# COMPRESSED SENSING SIGNAL RECOVERY

# VIA

# A* ORTHOGONAL MATCHING PURSUIT

Nazım Burak KARAHANOĞLU

TUBITAK - BILGEM

Hakan ERDOĞAN

Sabancı University

# Overview

- Introduction & Motivation
  - Compressed Sensing (CS) Problem
  - Matching Pursuits
  - Single vs. Multi-Path Search
- A*OMP: Best-first Search for Compressed Sensing
  - The Algorithm
  - Path Selection and Extension Mechanisms
  - Reconstruction Performance (1D & 2D)
- Conclusions

# Compressed Sensing (CS) Problem

CS question: Acquire a sparse signal $\mathbf{x}$ of length *N* via *M*<*N* (random) observations

➢Define: $\left.\begin{array}{l} \mathbf{x} : K\text{-sparse vector of length } N \gg K \\ \mathbf{y} : \text{observed vector of length } M \\ \mathbf{\Phi} : \text{observation matrix of size } M \times N \end{array}\right\}$ s. t. $\mathbf{y} = \mathbf{\Phi x}$

CS reconstruction problem: $$\boxed{\arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \ \text{ s.t. } \mathbf{\Phi x} = \mathbf{y}}$$

Broad categorisation of reconstruction approaches

- ➢ Greedy Pursuits
- ➢ Convex Optimization ($l_1$ minimization)
- ➢ Nonconvex Minimization ($l_p$ minimization, *0<p<1*)
- ➢ Bayesian Methods

# Matching Pursuits

Matching pursuits: iteratively build up / refine a sparse solution.

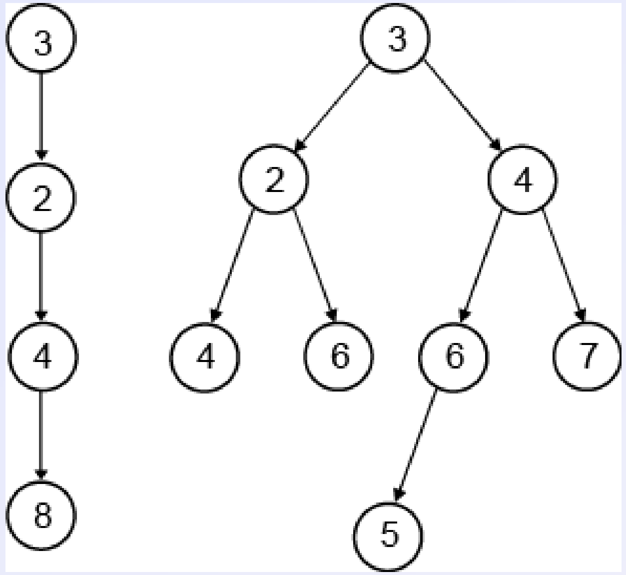
- Matching Pursuit,
- Orthogonal Matching Pursuit,
- Compressive Sampling Matching Pursuit and Subspace Pursuit (SP),
- Regularised OMP, etc.

Orthogonal Matching Pursuit (OMP)

- Identify a non-zero coefficient per iteration:
  - Select dictionary atom having max. inner-product with the residue.
  - Compute orthogonal projection of the residue over the set of selected atoms

# Single vs. Multi-Path

Single path strategies fall into errors especially when *K* increases.




## Multi path strategy:

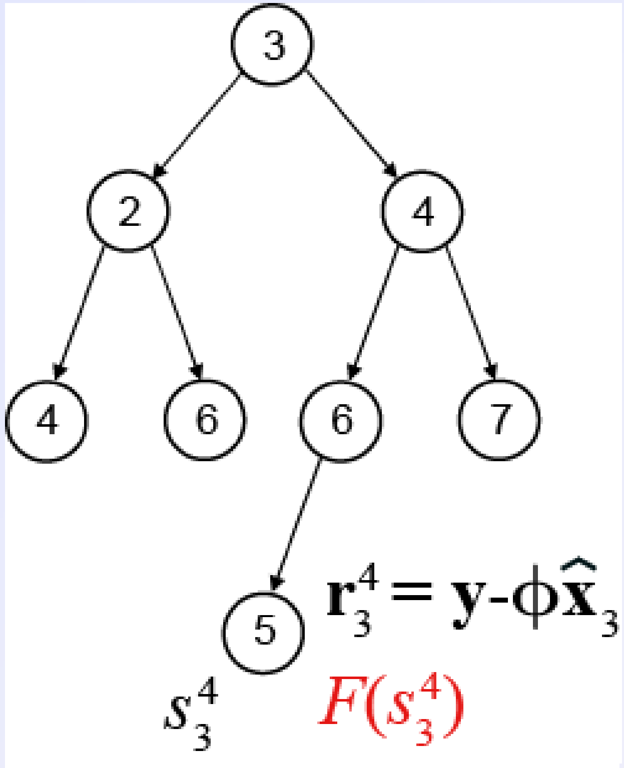
- ➢Consider more than one alternative at each expansion
- ➢Search among a number of dynamically evolving candidates.
- ➢Follow the most promising one, provided an appropriate path selection algorithm.

**A* Orthogonal Matching Pursuit (A*OMP)**

Combine A* search and OMP

# A*OMP: Best-first Search for CS

CS Problem: $\arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \;\; \text{s.t. } \Phi\mathbf{x} = \mathbf{y}$




- Nodes: dictionary elements
- *i*'th path: candidate solution $\hat{\mathbf{x}}_i$
- Each path has a residue.
- Each path is assigned a cost.

A*: build up and dynamically evaluate the search tree.

At each iteration,

- choose best path
- expand the best path

# A* OMP – The Algorithm

A*OMP:

Initialize the tree: *I* initial nodes

Select the best path (with minimum cost)

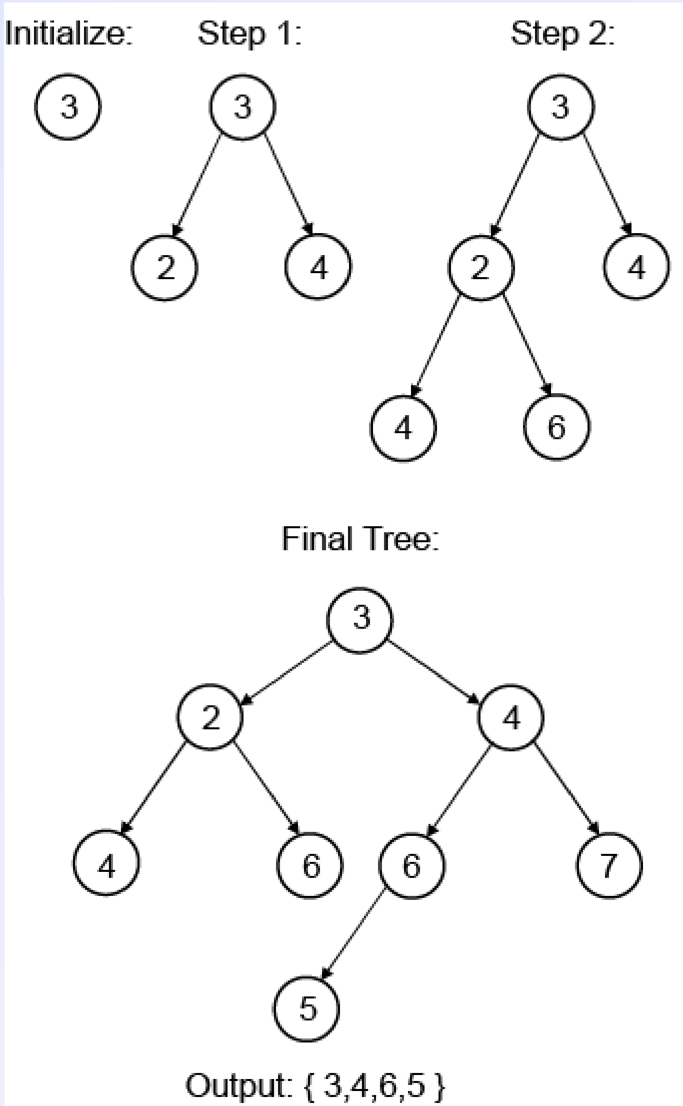
Iterate

Expand best path by its best *B* children (having max. inner-product with residue)

- Update residues (orthogonal proj.)
- Update cost of paths

Select the best path

Terminate when best path has length *K*



# A* OMP – Algorithmic Stages

Three important stages should be defined:

i. Initialization: choose ***I*** nodes with max. inner-product to **y**

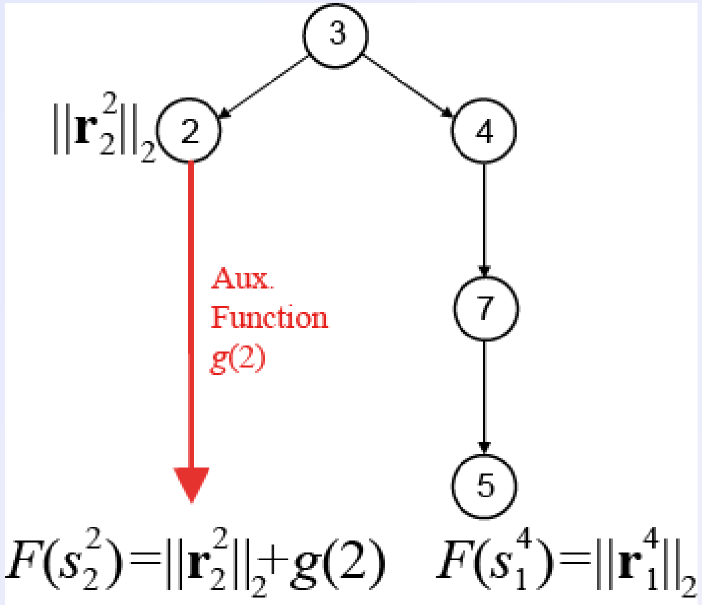
ii. Selection of the best path: how to compare paths with different lengths?

iii. Expansion of the best path: how to avoid too many paths in the tree?

# A* OMP – Best Path Selection

Cost model should compare paths with different lengths:

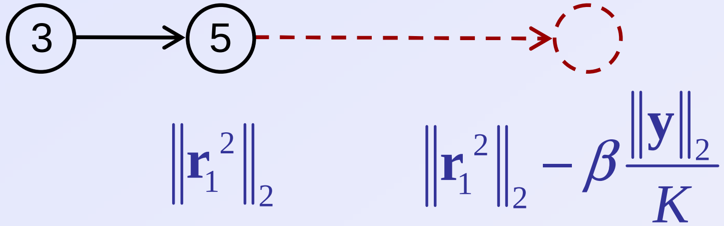
- *Auxiliary function mechanism* of A*
- Based on $\|\mathbf{r}_i\|_2$
- Should reflect how much $\|\mathbf{r}_i\|_2$ would decrease if the path were complete.
- Define cost models that generally (and loosely) hold.



# A* OMP – Cost Models

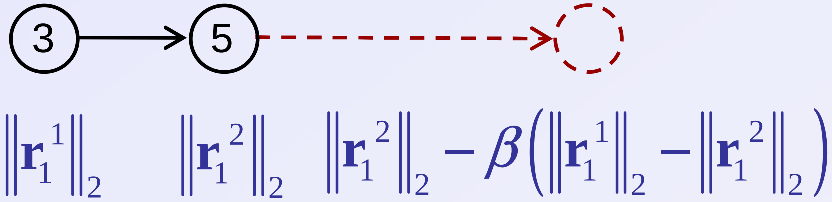
## Additive cost model:

$$F_{add}(s_i^{\,l}) = \left\|\mathbf{r}_i^{\,l}\right\|_2 - \beta \frac{K-l}{K}\left\|\mathbf{y}\right\|_2$$



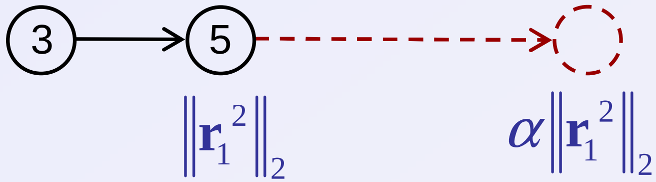

## Adaptive cost model:

$$F_{adap}(s_i^{\,l}) = \left\|\mathbf{r}_i^{\,l}\right\|_2 - \beta (K-l)\left(\left\|\mathbf{r}_i^{\,l-1}\right\|_2 - \left\|\mathbf{r}_i^{\,l}\right\|_2\right)$$




## Multiplicative cost model:

$$F_{mul}(s_i^{\,l}) = \alpha^{K-l}\left\|\mathbf{r}_i^{\,l}\right\|_2$$



# A* OMP – Expansion of the selected path

A* expands all children of the selected branch → too many paths ($\sim N^K$).

i.Extensions per path pruning:

Exploit $K \ll N$ (Many of the children are irrelevant.)

→ Expand only the best $B$ children

ii.Stack Size pruning:

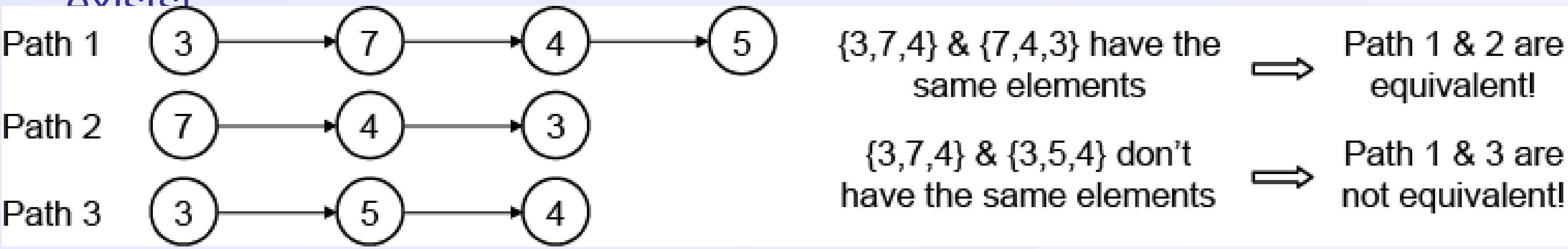
→ Limit max. stored paths to $P$.

→ If number of paths exceeds $P$ remove worst paths.

iii.Equivalent Path Pruning:

Permutations of nodes within a path are equivalent.

→ Add a path to the tree iff no equivalent path already exists!



# A* OMP – A Single Iteration

- $I = 1$
- $P = 4$
- $B = 3$

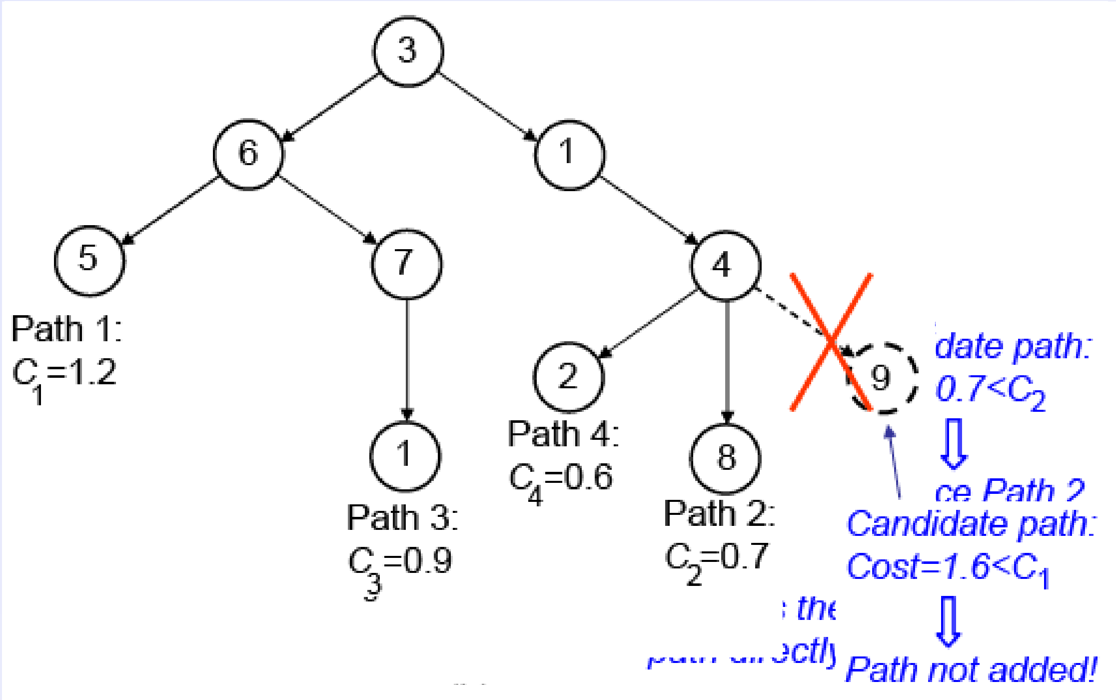
i. Best path: 4

Best extensions: 2,8 and 9

ii. Add node 2

iii. Add node 8, remove the worst path (2).

iv. Ignore node 9

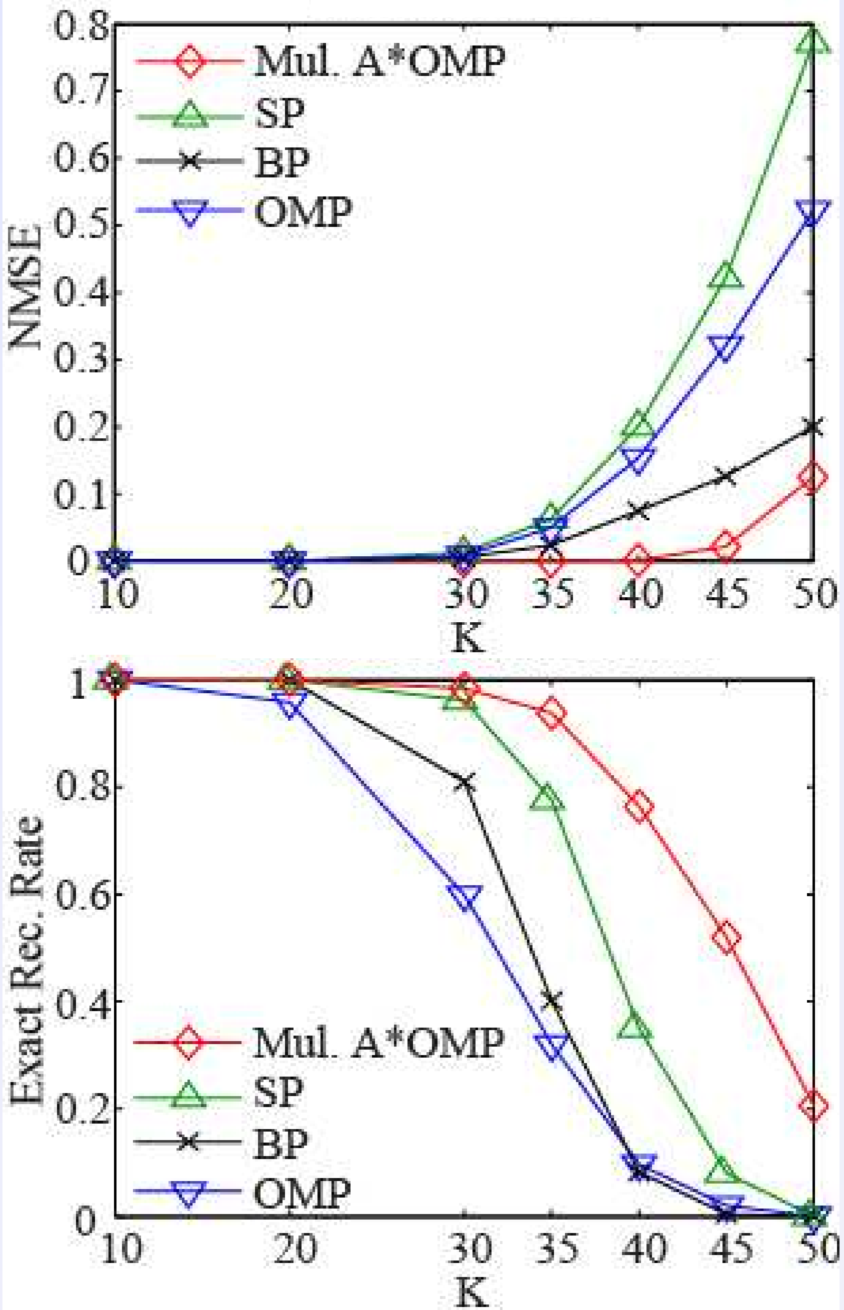


# A*OMP Performance – 1D

Problem: CS reconstruction of synthetical 1D signals

- Nonzero coefs. drawn from standart normal distribution
- $N = 256$
- $M = 100$
- $K = \{10 – 50\}$
- 500 random vectors for each $K$
- Individual random Gaussian observation matrix for each vector

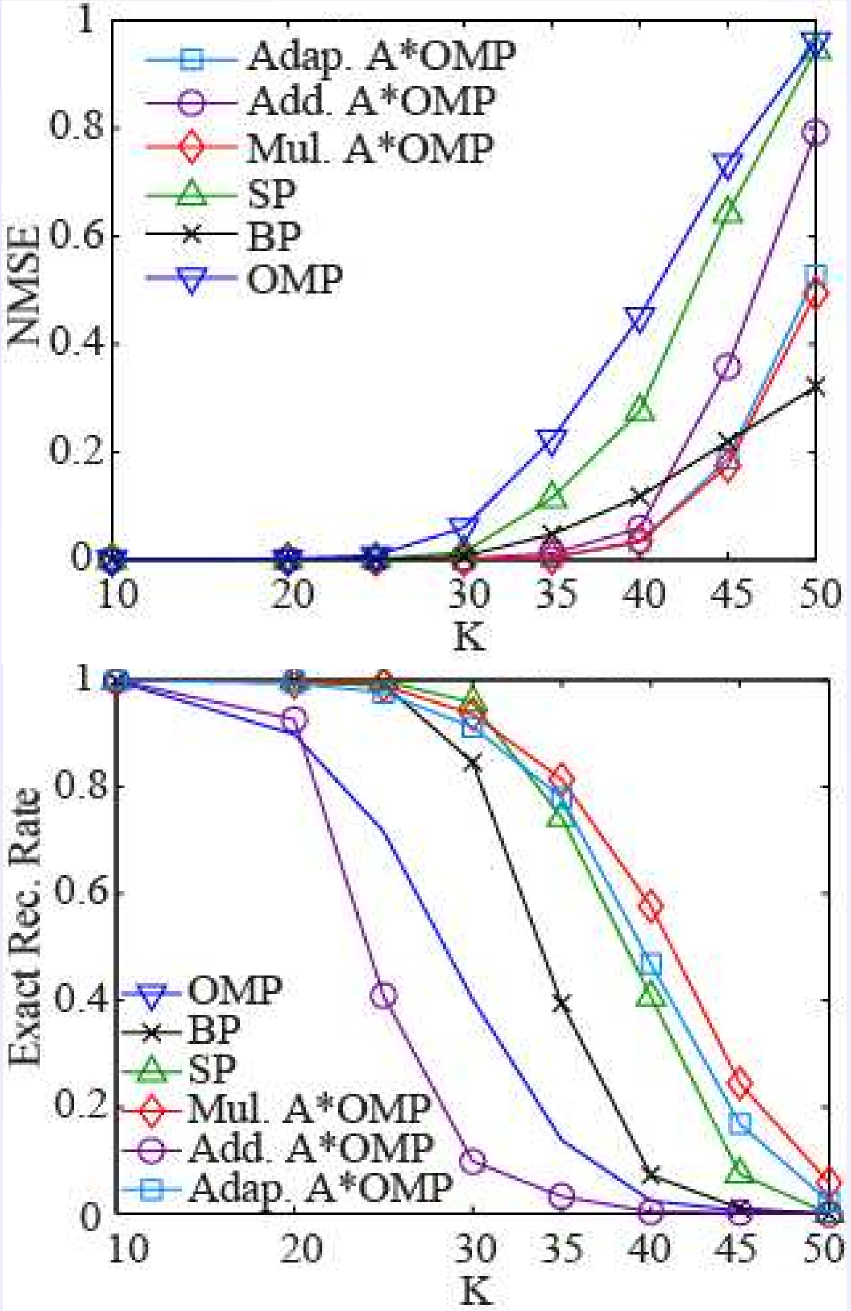
**A*OMP outperforms other algorithms!**



# A*OMP Performance – 1D

- Nonzero coefs. drawn from uniform distribution U[-1,1]
- *N* = 256
- *M* =100
- *K* = {10 – 50}
- 500 random vectors
- Individual random Gaussian observation matrices

**A*OMP outperforms other algorithms!**



# A*OMP Performance - Images

Problem: CS reconstruction of well-known images

- block-processing (8x8 blocks)
- 14-sparse (preprocessed) blocks in Haar Wavelet Basis
- 32 Gaussian observations from each block

Reconstruction Error (peak-SNR)

| | **BP** | **OMP** | **SP** | **Mul-A*OMP** | |
|---|---|---|---|---|---|
| | | | | B = 2 | B = 3 |
| Lena | 27.5 | 23.6 | 21.5 | 30.2 | 33.3 |
| Tracy | 34.6 | 30.8 | 27.9 | 38 | 42.5 |
| Pirate | 25.7 | 21.7 | 19.3 | 27.5 | 30.5 |
| Cameraman | 28.4 | 24.7 | 22.5 | 32.6 | 36.9 |
| Mandril | 22.3 | 18.4 | 16.1 | 24.1 | 26.7 |

# A*OMP Performance - Images

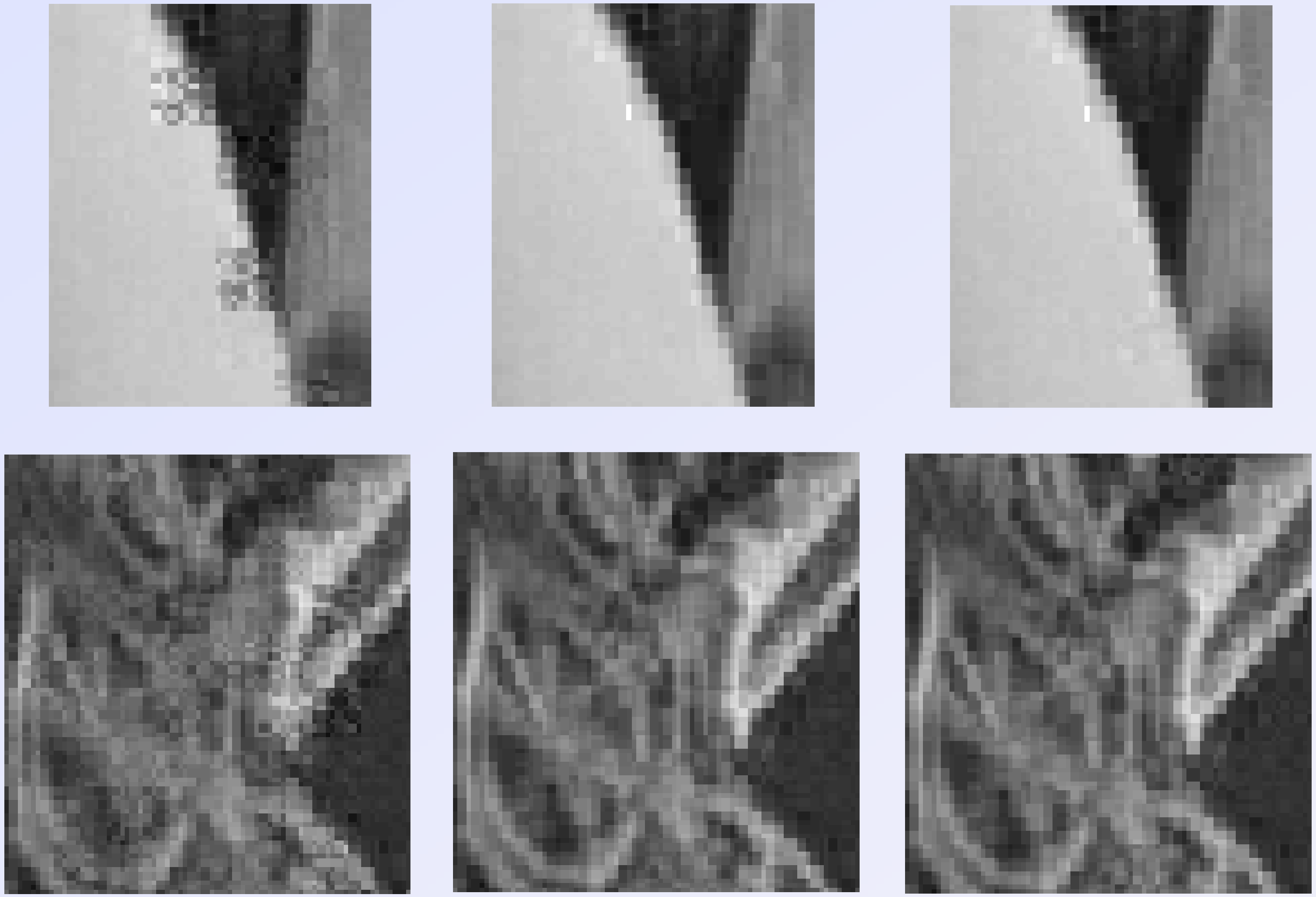
Reconstructed Images

BP

A*OMP

# A*OMP Performance - Images

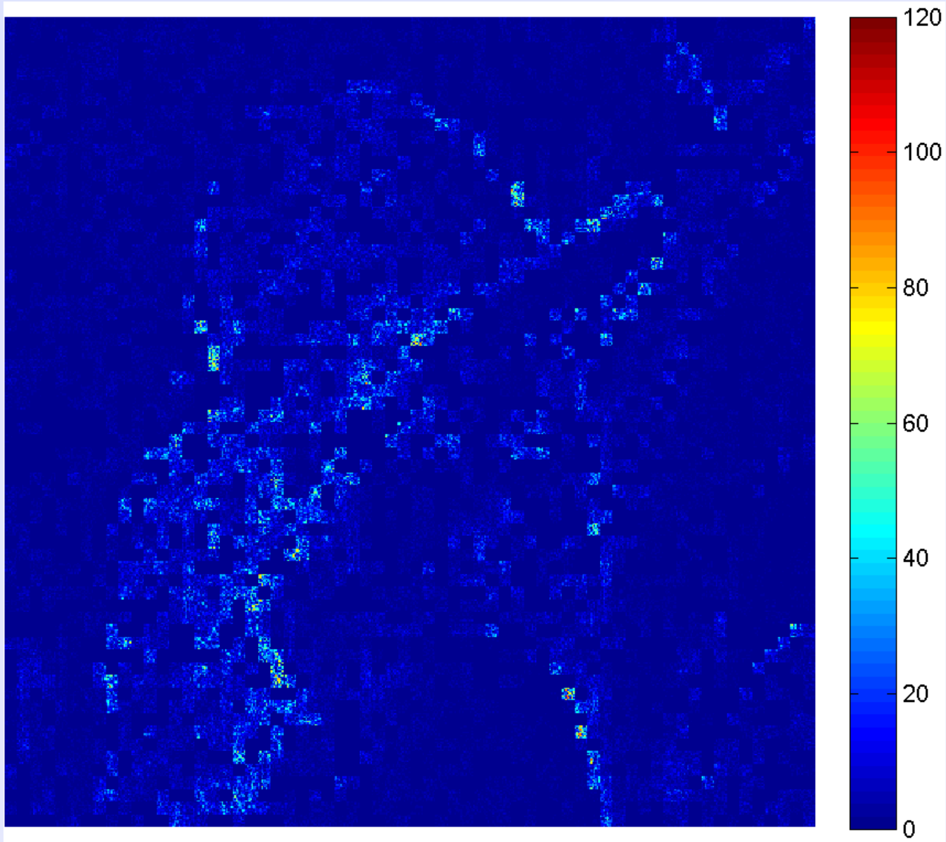
## Reconstruction Details



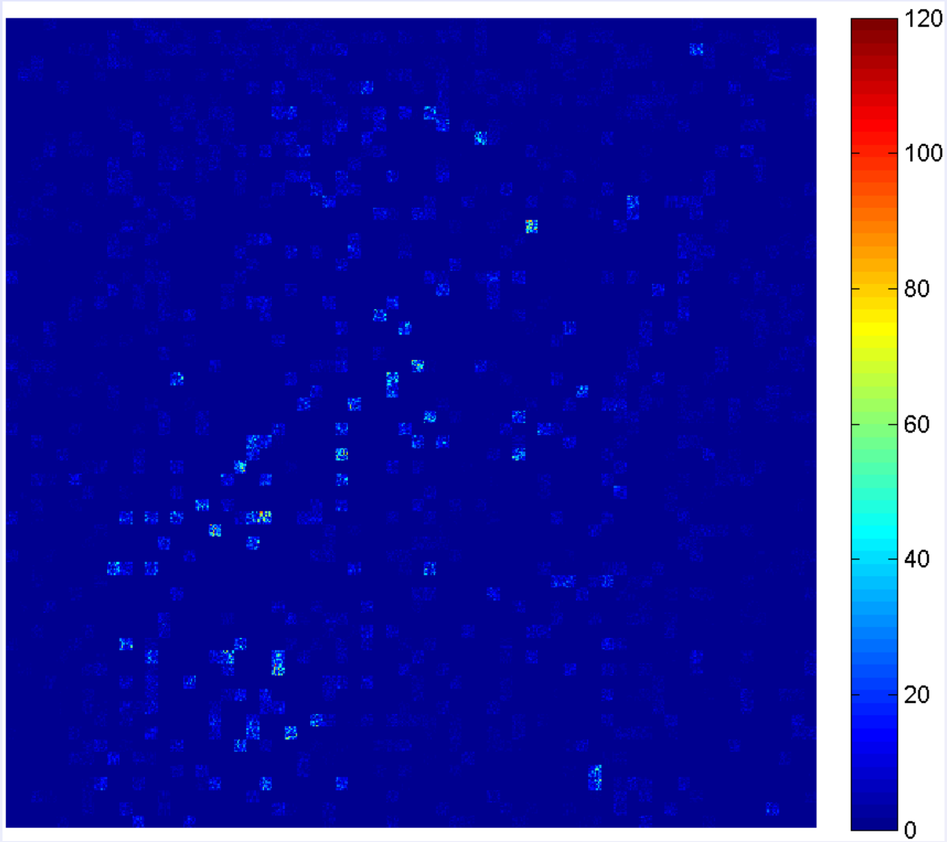
BP | Target Image | A*OMP

# A*OMP Performance - Images

Reconstruction Error per Pixel




BP




A*OMP

# Conclusions

- A*OMP: Multi-path search strategy that combines best-first search and OMP:
  - build up and dynamically evaluate the search tree
  - favor the paths that minimize the cost function
- Two dynamic cost functions (multiplicative and adaptive) in addition to the additive cost function
- Better reconstruction than OMP, SP and BP
- Matlab code available at:

  http://myweb.sabanciuniv.edu/karahanoglu/research/

- Real time implementation is also coming soon...