Reordering for Better Compressibility: Efficient Spatial Sampling in WSN

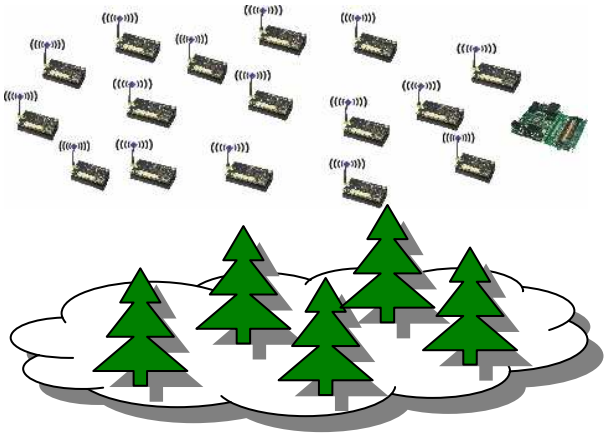
Mohammadreza Mahmudimanesh,
Abdelmajid Khelil and Neeraj Suri

Technische Universität Darmstadt

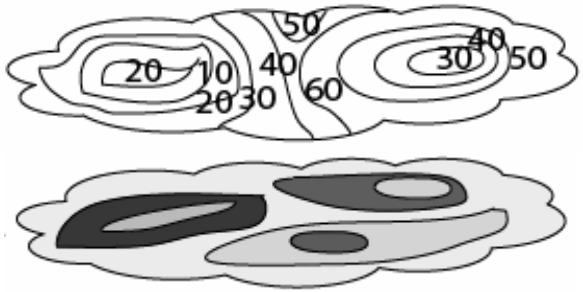
SUTC 2010

Motivation

- Wireless Sensor Networks (WSN)
  - Static battery-powered sensor nodes (SN)
    - Computation, comm. & energy limited!
  - A powerful sink
  - Redundant node deployment
- Emerging method to monitor physical environment



- Several applications require node-level accurate sampling
  - Map-based sensing
  - Prediction/Forecast
- Further main application/design requirements
  - Energy-/computation-efficiency
  - Balanced load among sensor nodes

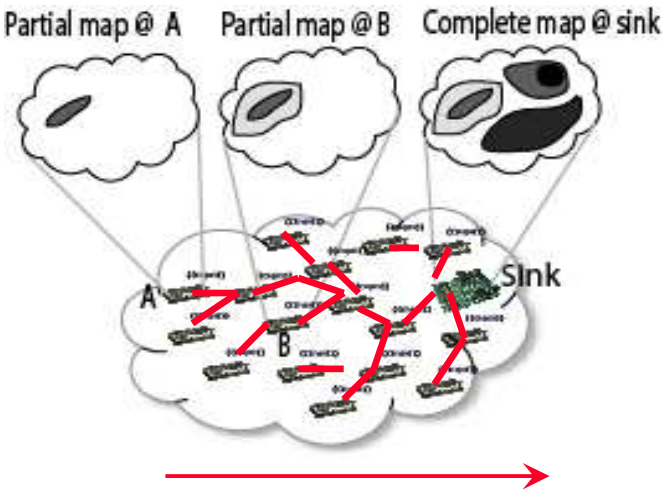


→ Uniform residual energy distribution required

Data Collection in WSN

- Naive: Each SN sends value to sink
  - But, most signals sampled from natural environments are compressible, so *a vast amount of acquired data is unnecessary*.
- In-network aggregation
- Spatial/temporal suppression
- In-network compression
  - SN sends model parameters instead of absolute values
  - Entropy coding, transform coding ..

→ Reduces global traffic, but significant computation overhead + unbalanced energy consumption

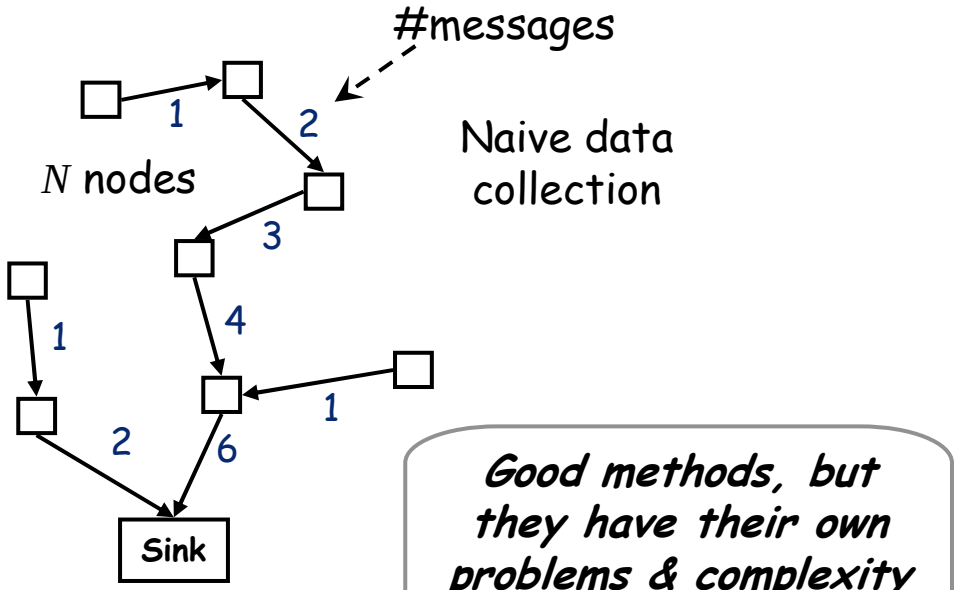


Compressive Wireless Sensing (CWS)

- Based on generic compressive sensing (CS) theory
  - Concrete mathematical foundations

- CWS compression
  - Same computation and comm. overhead on all SNs
  - Simple implementation at SNs (in expense of complexity at sink)
  - Universal:
    - no presumption about topology
    - no presumption about signal, however, inefficient for non-compressible signals

Related Work and Objectives

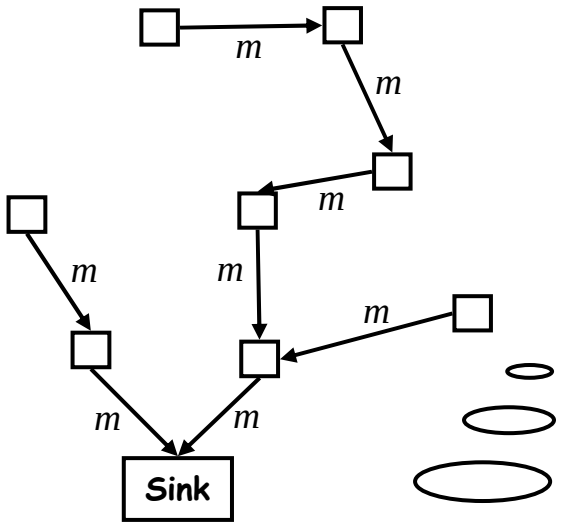


Good methods, but they have their own problems & complexity

* In-network aggregation
* suppression
* Model-based
* In-network compression

etc.

CWS



Remember: m would be $<<N$ only when the signal is *compressible enough*

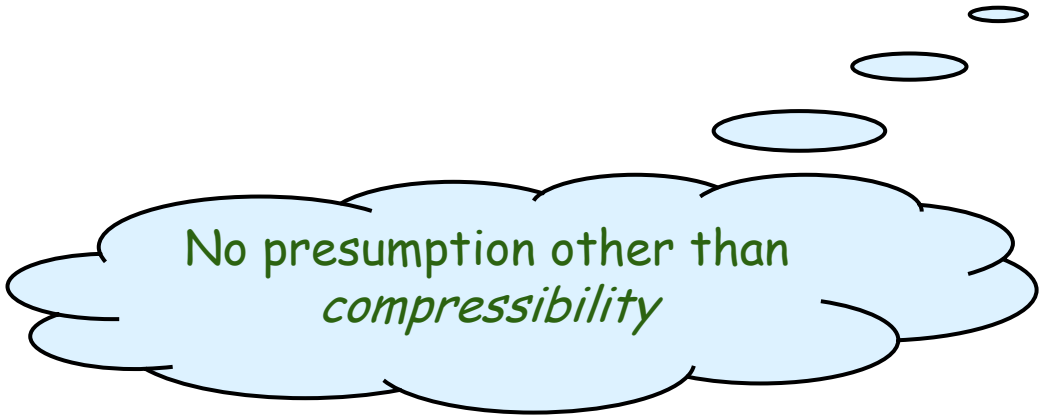
Objective: Improve CWS compressibility

Outline

- Motivation
- Related Work
- System Model
- Brief introduction to Compressive Wireless Sensing (CWS)
- Reordering for better Compressibility
  - Improving CWS through reordering
- Performance Evaluation
- Conclusion

System Model

- System model
  - N *static* sensor nodes and one sink
  - Bi-directional multihop communication
  - Underlying routing + aggregation tree
- Preconditions
  - Discrete environmental signal $\mathbf{f}$ is compressible (in space)
  - Environment doesn't change too drastically over time



Compressive Sampling (CS)

- Sampling/sensing basis Φ

$$\mathbf{f} = y_1\varphi_1 + y_2\varphi_2 + \ldots + y_N\varphi_N$$
where φ_i's are the basis vectors of Φ

- Representation basis Ψ

$$\mathbf{f} = x_1\psi_1 + x_2\psi_2 + \ldots + x_N\psi_N$$
where ψ_i's are the basis vectors of Ψ

- Preconditions
  - The signal is compressible (sparse in representation basis)
  - Select a suitable sampling basis (*coherence*)

CS – Terms & Definitions

- Sparsity
  - Vector $\mathbf{f}$ of size N is S-sparse when it has at most S nonzero elements, and all other $N-S$ elements are zero
- Coherence
  - The coherence between the sensing basis Φ and the representation basis Ψ is defined as $\mu(\Phi,\Psi)$

$$\mu(\Phi,\Psi) = \sqrt{n} \cdot \max_{1 \le k,\, j \le N} \left| \varphi_k \cdot \psi_j \right|$$

- Undersampling
  - Measuring a subset of the coefficients of projection of $\mathbf{f}$ on Φ :

$$\mathrm{M} \subset \{1, \ldots, N\} \quad \text{so that} \quad m = \left|\mathrm{M}\right| < N$$

$$\forall k \in \mathrm{M} \quad y_k = \left\langle \mathbf{f}, \varphi_k \right\rangle$$

CS - Reconstruction

- Signal vector $\mathbf{f}_{N\times 1}$ can be recovered from measurement vector $\mathbf{y}_{m\times 1}$ with probability of success exceeding $1\text{-}\delta$ when:

$$m \geq C \cdot \mu^2(\Phi, \Psi) \cdot S \cdot \log\left(\frac{N}{\delta}\right)$$

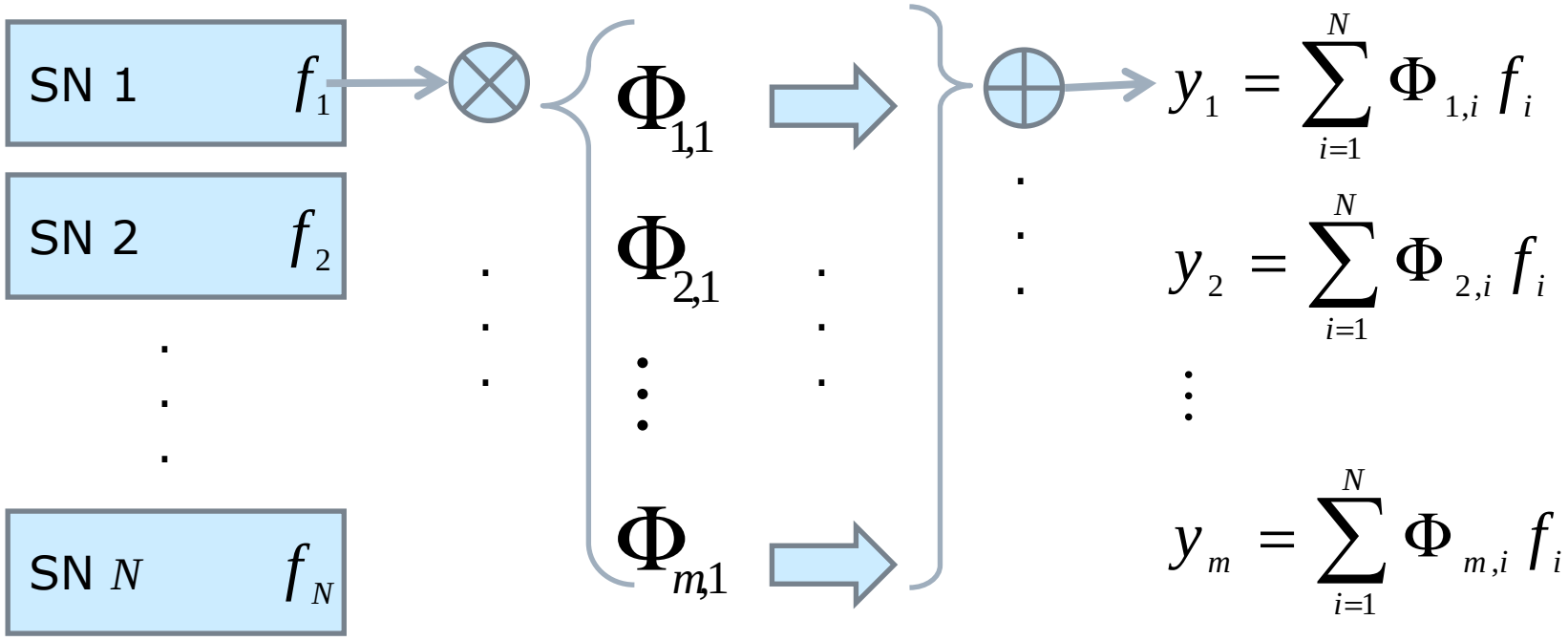
and $\mathbf{f}$ being expected to be S-Sparse in the Ψ-basis

- Reconstructed signal $\mathbf{f}^*$ is given by $\mathbf{f}^*=\mathbf{\Psi x}^*$ where $\mathbf{x}^*$ is the solution to the convex optimization program:

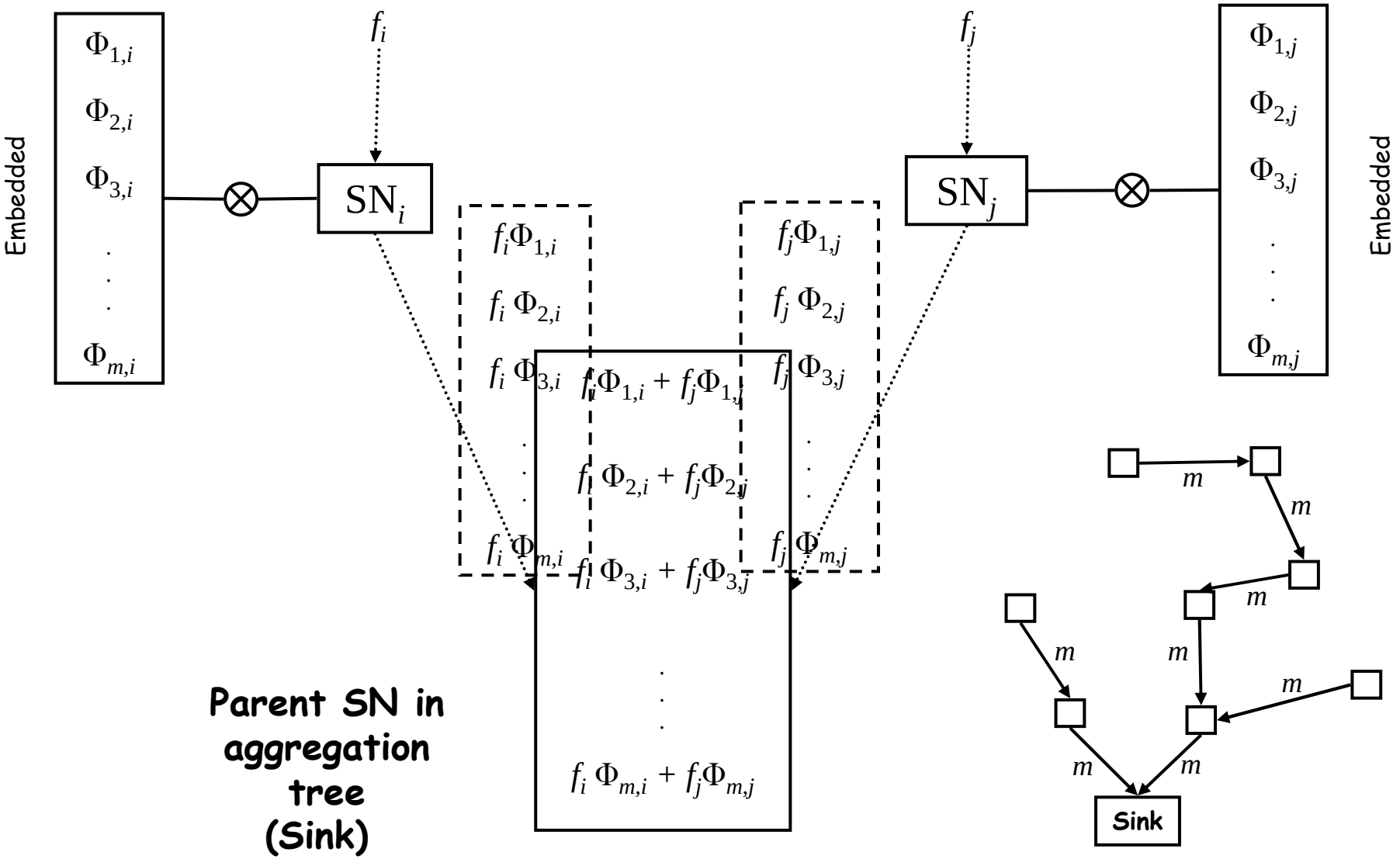
$$\min \|\mathbf{x}\|_{l_1} \quad \text{subject to} \quad y_k = \langle \varphi_k, \mathbf{\Psi x} \rangle, \quad \forall k \in \mathrm{M}$$

Compressive Wireless Sensing (CWS)

- Distributed computation of $\mathbf{y}_{m\times 1} = \mathbf{\Phi}_{m\times N}\,\mathbf{f}_{N\times 1}$
- Here $\mathbf{\Phi}_{m\times N}$ is the measurement matrix composed of rows in $\mathbf{\Phi}$ whose indices are in $\mathrm{M} \subset \{1,\ldots,N\}$



CWS – Building blocks



Outline

- Motivation
- Related Work
- System Model
- Brief Introduction to Compressive Wireless Sensing (CWS)
- Reordering for better Compressibility
  - Improving CWS through reordering
- Performance Evaluation
- Conclusion

Main Idea of Our Approach

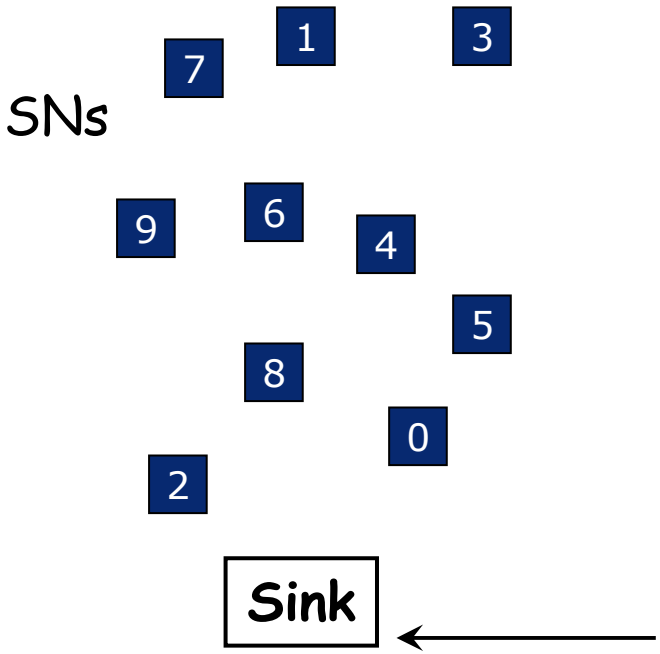
- Is it possible to implement CWS even more efficiently?
- CS main formula, *a second look* :

$$m \geq C \cdot \mu^2(\Phi, \Psi) \cdot S \cdot \log\left(\frac{N}{\delta}\right)$$

- C, μ and N are constant after WSN deployment, and δ depends on our desired accuracy level
- Is it possible to decrease S ?
  - The sparser the Ψ-transform of the signal, the lower parameter S, and accordingly fewer number of CS samples, thus conserving more energy

Signal Reordering Criteria

- The order of signal vector elements in a WSN is *conventional*
- *This order can be changed via updating the sampling matrix* Φ



Signal vector **a**

a_0	SN_0
a_1	SN_1
a_2	SN_2
a_3	SN_3
a_4	SN_4
a_5	SN_5
a_6	SN_6
a_7	SN_7
a_8	SN_8
a_9	SN_9

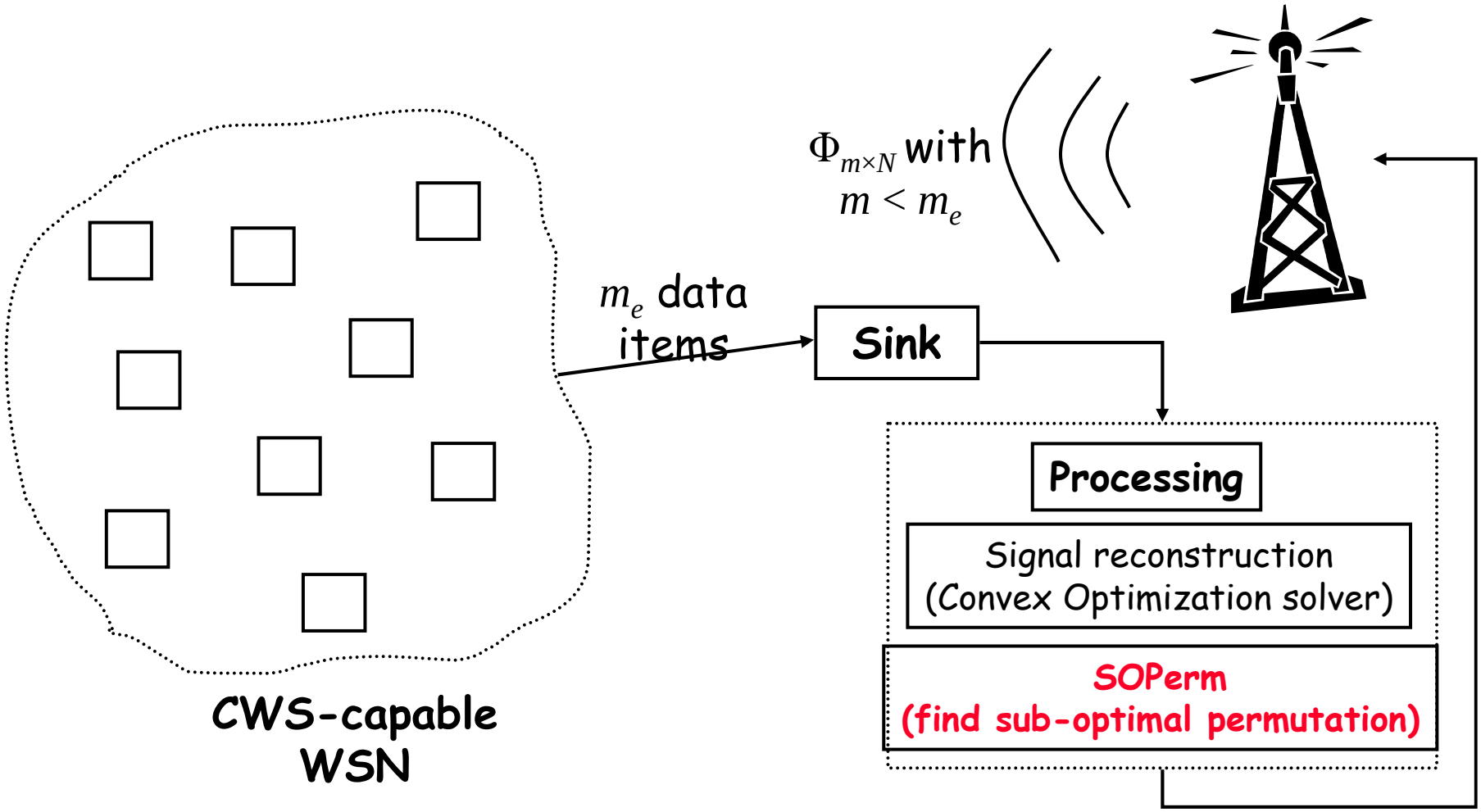
Signal vector **b**

a_0	SN_8
a_1	SN_6
a_2	SN_3
a_3	SN_2
a_4	SN_5
a_5	SN_0
a_6	SN_1
a_7	SN_7
a_8	SN_9
a_9	SN_4

Sampling Matrix Φ and Signal (Re)ordering

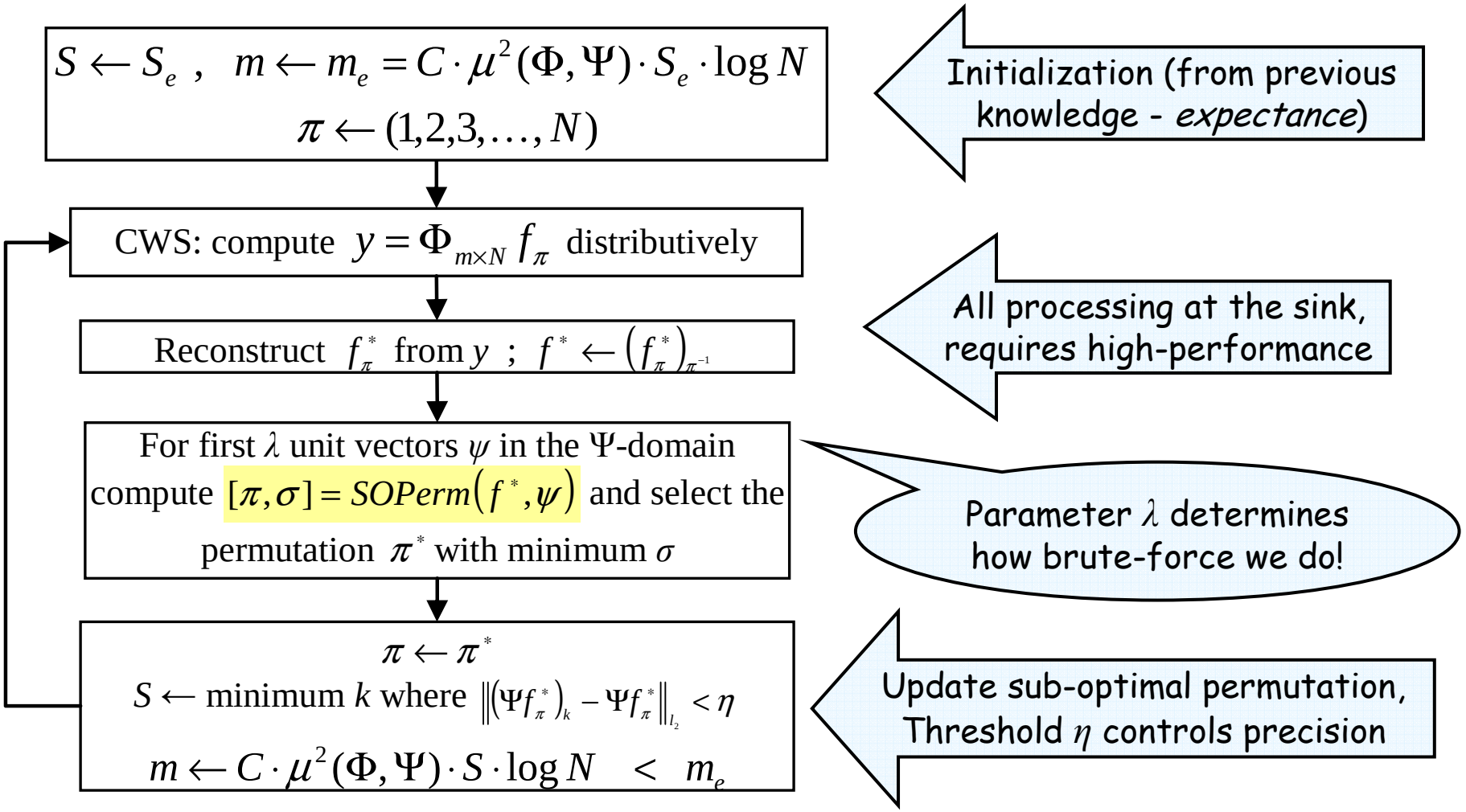
- Deployment
  - Embedding columns of Φ inside the SNs
- Initialization:
  - The initial order is specified in this phase by knowing which sample is multiplied by which column of Φ
- Reordering
  - Realizing a better permutation -> more compressibility
    - Permutation doesn't take place physically, but instead virtually by changing the sampling matrix $\mathbf{\Phi}_{m \times N}$
    - Updated $\mathbf{\Phi}$ is broadcasted over the WSN by the sink (some improvements exist to avoid flooding the entire matrix)

Illustrating Information Flow



From time to time we perform a m_e sampling to reduce aggregated error (depends on dynamic nature of our environment)

Reordering for Better Compressibility



SOPerm: Sub-Optimal Permutation

- Definition: we define the *difference* between two vectors $\mathbf{a}$ and $\mathbf{b}$ as:

$$D_{\mathbf{a},\mathbf{b}} = \sum_{i=1}^{N} \left| a_i - b_i \right|$$

- For each basis vector ψ_i $(1 \le i \le N)$ greedily reorder the signal vector $\mathbf{f}$ so that $D_{\mathbf{f},\psi_i}$ is minimized (more exactly near minimum) – The SOPerm algorithm

- Among all the basis vectors in Ψ-domain, we choose the permutation which results in the minimum difference
  - *SOPerm is run over basis vectors*
  - *Most likely we can find a good permutation among low-frequency basis vectors, i.e. among* $\{\psi_1, \psi_2, \ldots, \psi_n\}$ *where* $n < N$

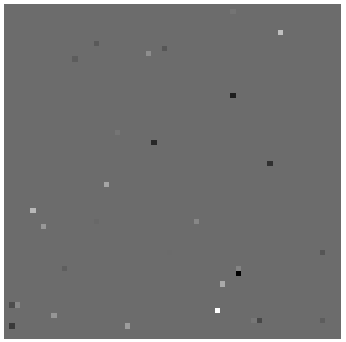
Outline

- Motivation
- Related Work
- System Model
- Brief Introduction to Compressive Wireless Sensing (CWS)
- Reordering for better Compressibility
  - Improving CWS through reordering
- Performance Evaluation
- Conclusion

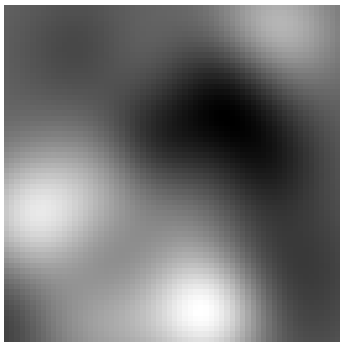
Synthetic Environment Simulation

Implemented in MATLAB

Random points



After Gaussian filter



- Dimensions of the synthetic environment image (pixels)
- Radius of the Gaussian filter in pixels: 1/8 times the dimension of the image

We simulate a simplified WSN by reading random pixels (**N=2000**) from the synthesized operational environment

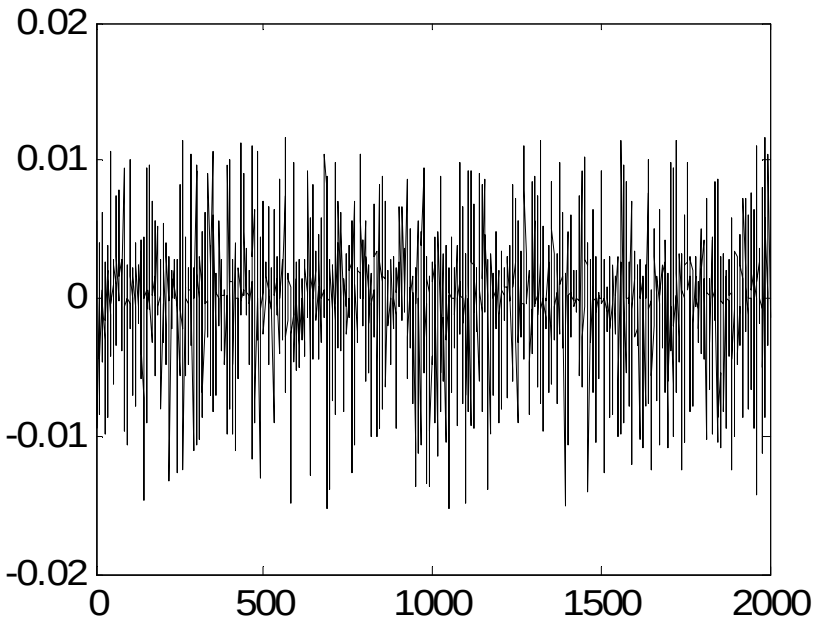
Performance metric for Reordering:

Compressibility = Sparsity of the signal in representation domain: N-S

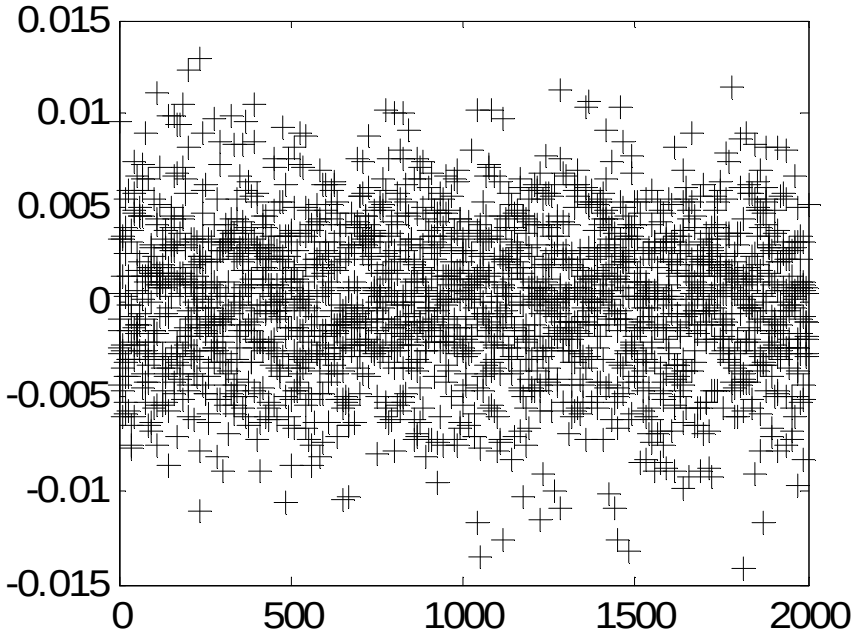
Simulation Results – Before Reordering

X axis has no dimension (vector index)

Y axis depends on physical parameter, in synthetic environment it is a real number between -1 and +1 because of Gaussian filter



Signal vector indexed by SN id's

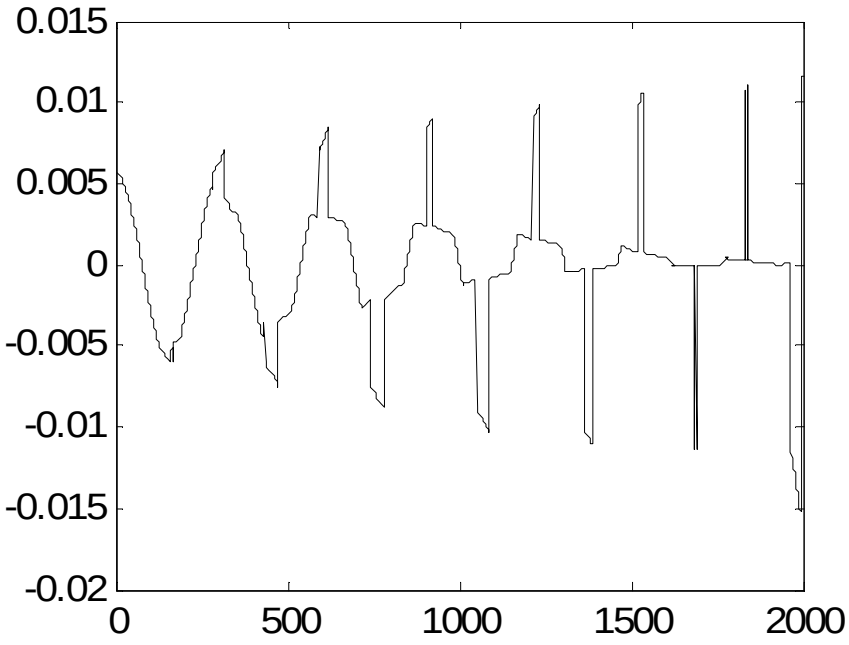


DCT transform of the Signal vector

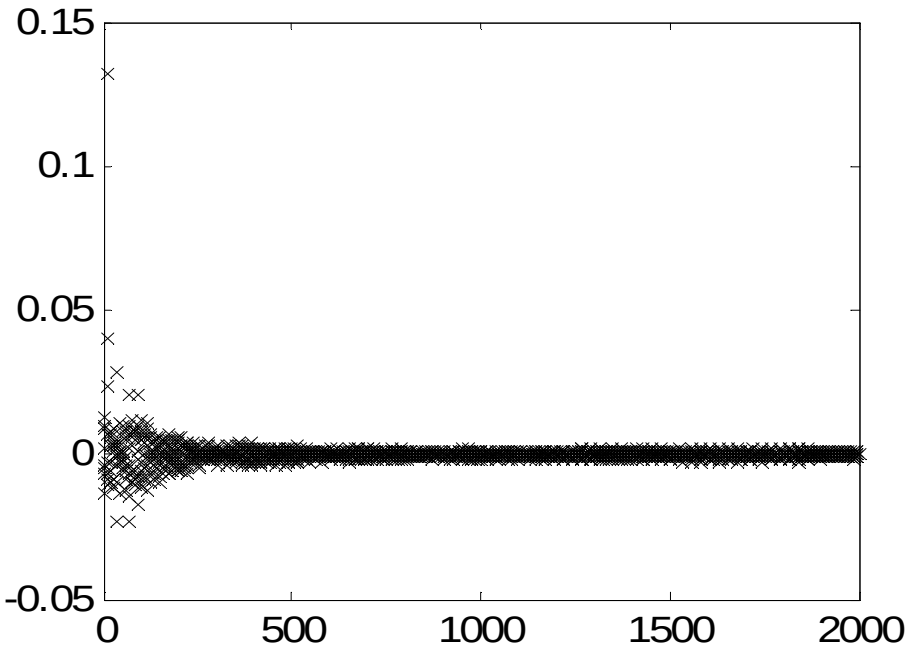
Simulation Results – Impact of Reordering

X axis has no dimension (vector index)

Y axis depends on physical parameter, in synthetic environment it is a real number between -1 and +1 because of Gaussian filter

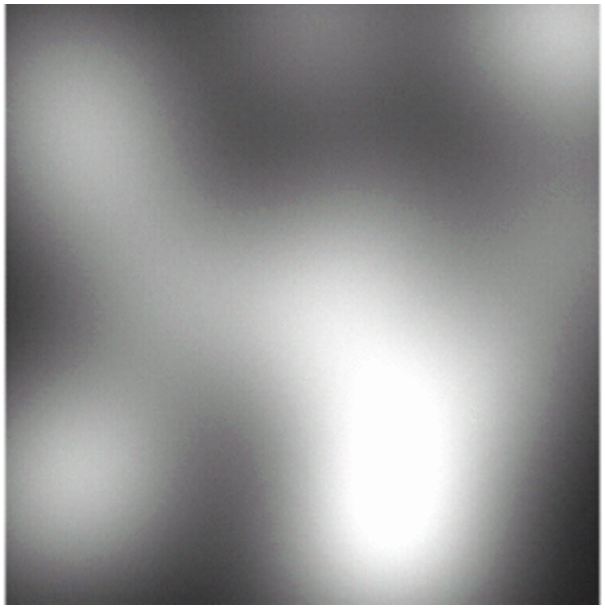


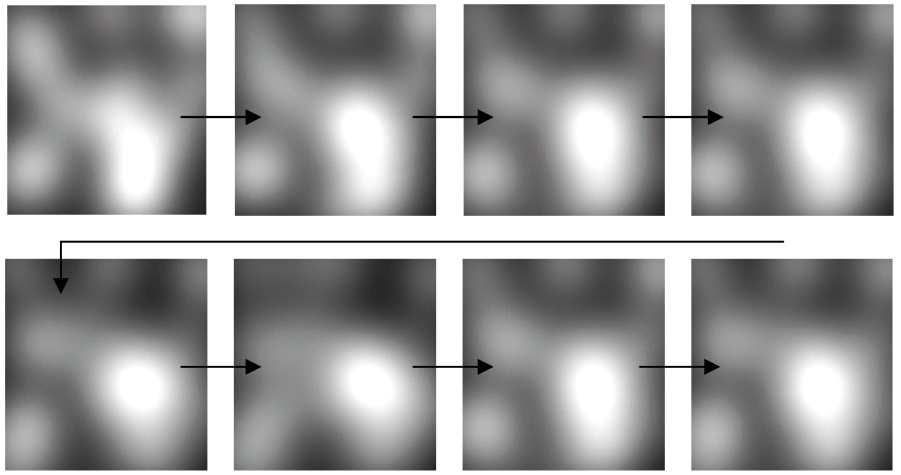
Sub-optimally permuted signal vector



DCT transform of the sub-optimally permuted vector

Simulation of a Dynamic Environment

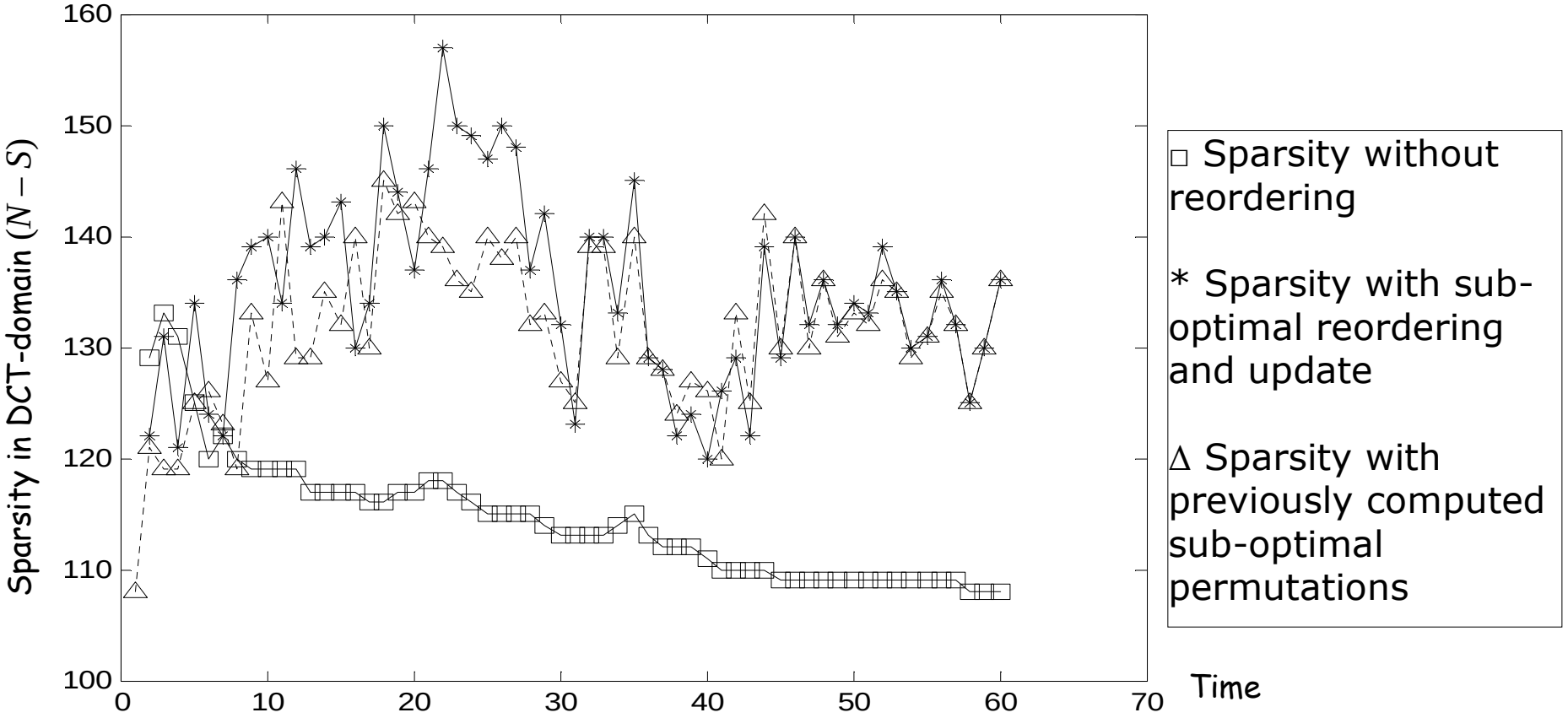




- Dynamic environment synthesized by moving random points while applying Gaussian filter

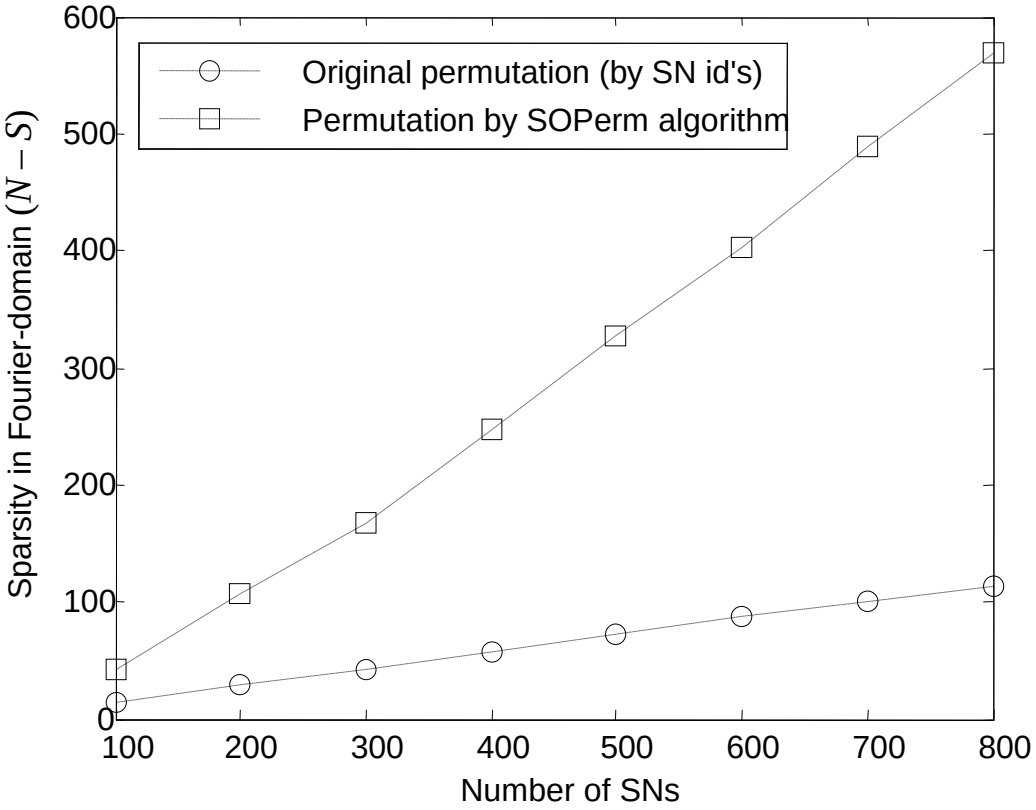
- Pixel step: 5-10

Impact of Dynamic Environments



When the environment doesn't change drastically, a sub-optimal permutation found at a point of time, stands better than the normal permutation for very next time periods

Impact of Number of Nodes



Overall Test is run for many simulated operational environments with different numbers of SNs and averaged over 100 different synthesized environments

Signal vector reorder by SOPerm stands always sparser in the DCT/Fourier domain

- Experiments consolidate our hypothesis
- Our SOPerm algorithm finds a more compressible permutation of discrete signal in a limited time

Conclusions

- C(W)S is an important field with promising results in distributed sampling esp. WSN
- The permutation (order) of WSN signal vector is conventional

→We can view a discrete signal in a more compressible way by changing the order of its samples

→ Provided a greedy approximate solution

- Our approach can be used for CS not only WSN

- Future Work: Generalized adaptive CWS sampling
  - Reordering can be interpreted as a special type of orthogonal linear transform (0-1 conversion matrix)
  - Generalization to improve efficiency of CWS by developing a linear transform that makes Ψ-transform of $\mathbf{f}$ sparsest

Thanks for your attention!

{reza, khelil, suri}@cs.tu-darmstadt.de

DEEDS Group,
Darmstadt University of Technology