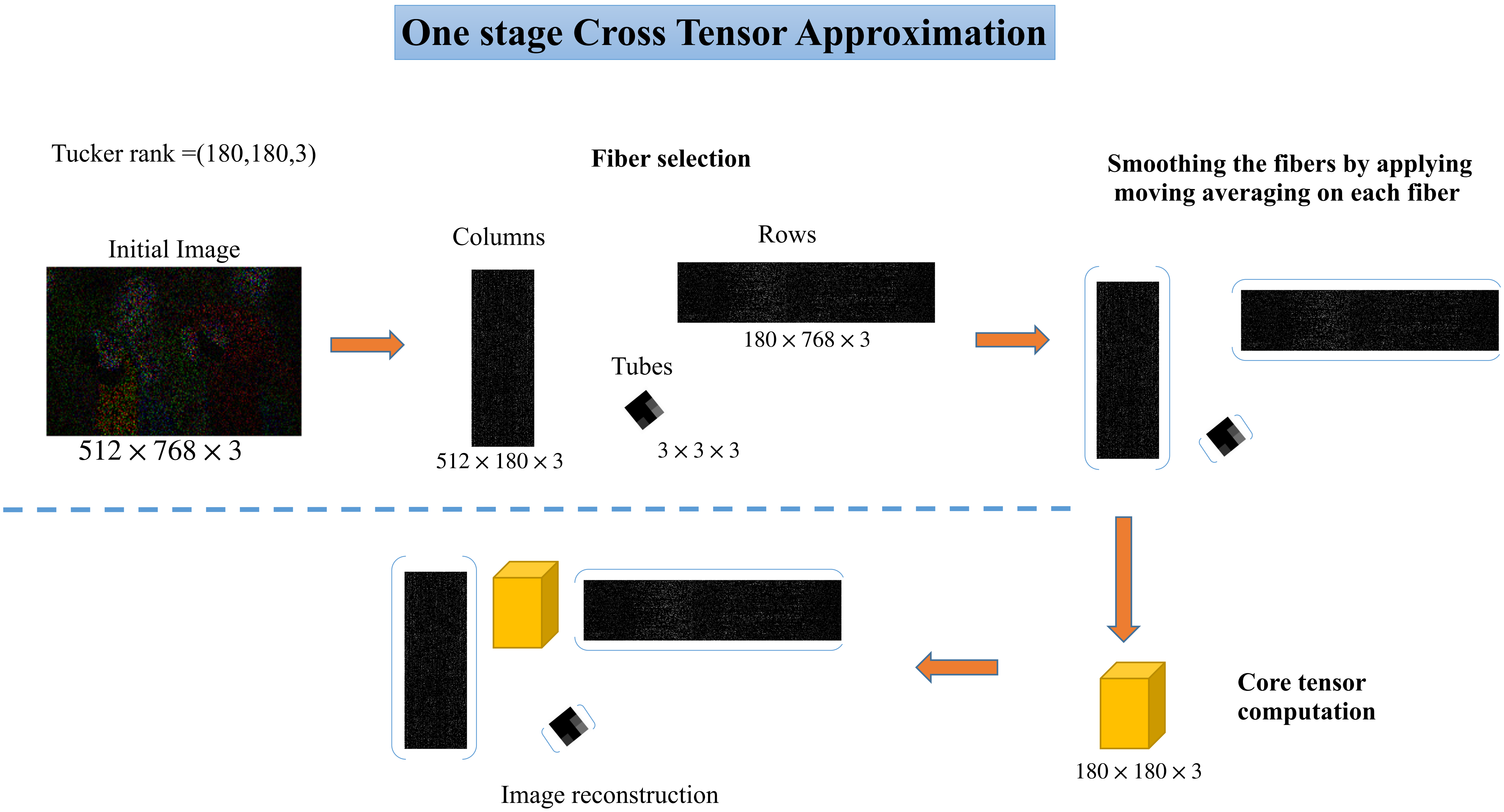
  $\underline{\mathbf{X}}^{(n+1)} \leftarrow \mathbf{P}_{\underline{\mathbf{\Omega}}}(\underline{\mathbf{X}}^{(n)}) + \mathbf{P}_{\underline{\mathbf{\Omega}}^{\perp}}(\underline{\mathbf{Y}}^{(n+1)})$,

  if $\frac{\left\|\underline{\mathbf{X}}^{(n+1)} - \underline{\mathbf{X}}^{(n)}\right\|_F}{\left\|\underline{\mathbf{X}}^{(n+1)}\right\|_F} < \varepsilon \textit{ or } n > \mathrm{MaxIter}$ **then**

    $\underline{\mathbf{X}}^* = \underline{\mathbf{X}}^{(n+1)}$ and **break**,

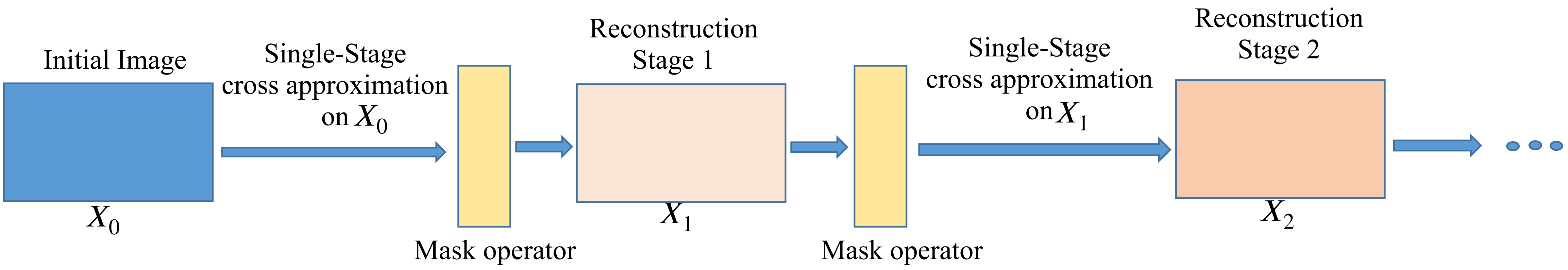
  end

end


One stage Cross Tensor Approximation
Tucker rank =(180,180,3)
Fiber selection
Smoothing the fibers by applying moving averaging on each fiber
Initial Image
512 × 768 × 3
Columns
512 × 180 × 3
Rows
180 × 768 × 3
Tubes
3 × 3 × 3
Core tensor computation
180 × 180 × 3
Image reconstruction

Multi-Stage Cross Approximation for Image/Video Completion

The one stage cross approximation does not work well in practice and we need multi-stage cross approximation by concatenating several single-stage cross approximation.

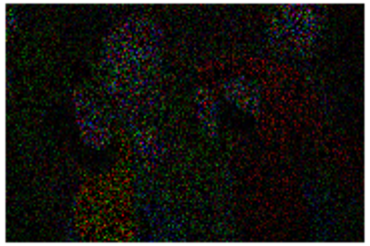


$$\ldots \color{blue}{Mask}(\color{red}{cross}(\color{blue}{Mask}(\color{red}{cross}(X_0)))) \ldots .$$

Linear operator   Linear operator

Multi-Stage Cross Approximation for Image/Video Completion

80% missing



PSNR= 7.1977

Iteration 10

PSNR = 12.8479

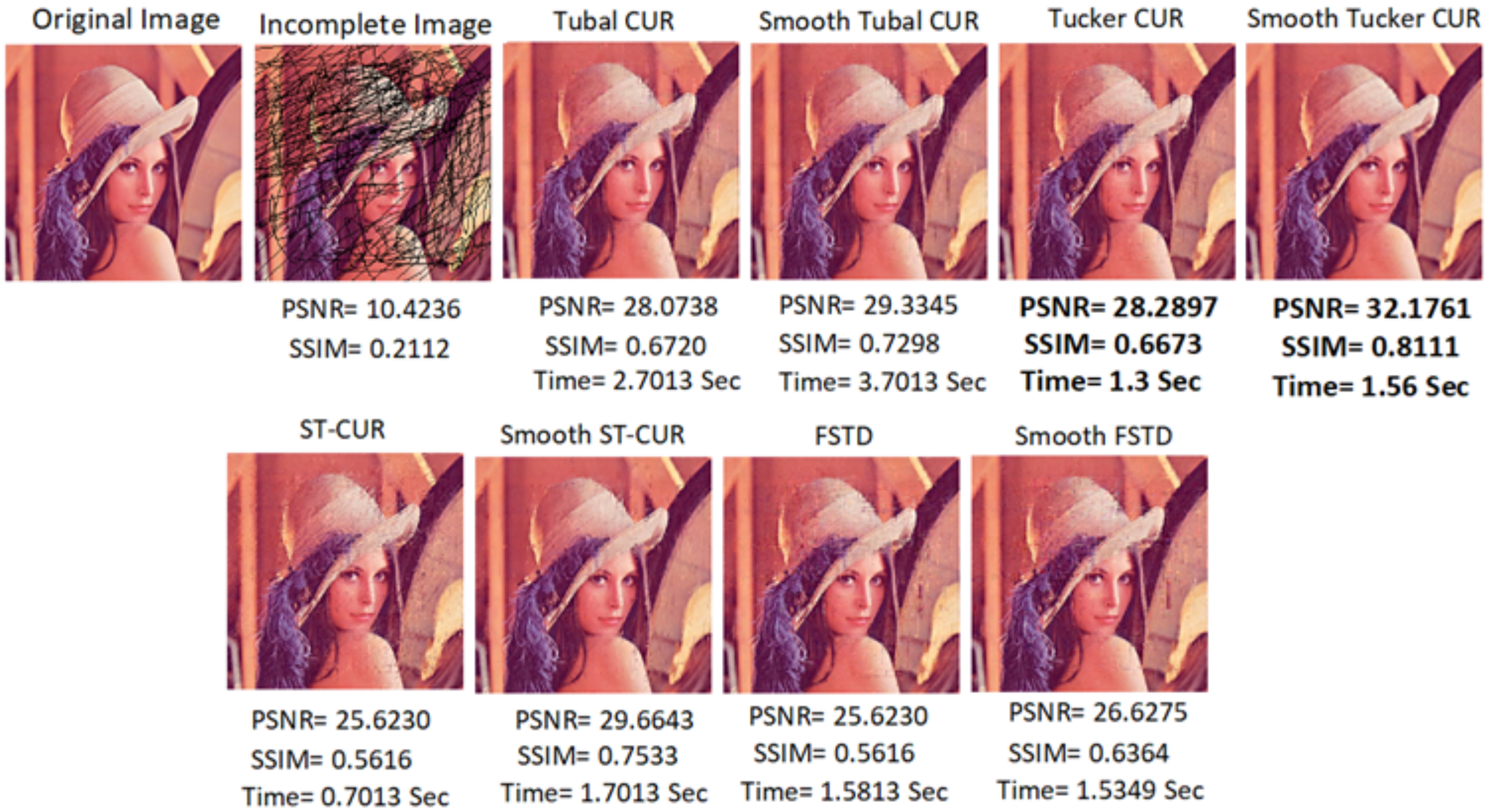
Iteration 50

PSNR = 24.3966

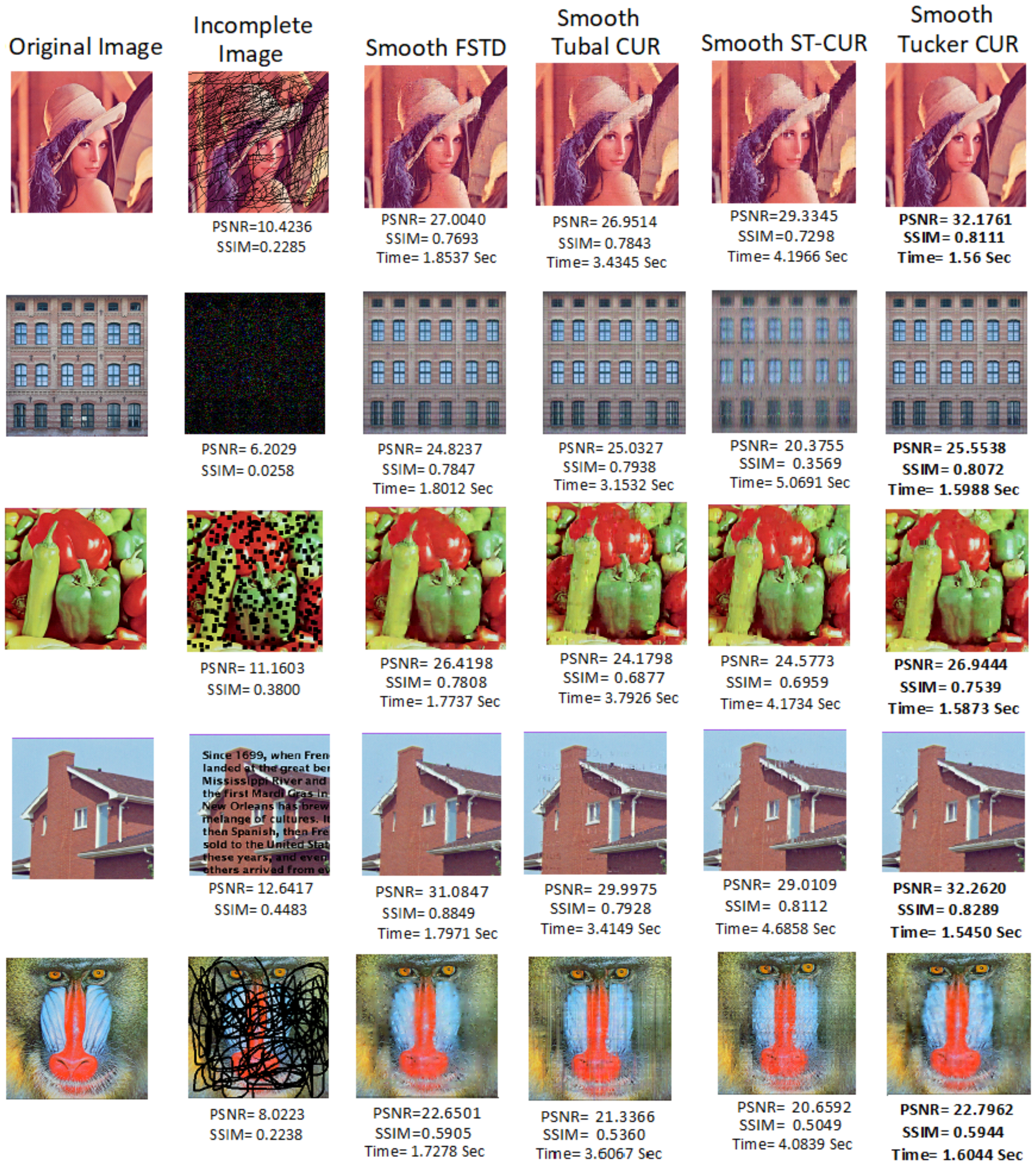
Iteration 100

PSNR = 27.4593

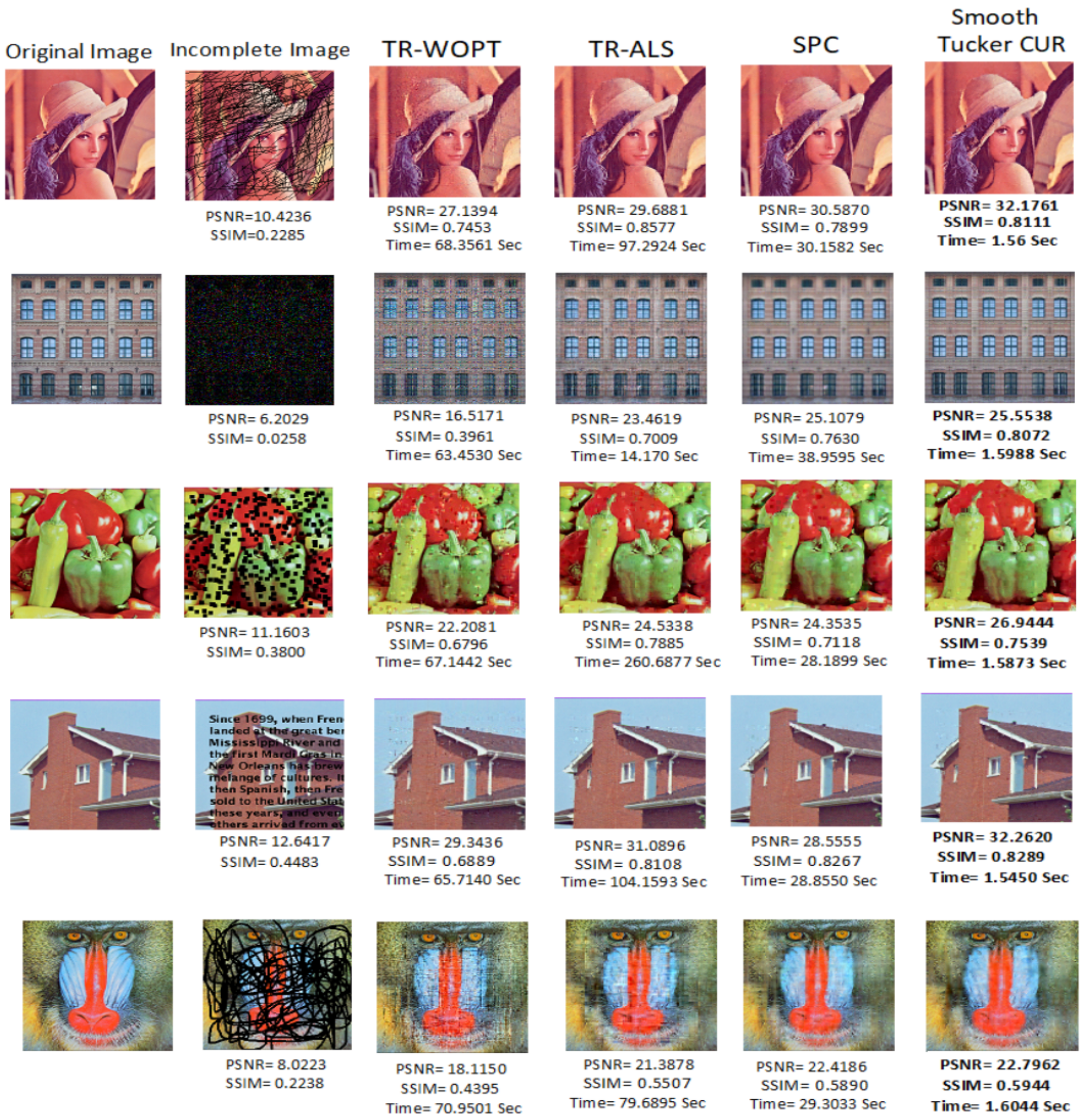
Computer Simulations: Images



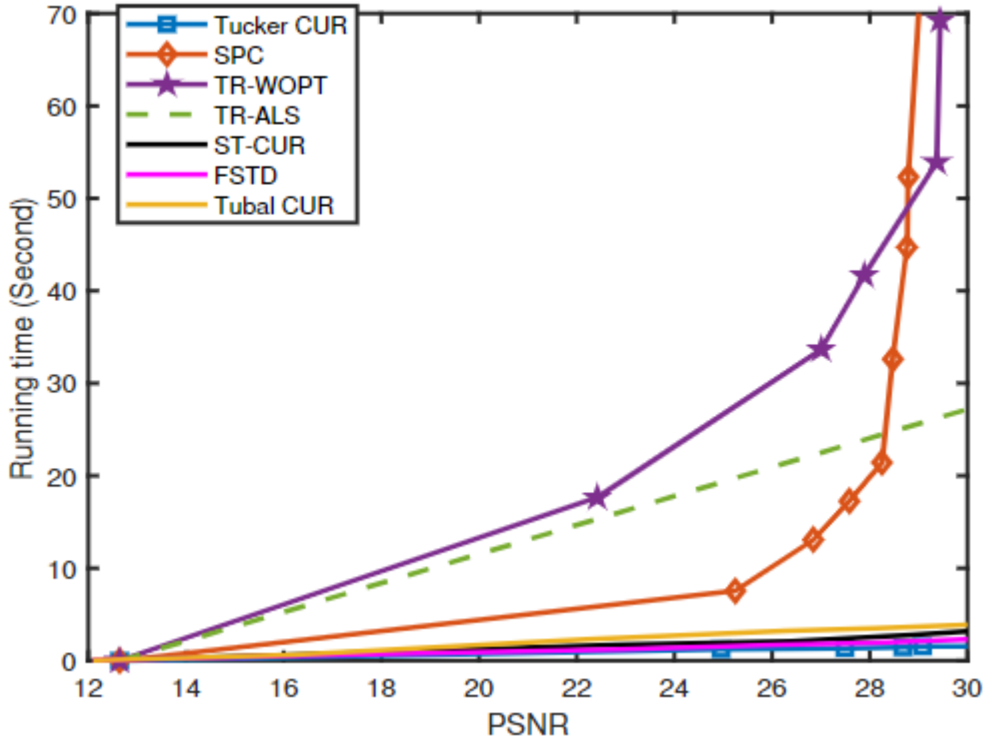
Computer Simulations: Images



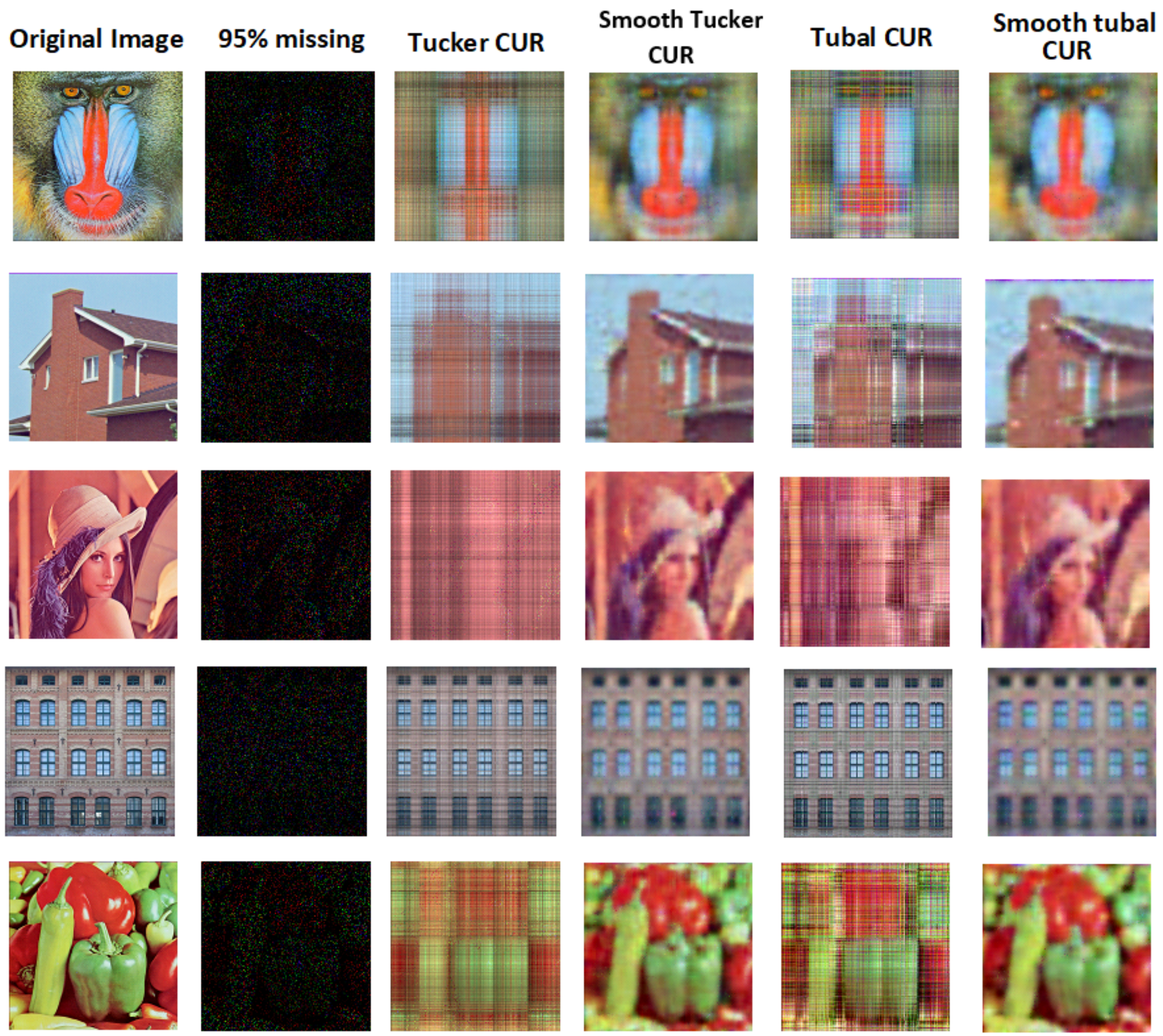
Computer Simulations: Images



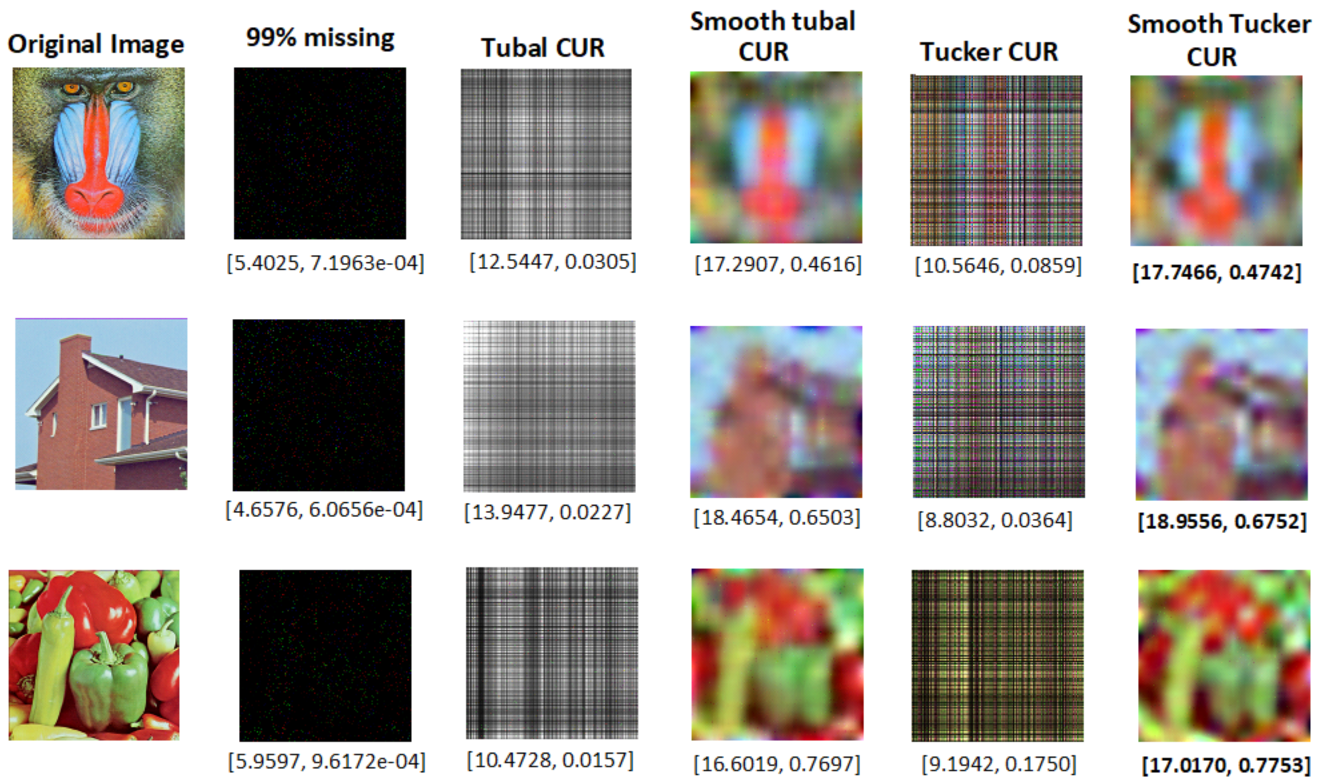
Computer Simulations: Images



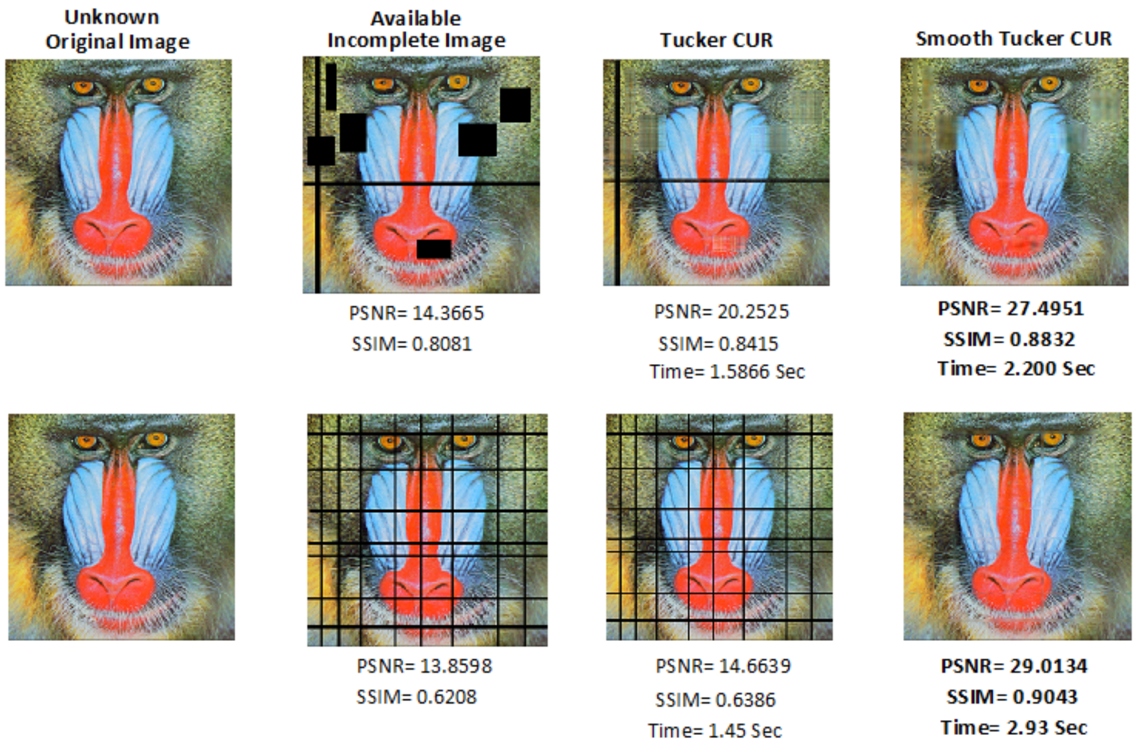
Computer Simulations: Images



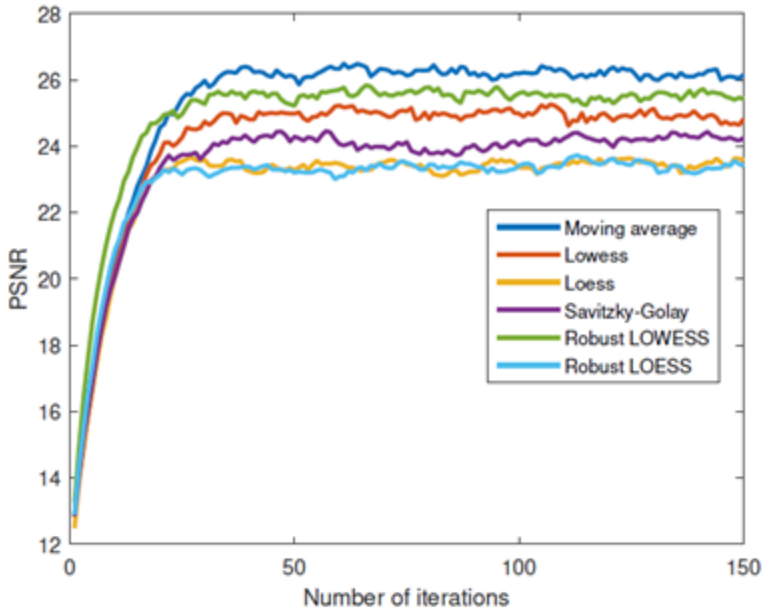
Computer Simulations: Images

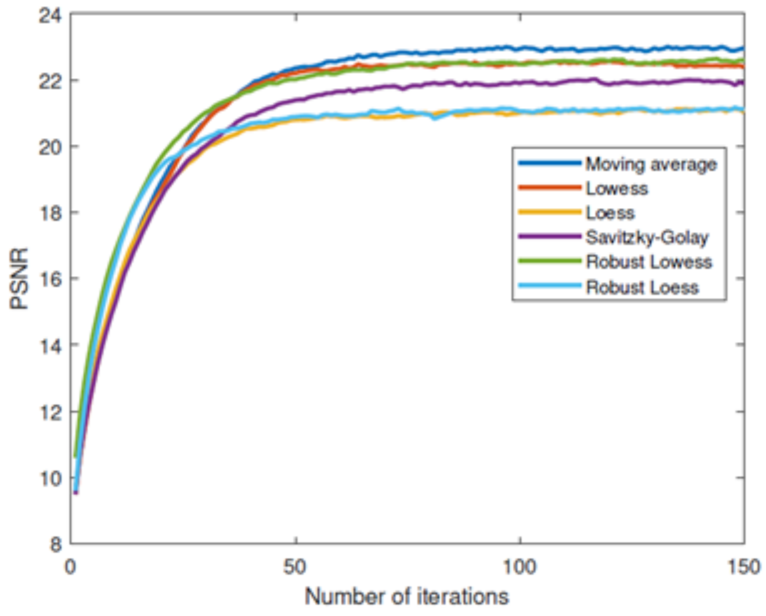


Computer Simulations: Images



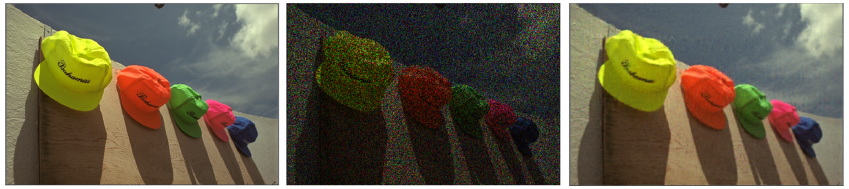
Computer Simulations: Images





Computer Simulations: Kodak Dataset

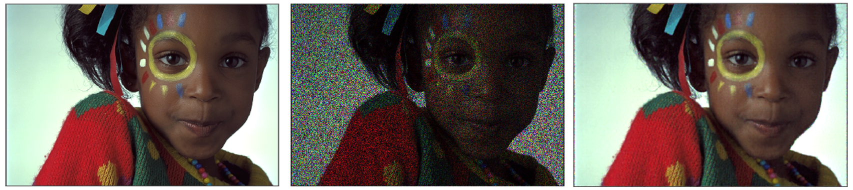
Original image | 70% missing | Reconstruction



PSNR= 30.1033

PSNR= 29.7648

PSNR= 30.4661



PSNR= 29.2252

Original image | 70% missing | Reconstruction

PSNR= 29.2632

PSNR= 29.8007

Computer Simulations: Kodak Dataset

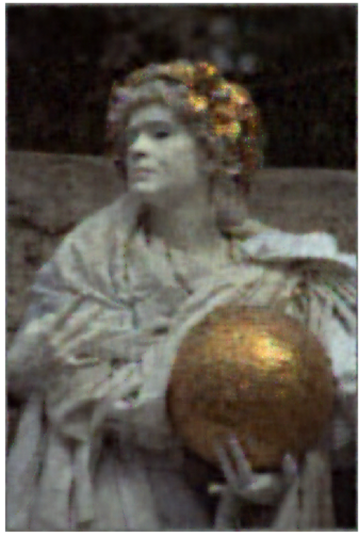

Original Image


PSNR= 25.3695


Original Image


PSNR= 26.8710


Original Image


PSNR= 25.5979


Original Image


PSNR= 26.6118


Original Image

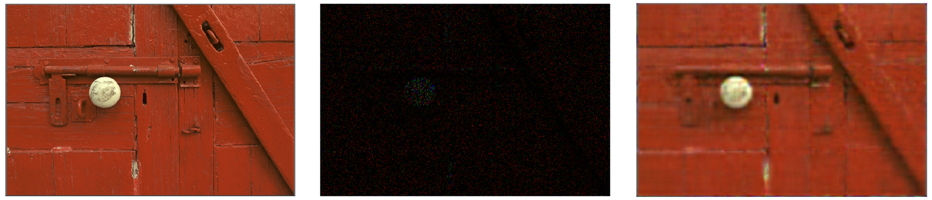

PSNR= 26.7307


Original Image


PSNR= 27.4714

Computer Simulations: Kodak Dataset

Original Image | 95% missing | Reconstructed image



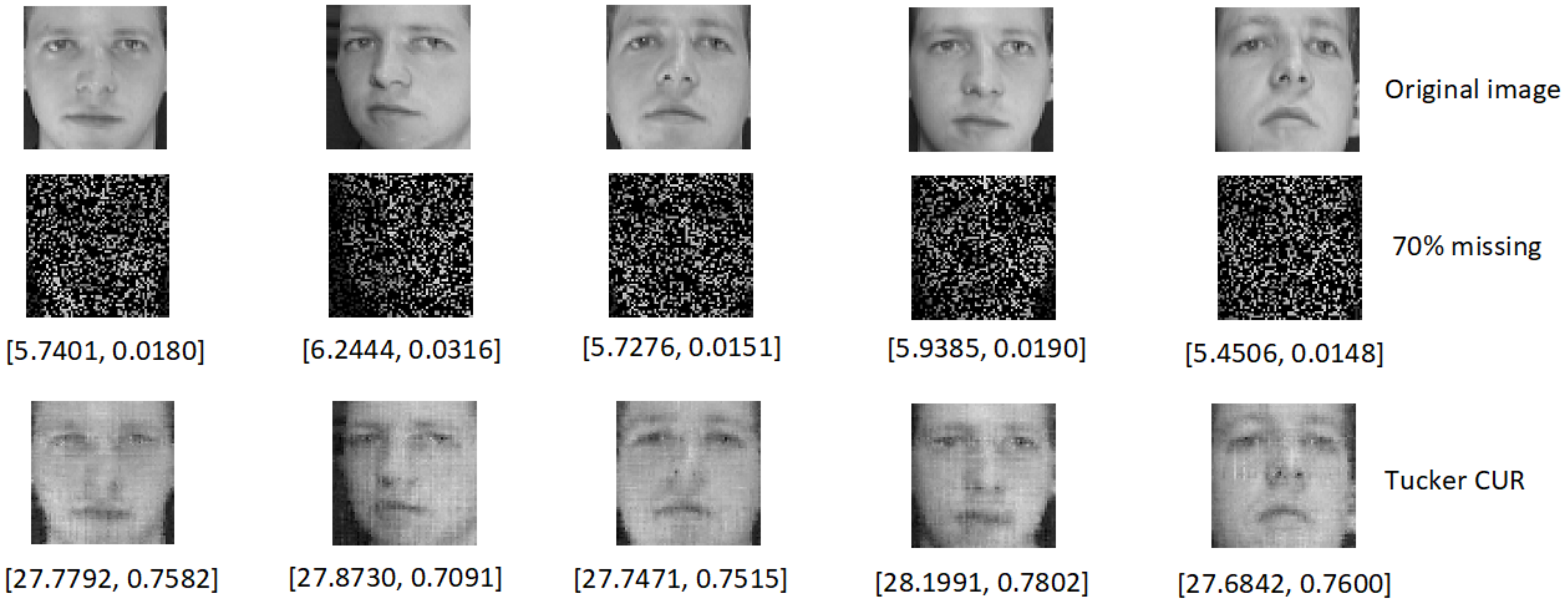
PSNR= 26.8444

PSNR= 24.3587

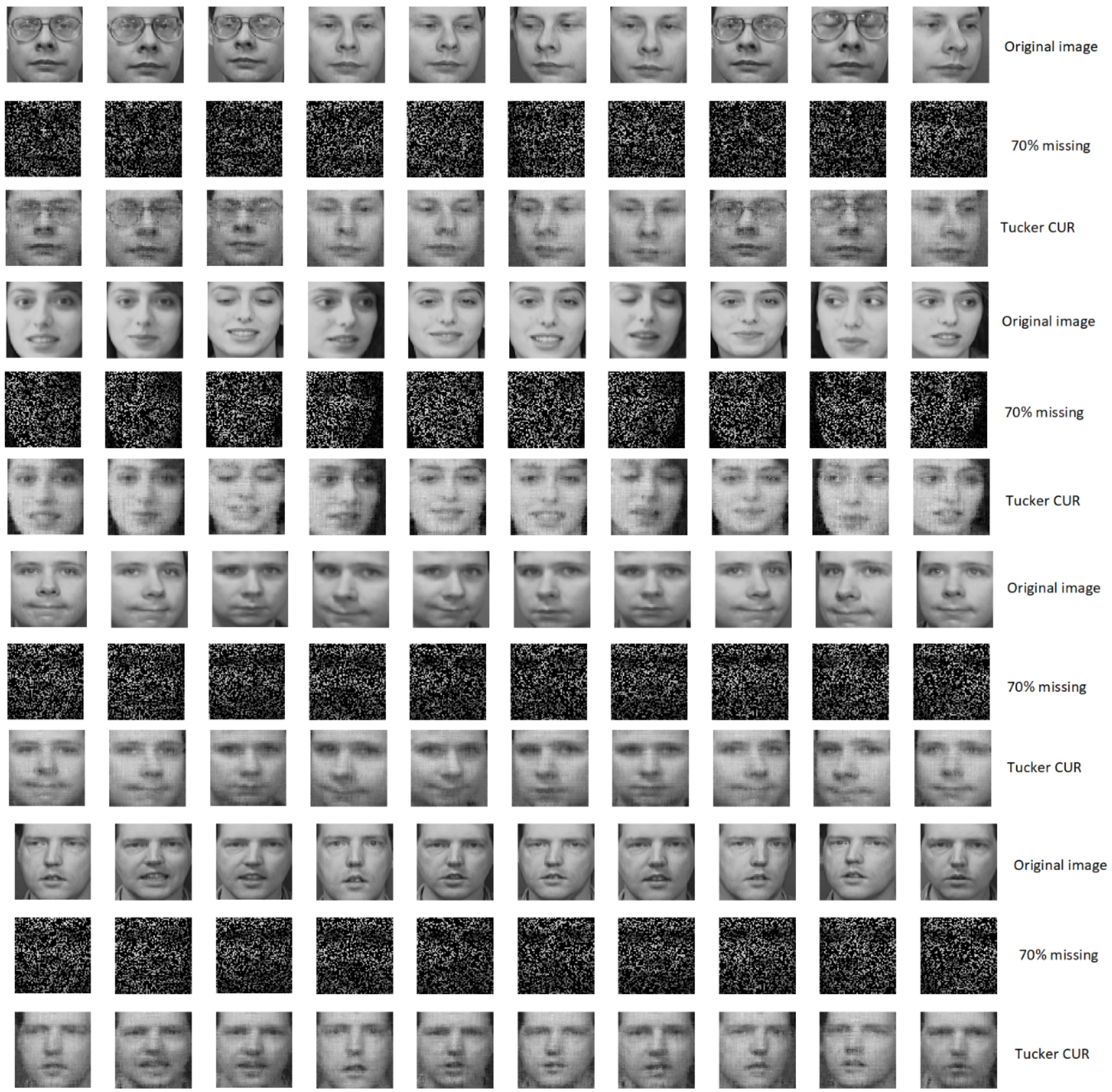
PSNR= 26.0939

PSNR= 22.2255

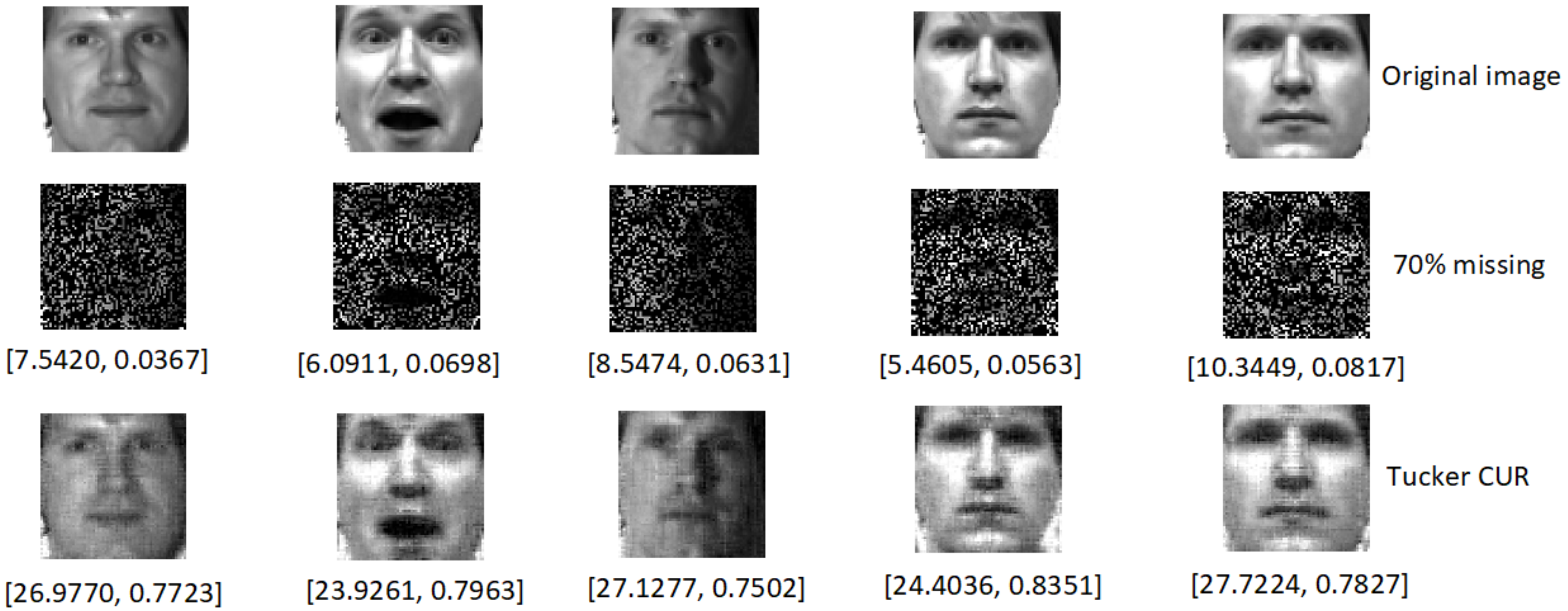
Computer Simulations: ORL Dataset



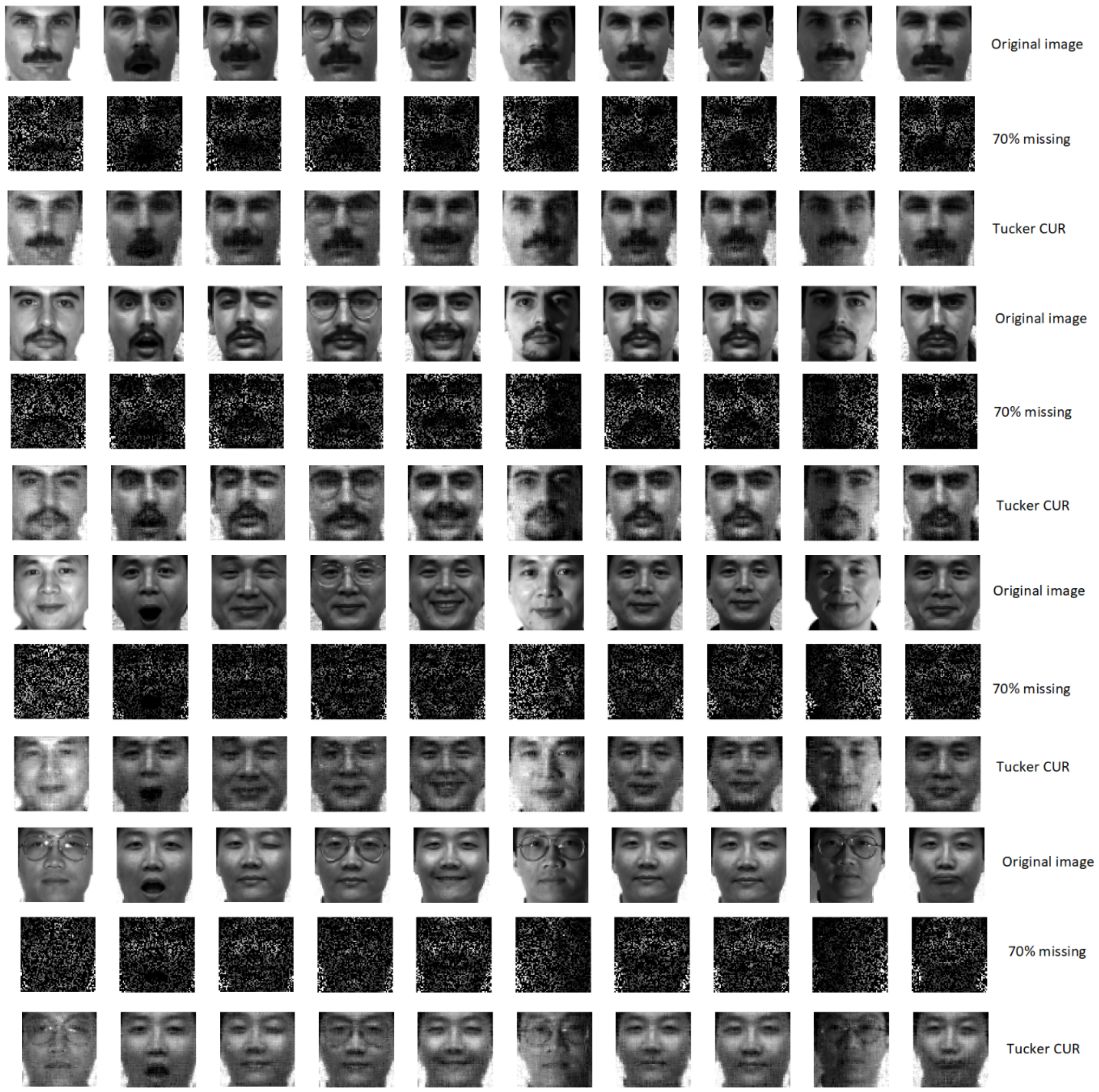
Computer Simulations: ORL Dataset



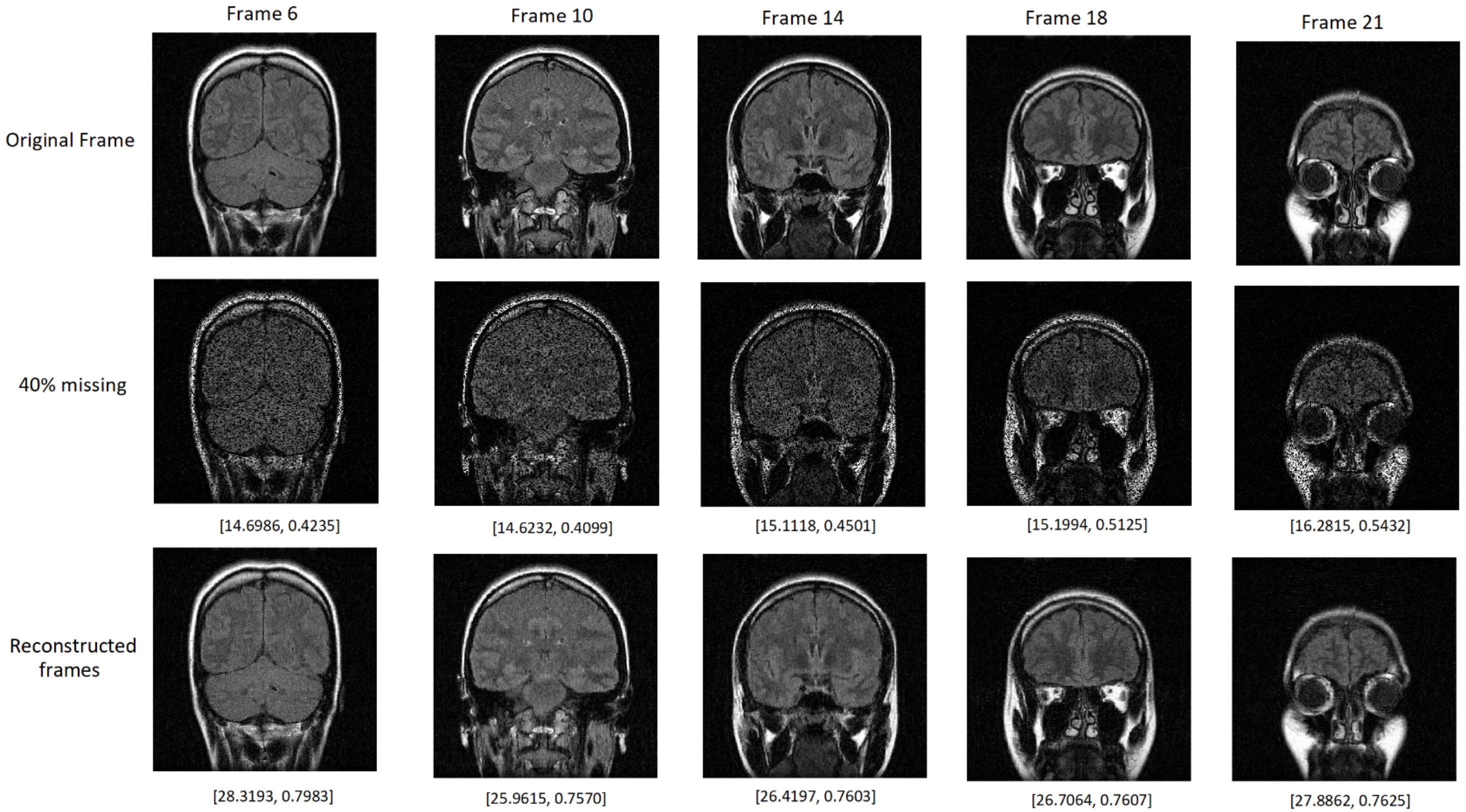
Computer Simulations: Yale Dataset



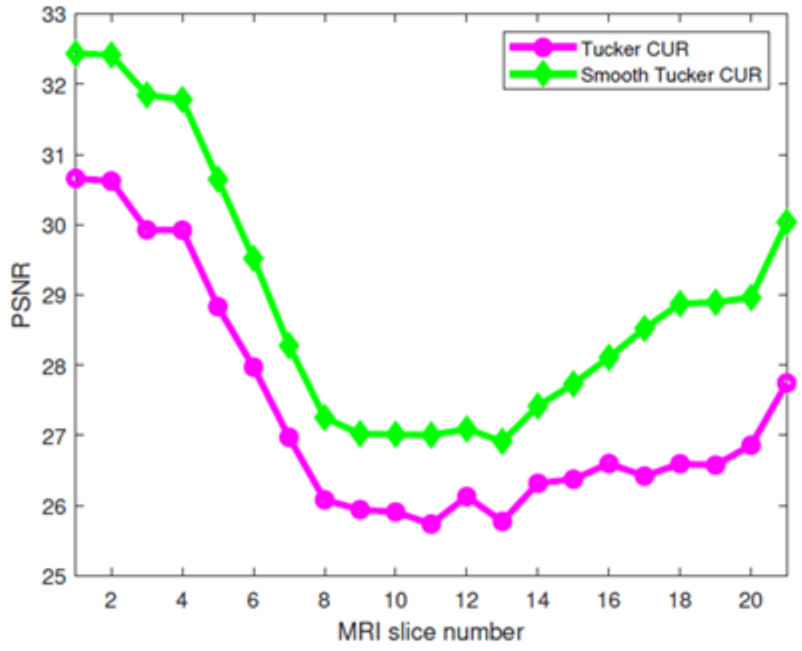
Computer Simulations: Yale Dataset

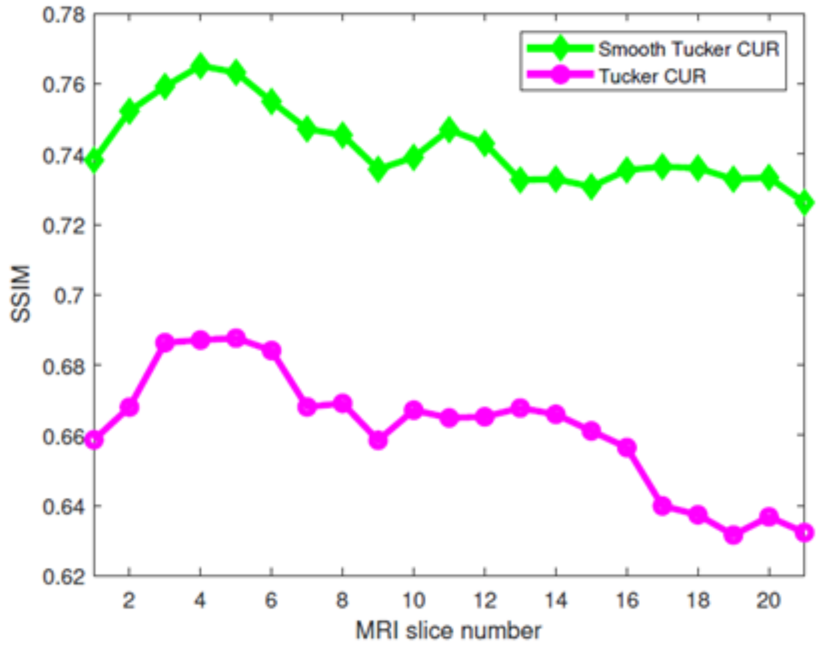


Computer Simulations: MRI Dataset

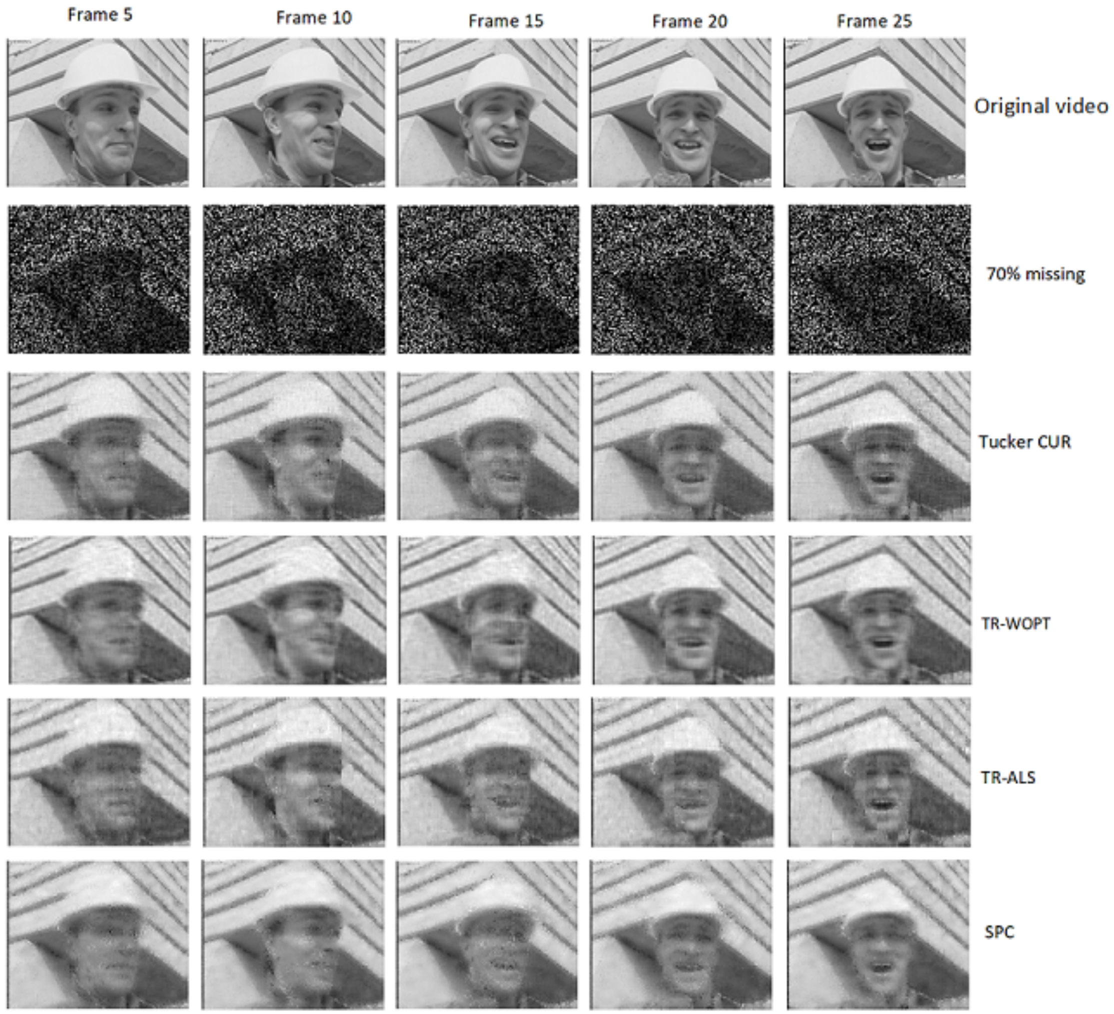


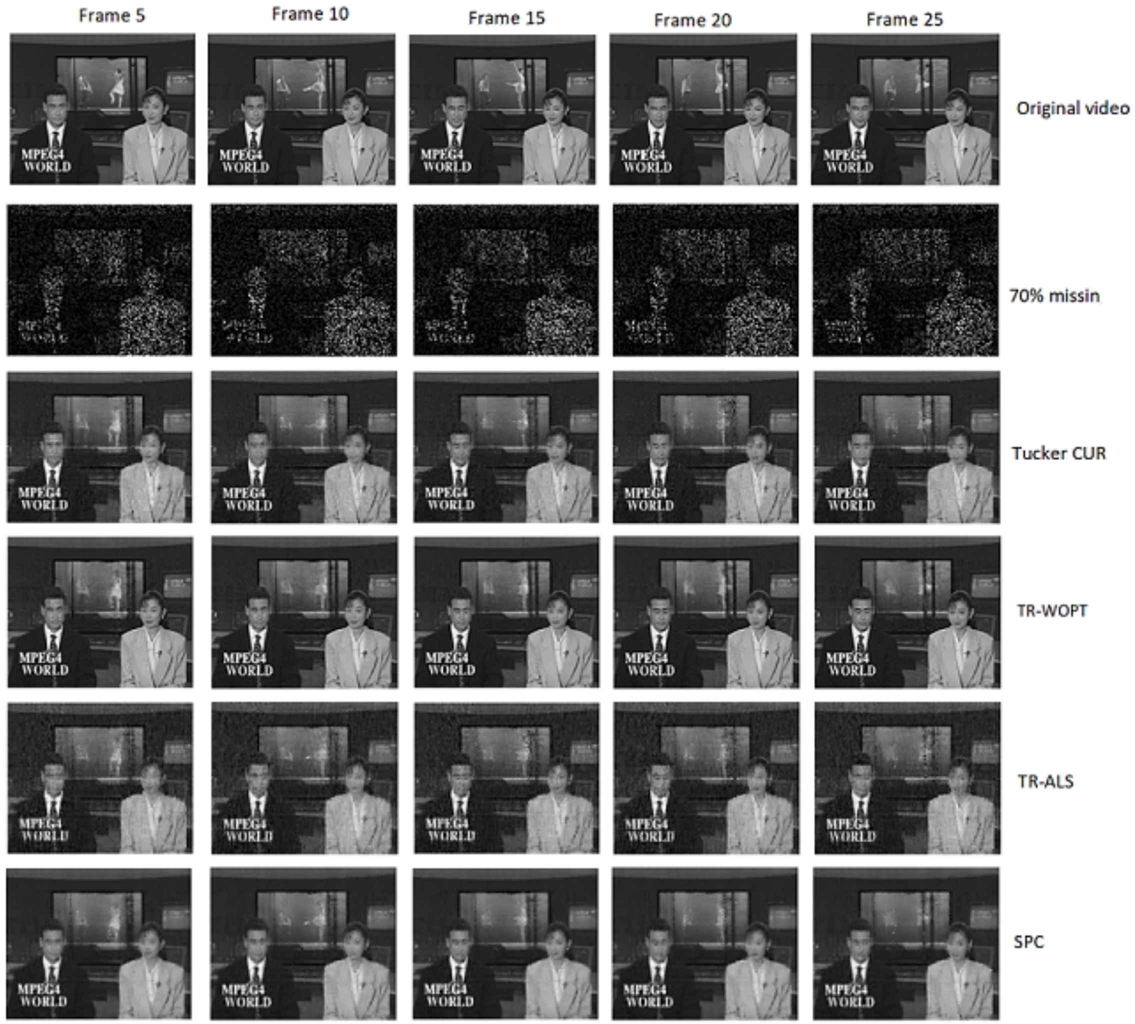
Computer Simulations: MRI Dataset



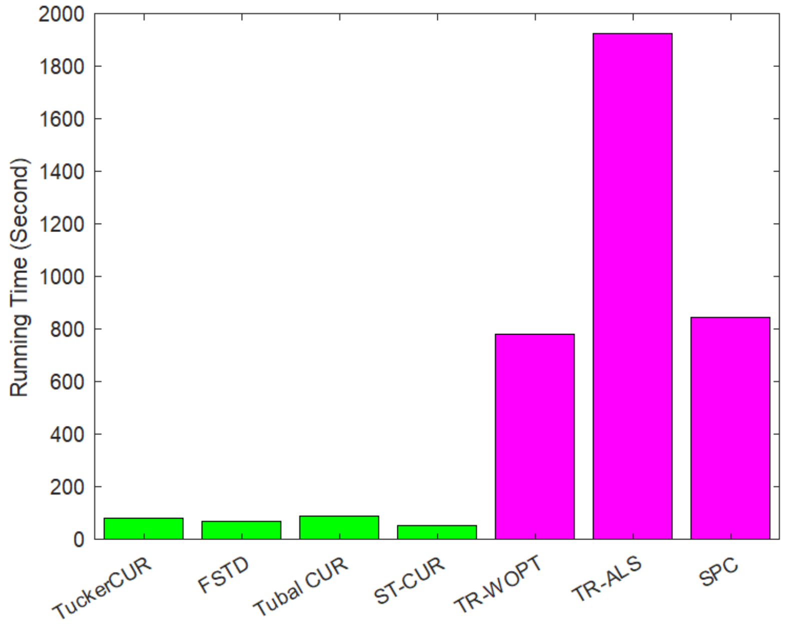


Computer Simulations: Video Dataset

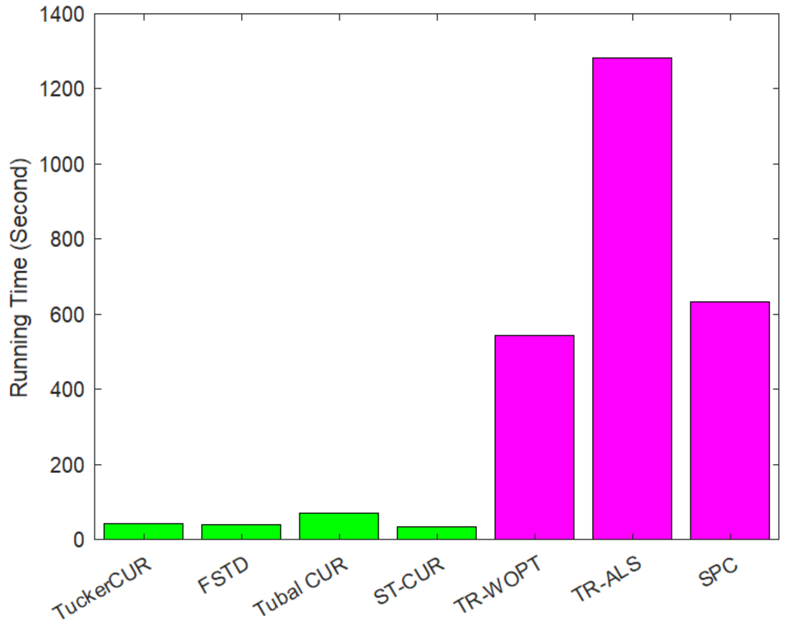




Computer Simulations: Video Dataset

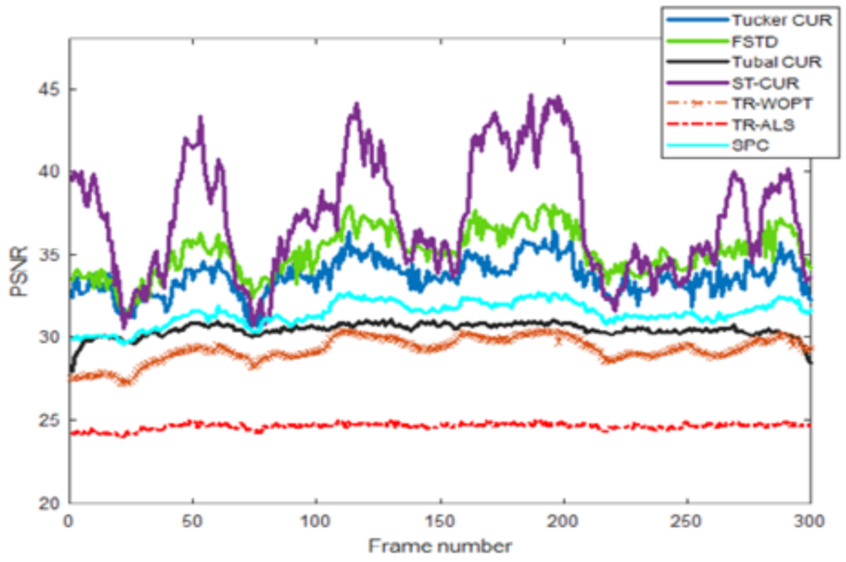


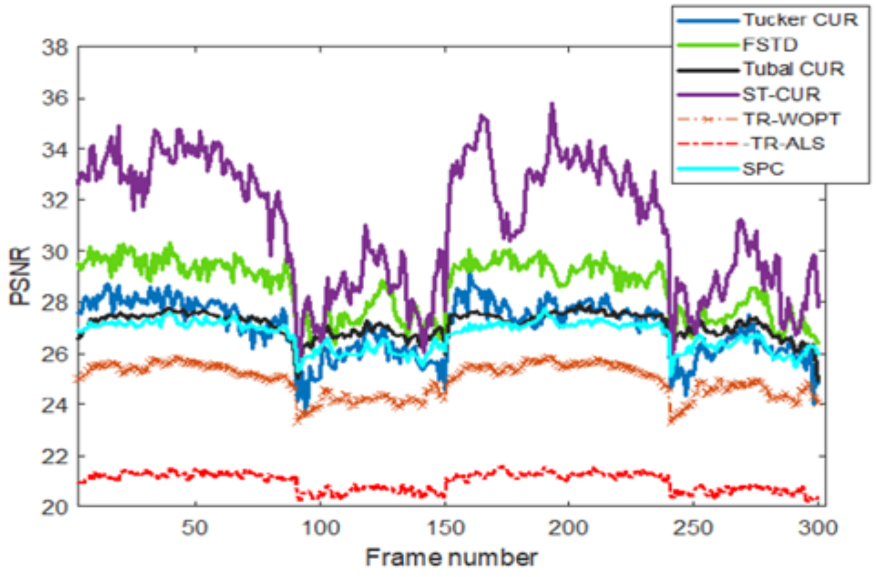
(a)



(b)

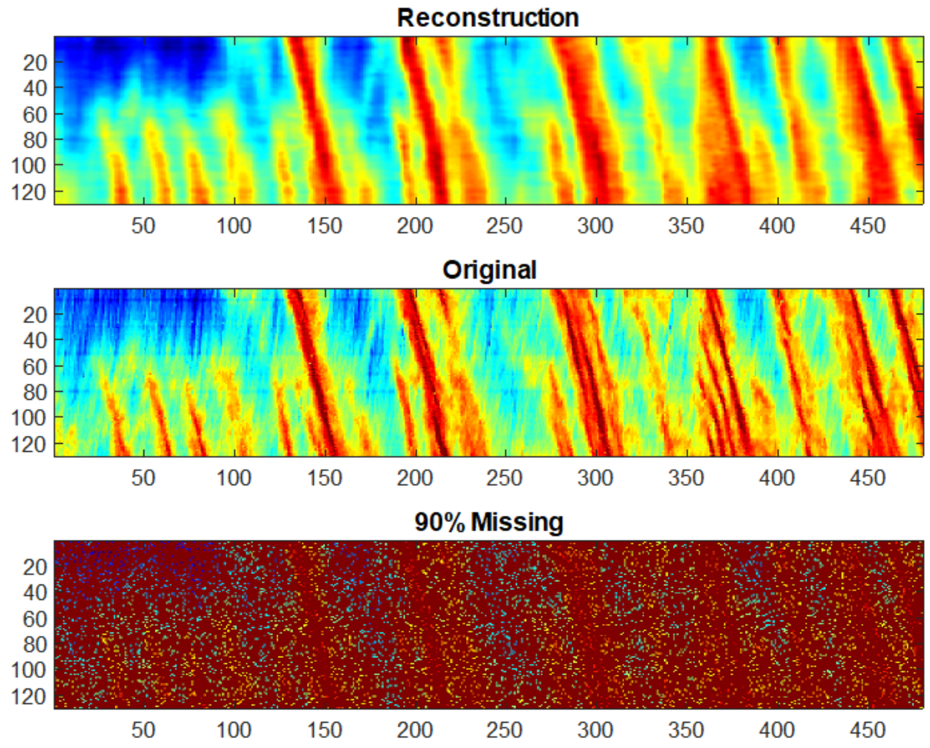
Computer Simulations: Video Dataset

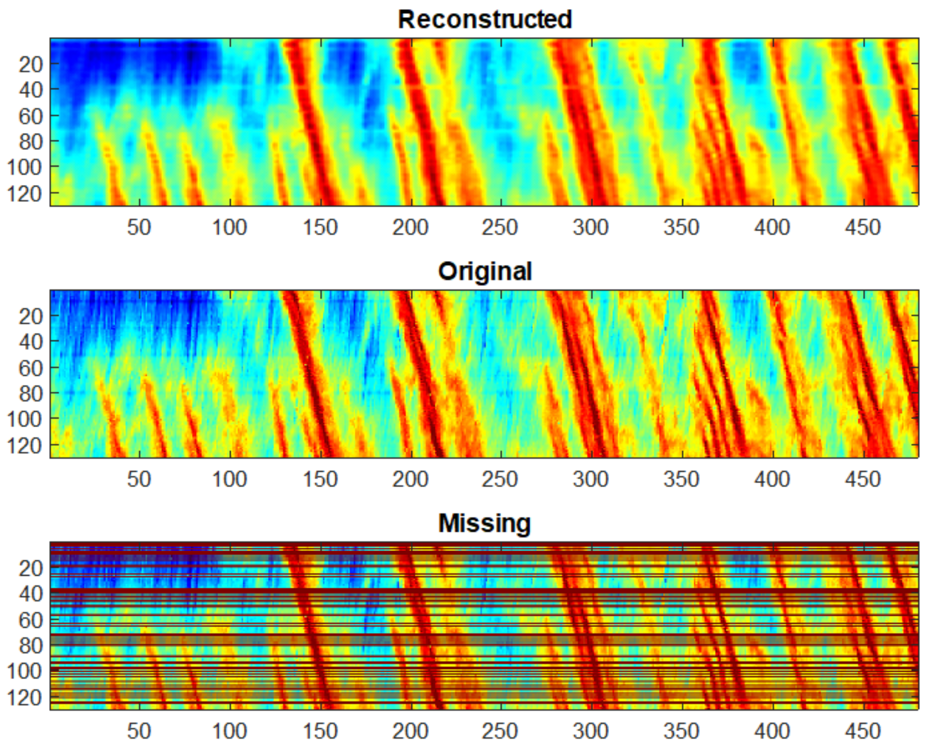




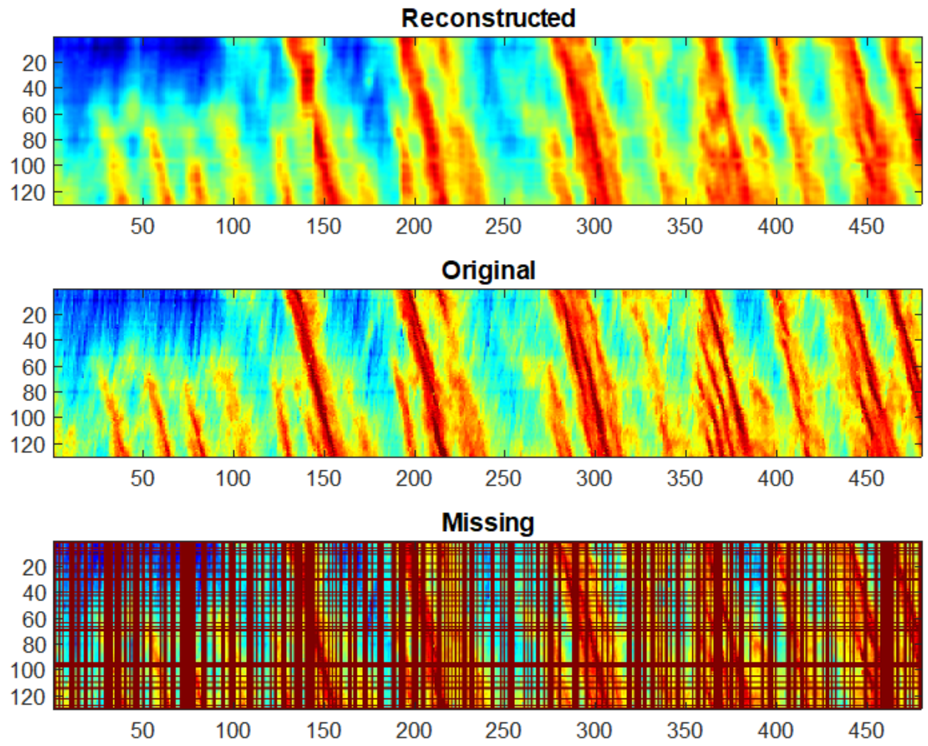
The codes can be accessed at: https://github.com/SalmanAhmadi-Asl/Cross_Tensor_Completion

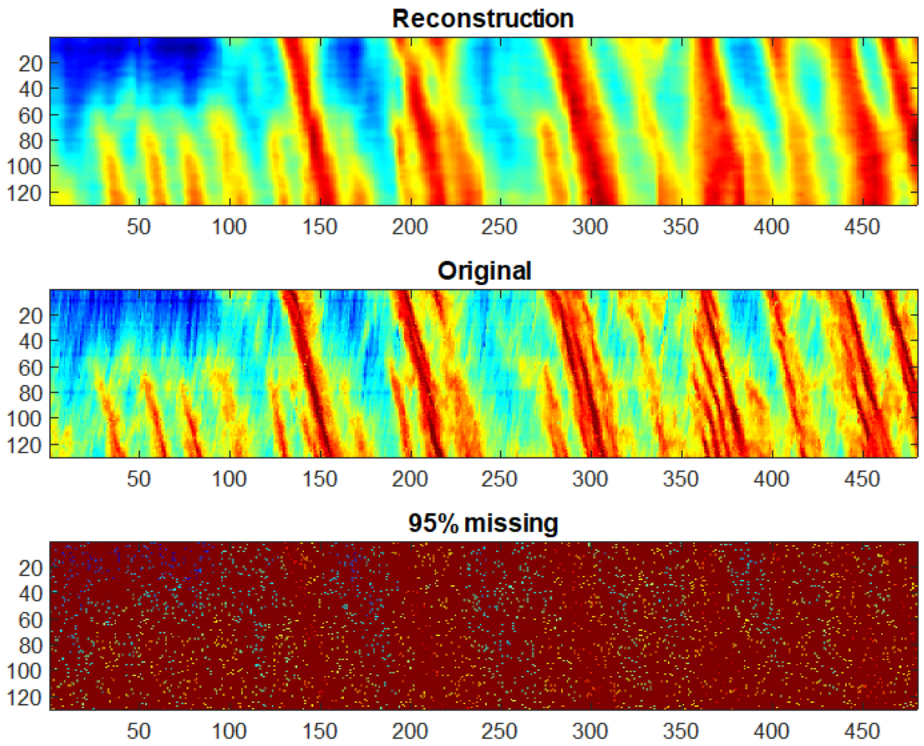
Spatiotemporal dataset: Traffic data





Spatiotemporal dataset: Traffic data





Summary

- **Cross approximation with uniform sampling works quite well for images/videos.**
- **Smoothing the selected fibers/slices provides better reconstruction accuracy.**
- **Image/video completion problem can be performed very fast using CUR approximation methods.**

Thanks for your attentions