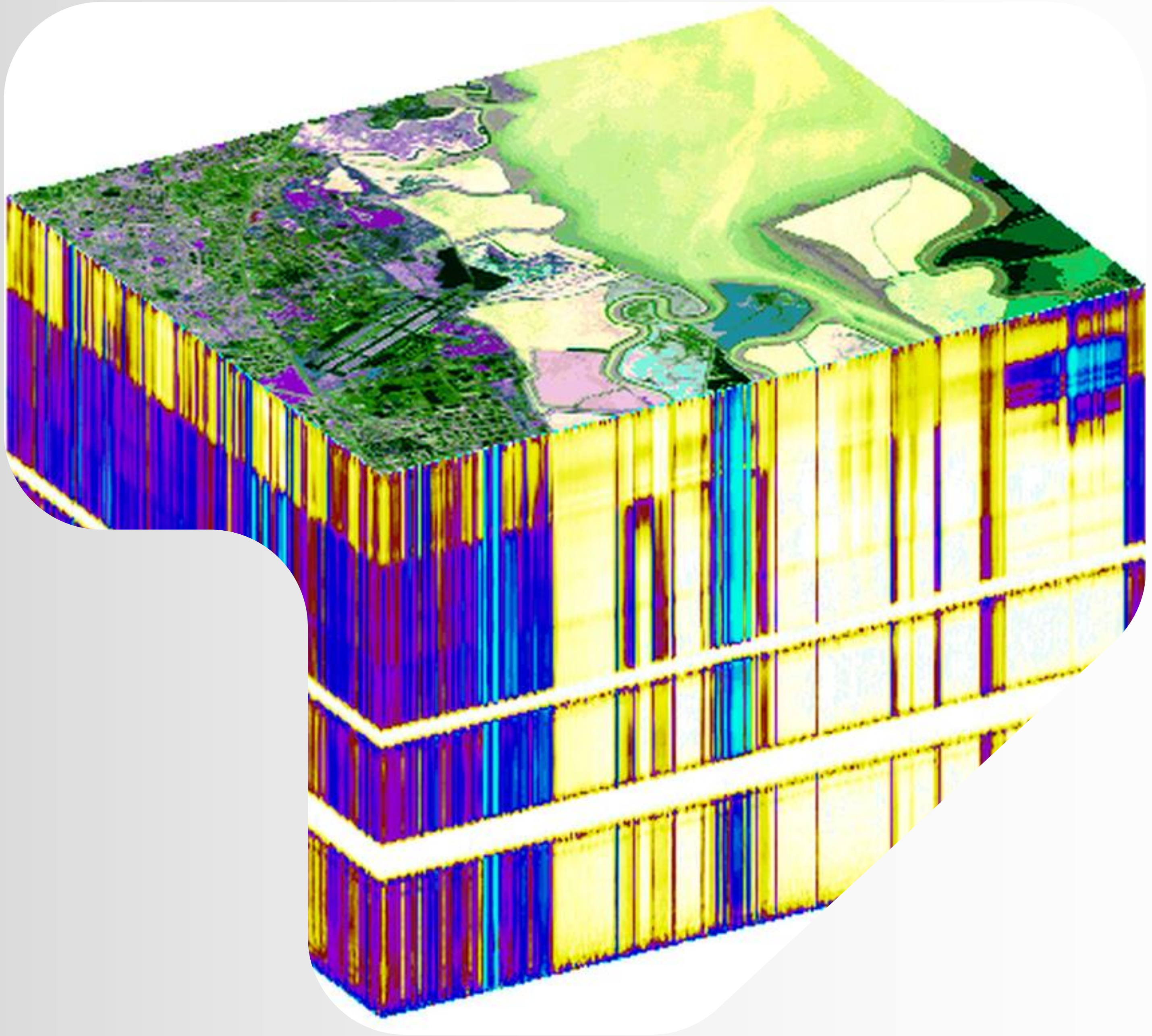
University of Crete

Project Presentation

"Graph and Rank Regularized Matrix Recovery for Snapshot Spectral Image Demosaicing„



Nikos Kontogeorgis
csd4655

Vasilis Papageridis
csd4710

CS-573: Optimization Methods

Introduction

- The objective of Hyperspectral Imaging (HSI) is the acquisition of the spectral profile at each spatial location in the field-of-view of a camera. However, acquiring imagery with high spectral, spatial, and temporal resolution is a formidable task, due to the challenges associated with obtaining 4D measurements.

- Snapshot Spectral Images (SSI) are more feasible and efficient to collect, however that also means that they lack the whole information contained in full HSI. This paper aims to instroduce a new algorithm that aids in reconstructing the full HSI just from a few or even a single SSI.

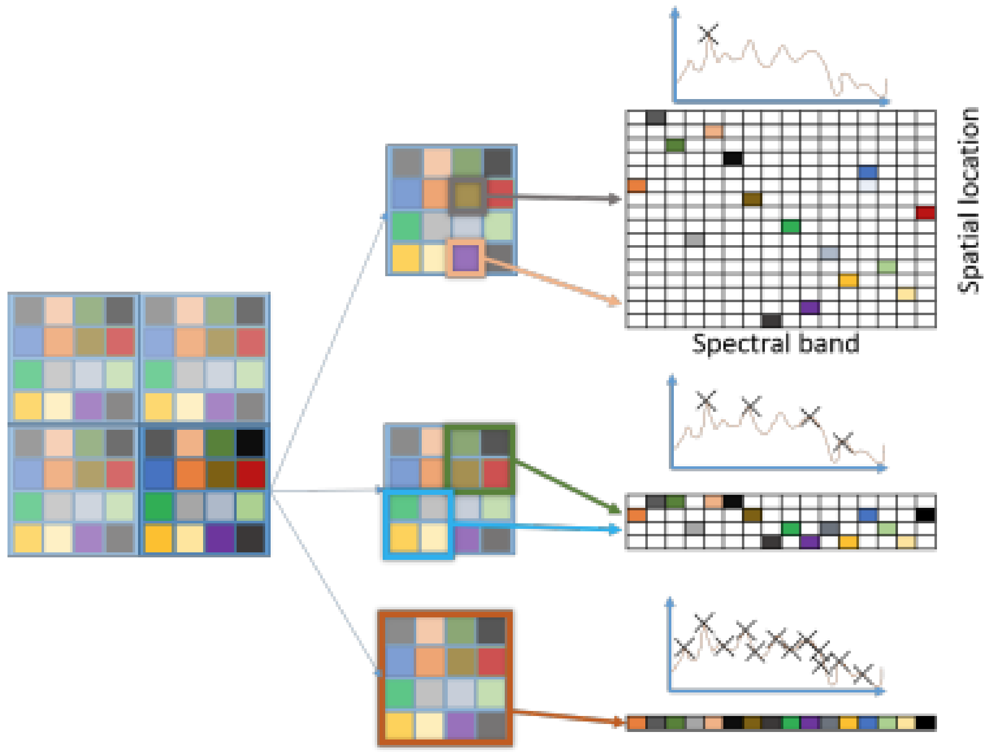


Fig. 2: A frame acquired by an SSI architecture employs a spectral filter pattern to associate a single spectral band to each spatial location - pixel (left), leading to significant sub-sampling (top right). A typical demosaicing approach involves binning together groups of pixels, known as "super-pixels", to produce lower spatial resolution observation with more (middle) or all (bottom right) spectral bands.

Statement of the Problem

The focus is on reconstructing a high-resolution HSI matrix from incomplete (subsampled) data acquired with SSI.

This problem is cast as a missing measurement recovery problem.

It is modeled as follows:

We model the SSI demosaicing problem as the regularized recovery of the spectral measurements matrix $\mathbf{X}$ from a limited set of measurements $\mathcal{A}(\mathbf{M}) \in \mathbb{R}^k$, expressed as

$$\begin{aligned} \min_{\mathbf{X}} \quad & \mathcal{R}(\mathbf{X}) \\ \text{subject to} \quad & \mathcal{A}(\mathbf{M}) = \mathcal{A}(\mathbf{X}) \end{aligned} \tag{5}$$

where $\mathcal{R}$ is a regularization term and $\mathcal{A}$ is the sampling operator.

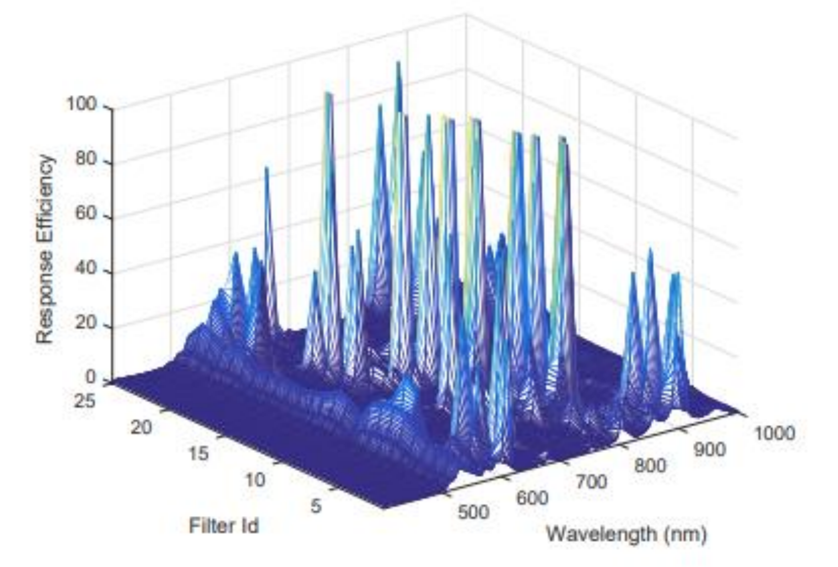


Fig. 3: Spectral response of the 25 spectral filters of the 5×5 IMEC SSI camera in the 400-1000nm range.

Demonstration of the sparsity that exists in data acquired using SSI.

The Proposal:
Graph and Rank regularized Matrix Recovery

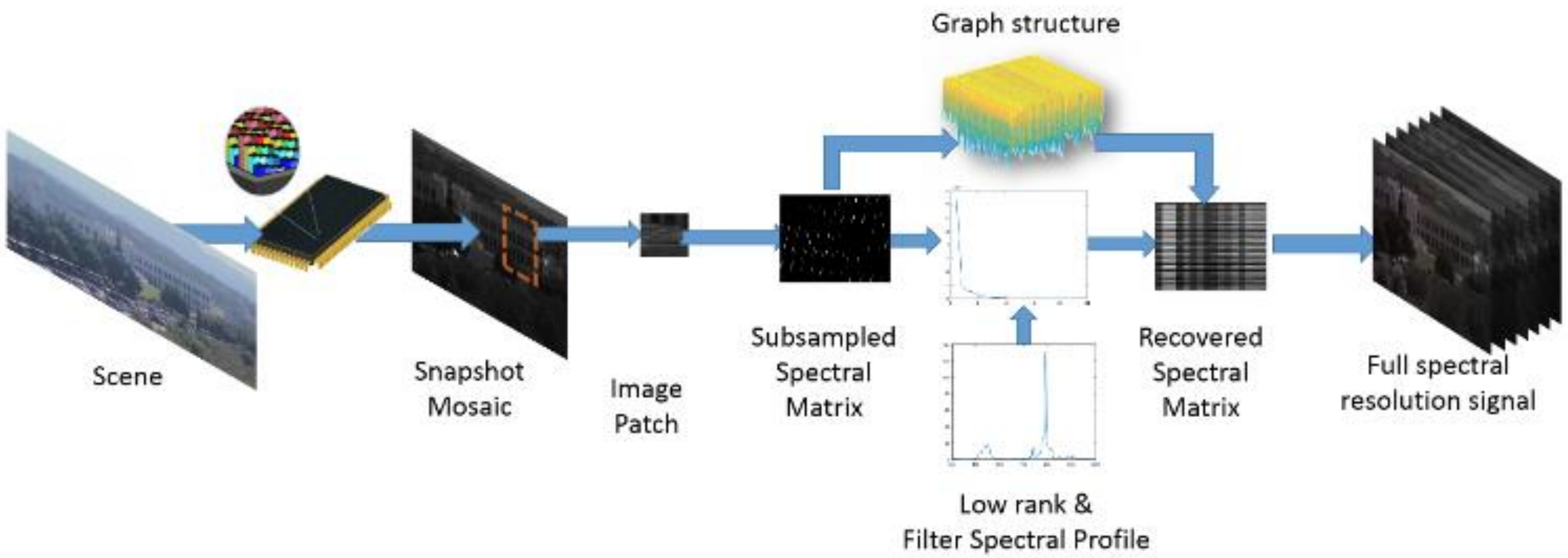


Fig. 4: Overview of acquisition and recovery process. The full spectral profile at a given location undergoes a modulation with the spectral profile of the filter associated with each pixel, producing a two-dimensional spectral snapshot of the scene. Spectral mosaic patches are extracted and transformed to undersampled measurement matrices with a recovered in order to produce the full spectral content of the scene.

The Proposal:
Graph and Rank regularized Matrix Recovery

A. *Rank Regularization*

$$\begin{aligned} &\underset{\mathbf{X}}{\text{minimize}}\ rank(\mathbf{X}) \\ &\text{subject to}\ \ \mathcal{A}(\mathbf{X}) = \mathcal{A}(\mathbf{M}) \end{aligned}$$

Formally, one can recover an accurate approximation $\mathbf{X}$ of the matrix $\mathbf{M}$ from a small number of entries by solving the minimization problem

$$\underset{\mathbf{X}}{\text{minimize}}\ \|\mathbf{X}\|_* + \|\mathcal{A}(\mathbf{M}) - \mathcal{A}(\mathbf{X})\|_F \tag{7}$$

B. *Graph Regularization*

The graph Laplacian matrix is an approximation to the Laplace-Beltrami operator and is responsible for encoding the underlying manifold structure of the data such to spatial proximity. In order to enforce the graph smoothness constraint, we introduce the associated regularization term given by

$$\|\nabla_{\mathcal{G}}\mathbf{X}\|_F = tr(\mathbf{X}\mathbf{L}\mathbf{X}^T). \tag{13}$$

The Proposal:
Graph and Rank regularized Matrix Recovery

C. Graph and Rank regularized Matrix Recovery (GRMR)

$$\underset{\mathbf{X}}{\text{minimize}}\ \|\mathbf{X}\|_* + \|\mathcal{A}(\mathbf{X}) - \mathbf{y}\|_2^2 + \beta\ tr(\mathbf{X}\mathbf{L}\mathbf{X}^T)$$

In this work, we adopt the proximal splitting method, by defining two functions $f(\mathbf{X})$ and $g(\mathbf{X})$, such that the minimization in (14) is equivalently expressed as

$$\underset{\mathbf{X}}{\text{minimize}}\ f(\mathbf{X}) + g(\mathbf{X}) \tag{15}$$

where

$$f(\mathbf{X}) = \|\mathbf{y} - \mathcal{A}(\mathbf{X})\|_2^2 + \beta tr(\mathbf{X}\mathbf{L}\mathbf{X}^T) \tag{16}$$

and

$$g(\mathbf{X}) = \|\mathbf{X}\|_* \tag{17}$$

The solution to the minimization in (15) can be obtained by the proximal gradient method [36], an iterative approach where each iteration is given by

$$\mathbf{X}^{t+1} = \mathbf{prox}_{\lambda^t g}(\mathbf{X}^t - \lambda^t \nabla f(\mathbf{X}^t)) \tag{18}$$

For the GRMR objective, the gradient of $f(\mathbf{X}^t)$ in (16) is given by the sum of the gradient with respect to the l_2 norm and the trace

$$\nabla f(\mathbf{X}^t) = \mathcal{A}^*(\mathcal{A}(\mathbf{X}^t) - \mathbf{y}) + \beta \mathbf{X}^t \mathbf{L}^T \tag{19}$$

The proximal operator of the nuclear norm in (17) is given by a thresholding operator applied on the singular values of matrix $\mathbf{X}$. Two versions of thresholding operators, a soft and a hard, can be used. The soft-thresholding operator is given by

$$\mathcal{D}_\tau(\mathbf{X}) = \mathbf{U}\mathcal{D}_\tau(\mathbf{\Sigma})\mathbf{V}^T,\ \mathcal{D}_\tau(\mathbf{\Sigma}) = \text{diag}(\{\sigma_i - \tau\}_+) \tag{21}$$

In this work, we employ the hard-thresholding operator

$$\mathcal{H}_\tau(\mathbf{X}) = \mathbf{U}\mathcal{H}_\tau(\mathbf{\Sigma})\mathbf{V}^T,\ \mathcal{H}_\tau(\mathbf{\Sigma}) = \begin{cases} \sigma_i & \text{if } i \leq r \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

such that only the r top singular values are preserved, giving rise to pre-determined rank matrices.

The Algorithm:
Graph and Rank regularized Matrix Recovery

Algorithm 1: Graph and Rank Regularized Matrix Recovery (GRMR)

Input: The sampled measurements matrix $\mathbf{M}$,
the regularization parameter β and rank estimate k,
the maximum number of iterations $limit$ or
variation less than δ between successive iterations.

Output: The estimated spectral measurements matrix $\mathbf{X}$.

```
initialization
  Initialize X^(0) from interpolation, t = 0
  Calculate the Degree and Weight matrix from X^(0) and
  estimate the Laplacian matrix L according to Eq. (12)
while t < limit or ||X^(t+1) − X^(t)||_2 ≤ δ do
    Estimate update parameter μ_t according to Eq. (24)
    Estimate intermediate variable Z^(t)
```

$$\mathbf{Z}^{(t)} = \mathbf{X}^{(t)} - \mu_t(\mathcal{A}^*(\mathcal{A}(\mathbf{X}^{(t)}) - \mathbf{y})) - \beta\mathbf{X}^{(t)}\mathbf{L}^T$$

```
    Obtain the updated estimate of matrix X^(k+1):
```

$$(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}) = \mathrm{SVD}(\mathbf{Z}^{(t)})$$
$$\mathbf{X}^{(t+1)} = \mathbf{U}\mathcal{H}_k(\mathbf{\Sigma})\mathbf{V}^T$$

```
    set  t ⟵ t + 1
end while
```

1. **Input**: Subsampled measurements matrix M, regularization parameter β, rank estimate k, maximum iterations *limit* or convergence threshold δ.
2. **Initialization**: Interpolate to initialize spectral measurements matrix $X^{(0)}$, set iteration $t = 0$.
3. **Degree and Weight Matrix Calculation**: Calculate Degree and Weight matrix from $X^{(t)}$ and estimate the Laplacian matrix L.
4. **Iterative Update**:
   (a) Estimate intermediate variable $Z^{(t)}$ using current $X^{(t)}$, sampling operator A, measurements matrix y, and Laplacian L.
   (b) Apply regularization involving graph structure and rank penalty.
5. **Matrix Update**: Update X using SVD of $Z^{(t)}$, retaining only k largest singular values and vectors.
6. **Convergence Check**: Repeat until $\|X^{(t+1)} - X^{(t)}\|_F \leq \delta$ or iteration count exceeds *limit*.
7. **Output**: Final spectral measurements matrix X after convergence.

Our Twist

Remember that:

In this work, we employ the hard-thresholding operator

$$\mathcal{H}_\tau(\mathbf{X}) = \mathbf{U}\mathcal{H}_\tau(\mathbf{\Sigma})\mathbf{V}^T, \ \mathcal{H}_\tau(\mathbf{\Sigma}) = \begin{cases} \sigma_i & \text{if } i \le r \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

such that only the r top singular values are preserved, giving rise to pre-determined rank matrices.

What we did

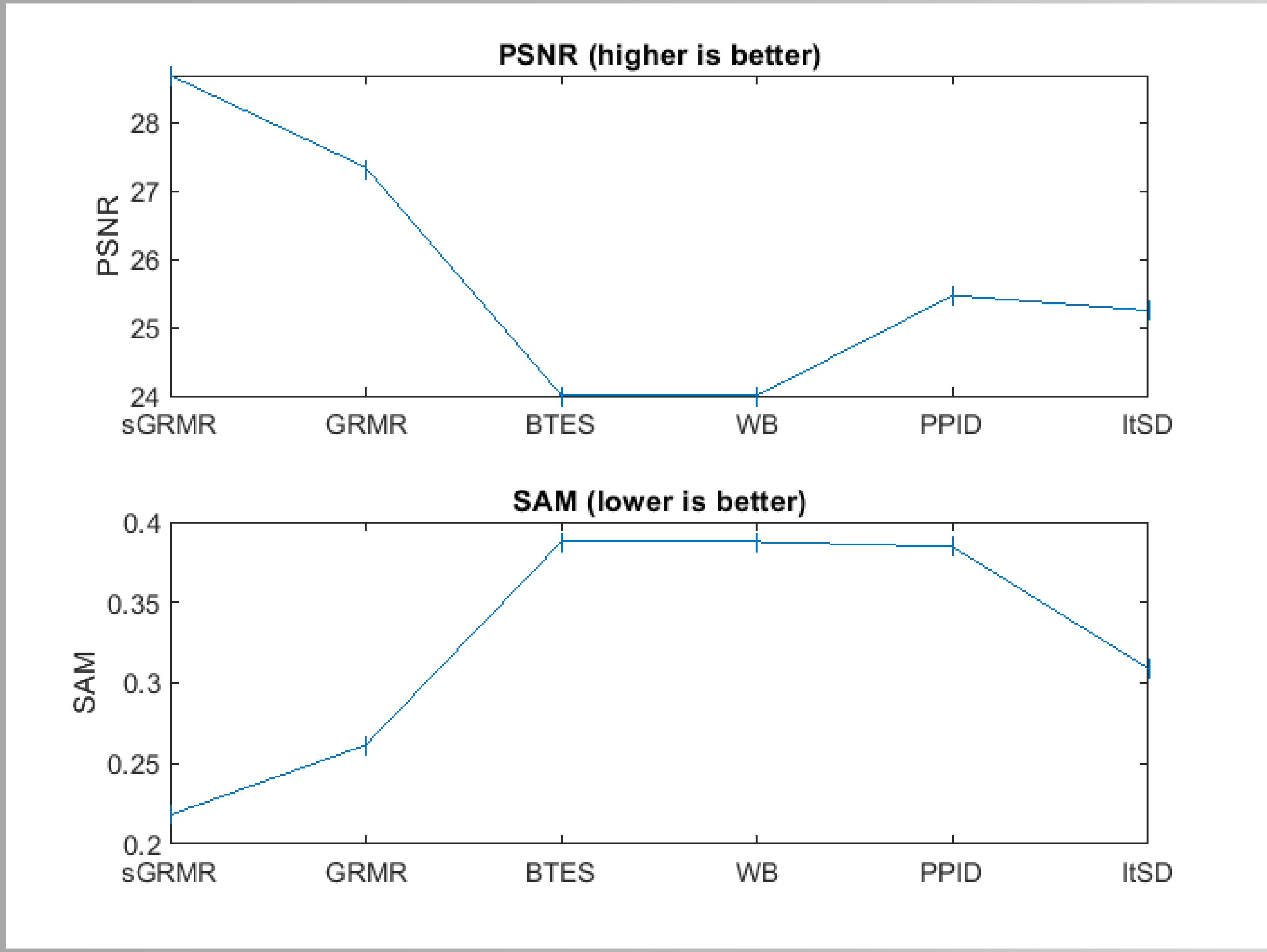
- Toss a coin (i.e. the following have 50% chance each):
  - If heads, do as above
  - If tails, keep the top $r' \leqslant r$ singular values instead (a random int)

We aimed to introduce an element of stochasticity to the algorithm. After a few attempts, stochasticity in this form seemed to be successful.

We'll refer to it as sGRMR (stochastic GRMR), for lack of a better name.

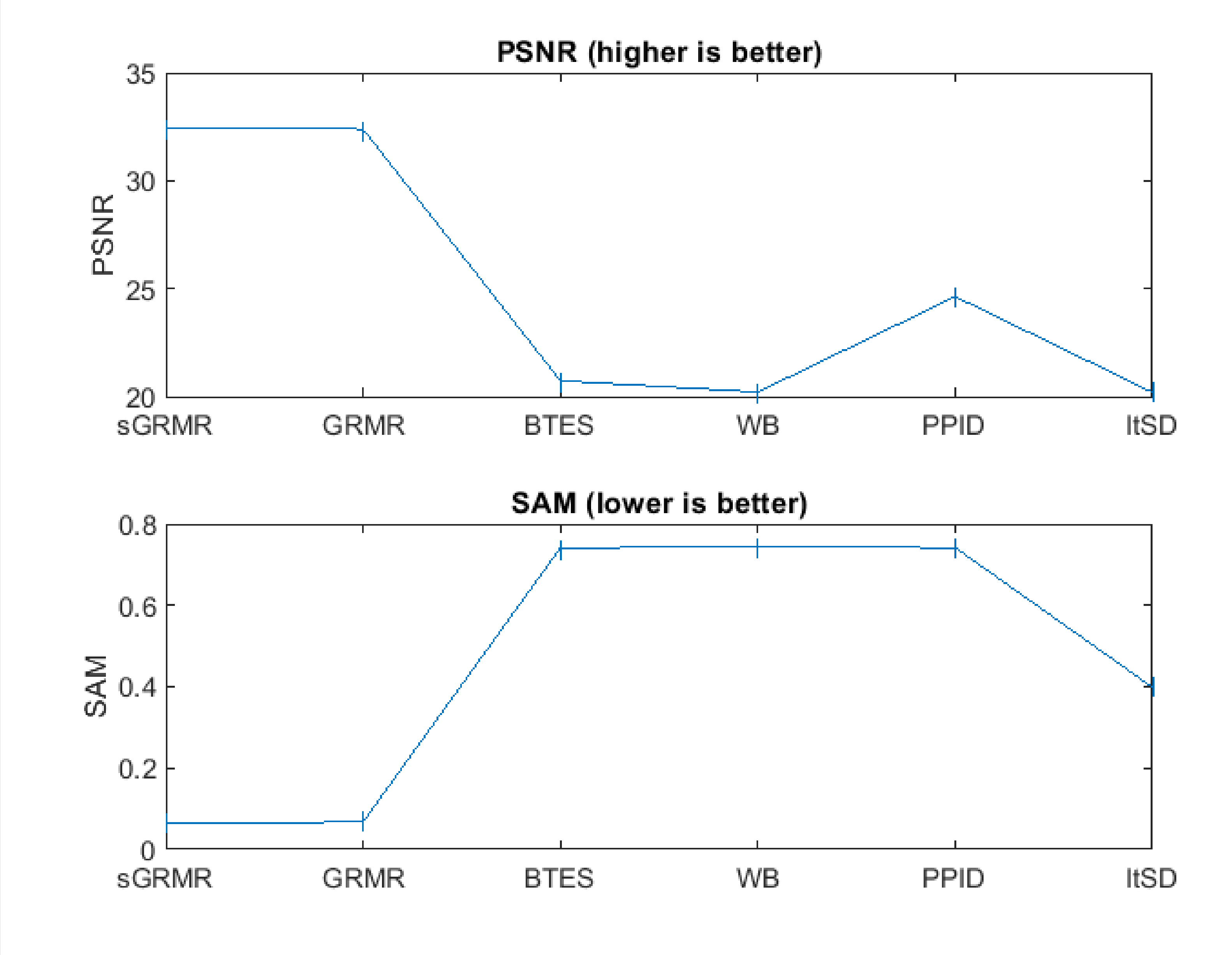
Our Twist

Evaluation on the CAVE dataset:



Notable improvement :)

Evaluation on the SENTINEL dataset:



Barely visible improvement :|

Ερωτήσεις?
Σχόλια?
Σκέψεις?
Προβληματισμοί?

Thank You!