Sparse Representations and Dictionary Learning Evaluation Framework SMALLbox

Ivan Damnjanovic, Matthew E.P. Davies, Mark D. Plumbley
Centre for Digital Music, Queen Mary University of London
{name.surname}@elec.qmul.ac.uk

Spars'11 workshop, Edinburgh, Scotland, UK, June 27-30, 2011

Overview

1. **Motivation**
2. Design Approach to SMALLbox
3. Implementation
4. Examples
5. Conclusions and Future Work

Motivation (I)

- Sparse representations has become a very active research area.
- Many toolboxes implementing sparse algorithms have become freely available.
- A need exists for a proper testing and benchmarking environment.
- The **SPARCO** framework provides a **large collection** of imaging, signal processing, compressed sensing, and geophysics **sparse reconstruction problems**. It also includes a **large library of operators** that can be used to create new test problems.
- However, using **SPARCO** with other sparse representations toolboxes, such as **SparseLab** is **non-trivial** because of **inconsistencies** in the **APIs** of the toolboxes.
- **SPARCO** considers only **sparse representation** in a known dictionary. It does **not provide a problems for dictionary learning algorithms**.

Motivation (II)

- Design and implement a MATLAB toolbox with three main aims:
  - to enable an easy way of **comparing dictionary learning algorithms**,
  - to provide a unifying API that will enable **interoperability and re-use** of already available **toolboxes for sparse representation and dictionary learning**,
  - to aid the **reproducible research** effort in the sparse signal representations and dictionary learning.

Overview

1. Motivation
2. **Design Approach to SMALLbox**
3. Implementation
4. Examples
5. Conclusions and Future Work

Design Approach (I)

- SMALLbox has been designed to enable an easy exchange of information and comparison of different modules developed through a unified API data structure.
- The structure is made to fulfil two main functionalities:
  - to separate typical sparse signal learning problems into three meaningful units,
  - to provide a seamless connection between the three types of modules and ease of communication of data between them.

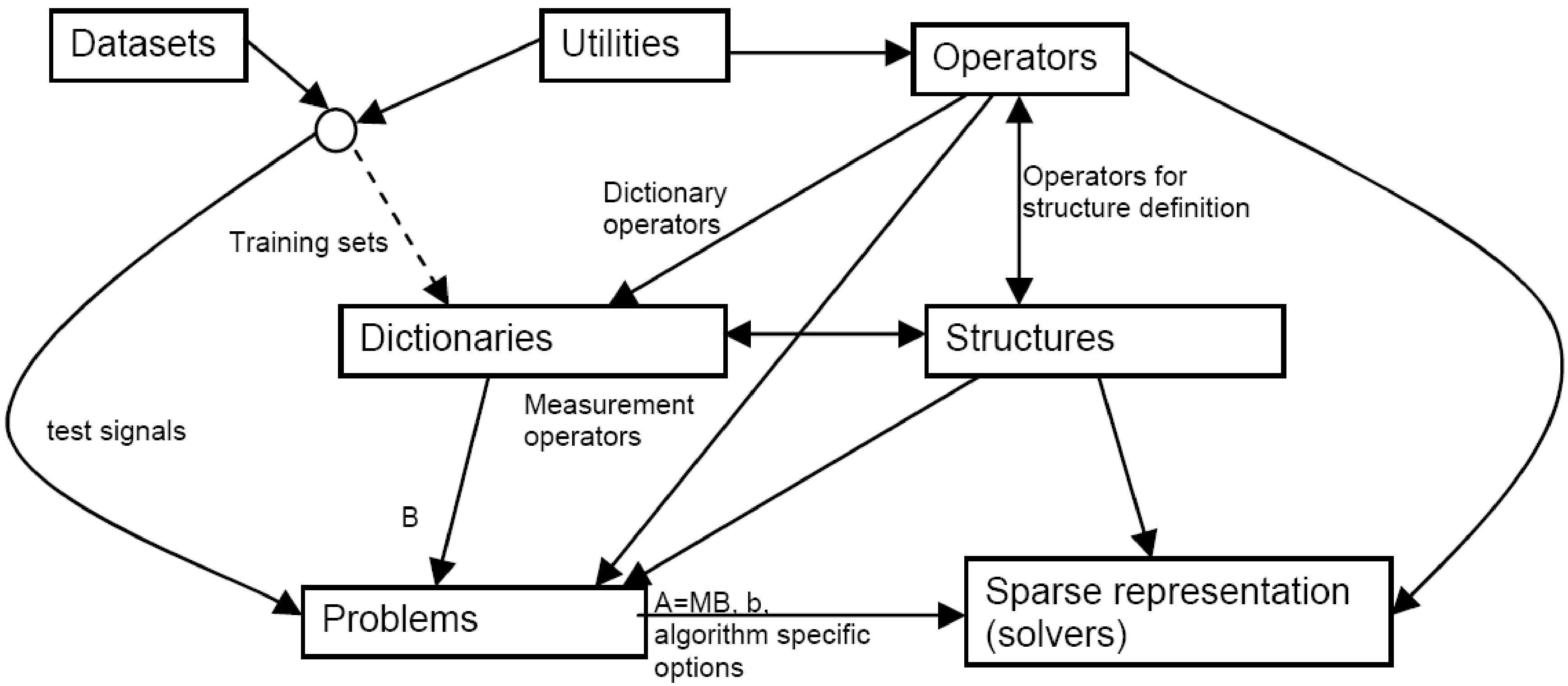
Design Approach (II)

- **Problem specification** - preparing data for learning the structures, representation and reconstruction

- **Dictionary learning** - using a prepared training set to learn the natural structures in the data

- **Sparse representation** - representing the signal with a pre-specified or learned dictionary

Design Approach (III)



Overview

1. Motivation
2. Design Approach to SMALLbox
3. **Implementation**
4. Examples
5. Conclusions and Future Work

Implementation (I)

- The **SMALLbox** (v.1.0) evaluation framework is implemented as an open-source *MATLAB* toolbox and it is downloadable at:

  http://small-project.eu

- For most recent version and contributions you can visit:

  https://code.soundsoftware.ac.uk/projects/smallbox

- documentation includes step-by-step instructions of how to:
  - implement new problems
  - add new sparse-representation algorithms
  - add new dictionary learning algorithms

Implementation (II)

- the **SMALLbox** installation scripts download several third party toolboxes as required:
  - SPARCO (v.1.2) - set of sparse representation problems
  - SparseLab (v.2.1) - set of sparse solvers
  - Sparsify (v.0.4) - set of greedy and hard thresholding algorithms
  - SPGL1 (v.1.7) - large-scale sparse reconstruction solver
  - GPSR (v.6.0) - Gradient projection for sparse reconstruction
  - KSVD-box (v.13) and OMP-box (v.10) - dictionary learning
  - KSVDS-box (v.11) and OMPS-box (v.1) - sparse dictionary learning.
  - SPAMS – Online Dictionary Learning [1]

[1]An API for SPAMS – Online Dictionary Learning is included, but user needs to install the toolbox, due to licensing issues

Implementation (III)

- The SMALLbox provides a “glue” structure to allow algorithms from those toolboxes to be used with a common API. The structure consists of three main sub-structures:
  – ***Problem*** structure - design to be backward compatible with the SPARCO problem structure, so any of SPARCO sparse representation problems or one of the dictionary learning problems provided in SMALLbox can be specified.
  – ***DL*** structure - If the problem is dictionary learning, one or more algorithms can be specified, and tested with specified set of parameters.
  – ***Solver*** structure - can be used to specify any solver provided in 3rd party toolboxes or any provided in SMALLbox to sparsely represent the signal in a dictionary (either defined in the Problem structure or learned in the previous step).

Implementation (IV)

- Implemented in the most recent SMALLbox:
  - **Problems** – Image Denoising, Automatic Music Transcription, Representation of image with patches from another one.
  - **Solvers** – OMP, MP, PCGP
  - **Dictionary Learning** – RLS-DLA
  - **Example scripts**

Overview

1. Motivation
2. Design Approach to SMALLbox
3. Implementation
4. **Examples**
5. Conclusions and Future Work

Image Denoising Example

- **Image Denoising** KSVD vs RLS-DLA

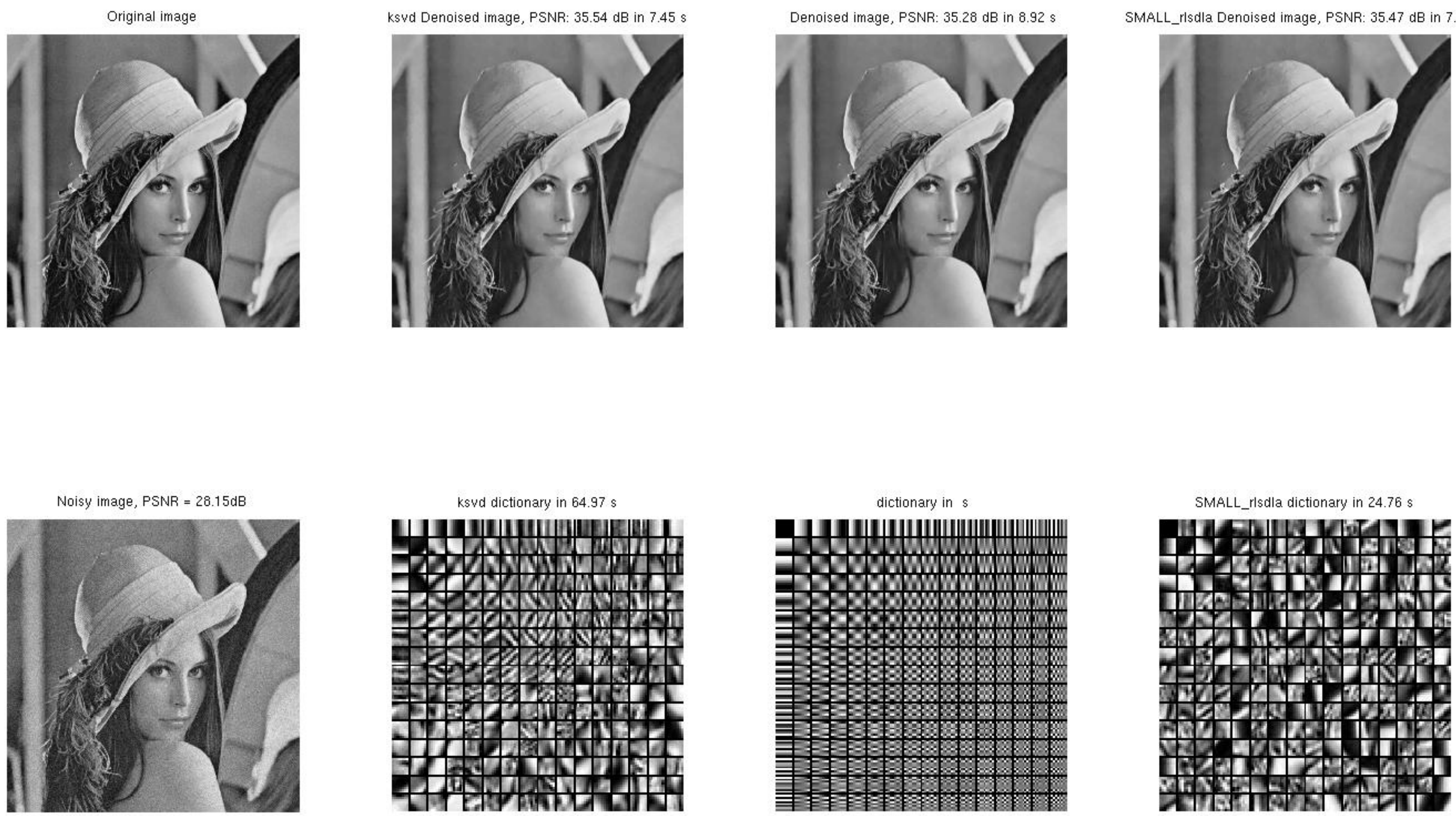


Image Denoising results (Lena)

sigma	10	20	25	50	100
Ksvd Psnr	35.55	32.42	31.27	27.84	24.39
Rlsdla Psnr	35.47	32.43	31.31	27.96	24.41
Ksvd Time	72.12	46.23	42.25	37.57	31.73
Rlsdla Time	28.36	23.6	22.39	20.34	8.58

Overview

1. Motivation
2. Design Approach to SMALLbox
3. Implementation
4. Examples
5. **Conclusions and Future Work**

Conclusions

- SMALLbox is an Evaluation Framework that enables easy prototyping, testing and benchmarking of sparse representation and dictionary learning algorithms.
- Achieved through a set of test problems and an easy evaluation against state-of-the-art algorithms.
- For instructions how to download the SMALLbox:

  http://small-project.eu/

  or

  https://code.soundsoftware.ac.uk/projects/smallbox

- As a part of the EU FET SMALL project, more problems, solvers and dictionary learning techniques that are developed will be included in SMALLbox as the project proceeds.

Acknowledgements

- This research is supported by EU FET-Open ProjectFP7-ICT-225913 “SMALL”, and ESPRC Platform Grant EP/045235/1.
- The authors would also like to thank to all researchers working on the SMALL project for fruitful discussion and help in developing SMALLbox.

http://small-project.eu/

Thank you!

Examples (add)

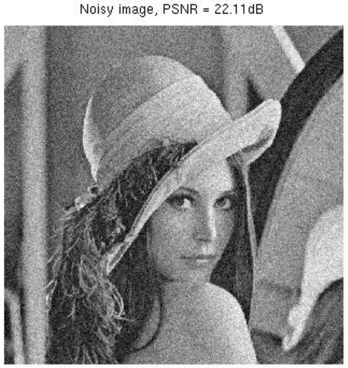
- **Image Denoising** KSVD vs Sparse-KSVD comparison

Original image

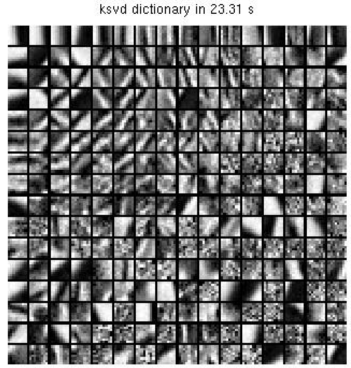
ksvd Denoised image, PSNR: 32.40 dB in 9.73 s

ksvds Denoised image, PSNR: 32.31 dB in 3.57 s

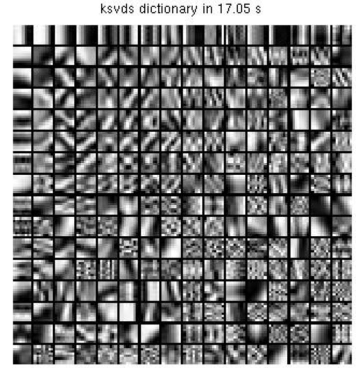
Noisy image, PSNR = 22.11dB



ksvd dictionary in 23.31 s



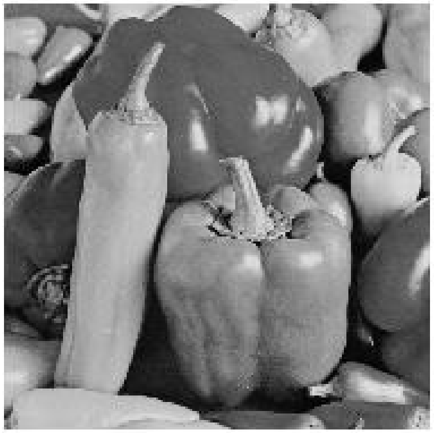
ksvds dictionary in 17.05 s



Examples (add)

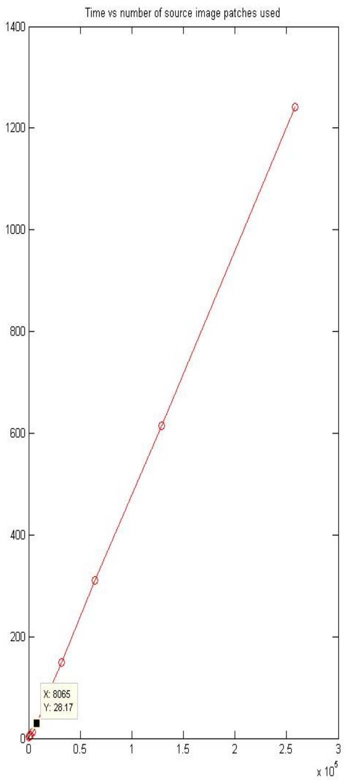
Pierre_Villars_Example.m

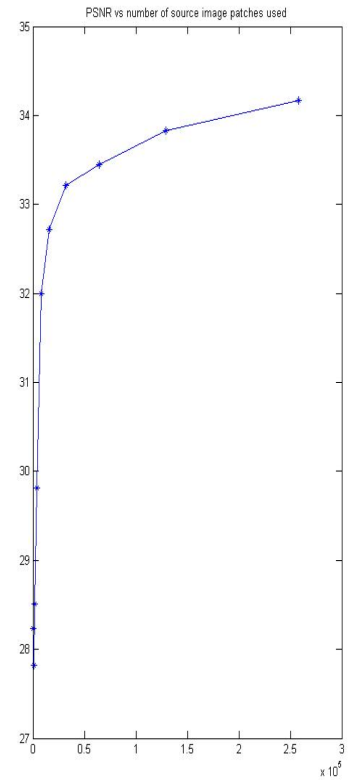
- How to represent an image using sliding patches from the other one?
- How many source patches is needed?
- Does it matter what is in the source image?
- SMALL_MP is used to find three most correlated patches from the source image.
- Example was used to represent Peppers with Barbara and vice-versa



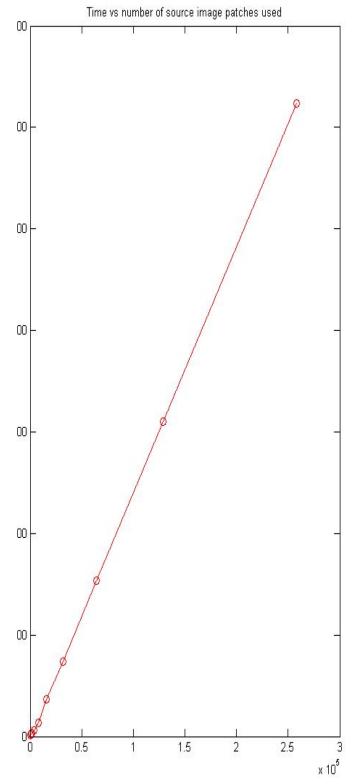
Examples (add)

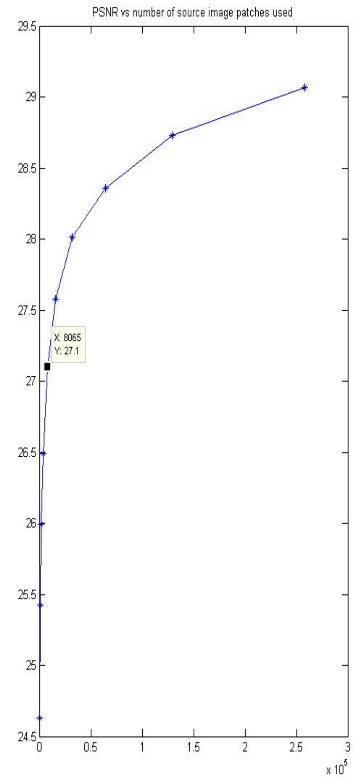
Peppers from Barbara





Barbara from Peppers





•Dictionary of 8065 equidistant patches from Barbara will give PSNR of 32dB in 28.17s

•Other way around, representing Barbara with all patches (258 10^3 patches) from Peppers will give only 29.07 dB of PSNR in more then 20 minutes.

Examples (add)

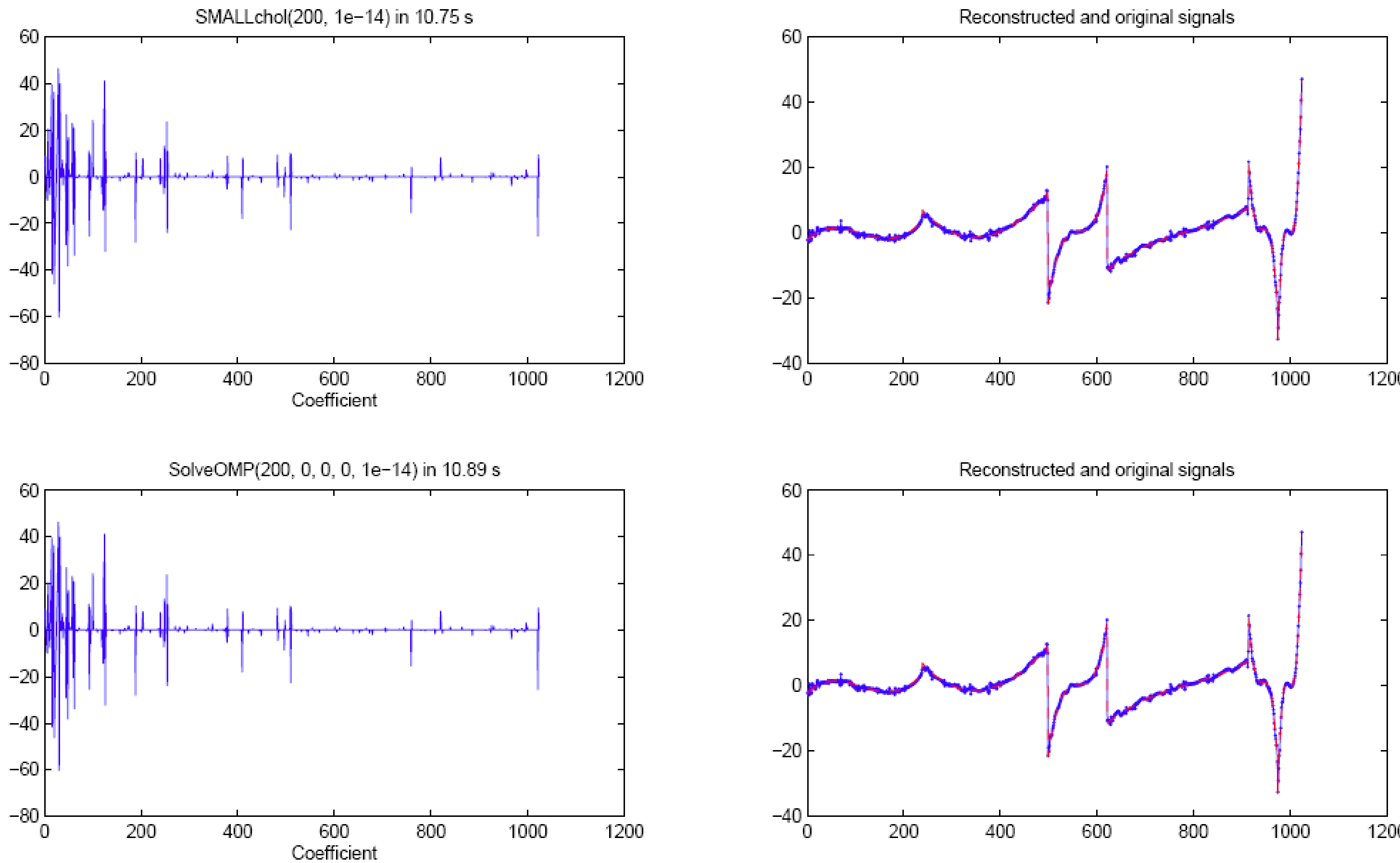
- **small_solver_test.m**
  - generate SPARCO problem 6 (sparse representation of b – a piecewise cubic polynomial signal, in B – a Daubechies basis with M – a Gaussian ensemble measurement matrix),
  - test solvers on the problem (SMALL_chol, Solve_OMP),
  - show computational time and plot solutions and reconstructed signals against the original.

Examples (add)

```
% Generate SPARCO problem
SMALL.Problem = generateProblem(6, 'P', 6, 'm', 270,'n',1024, 'show');

i=1;

% SMALL OMP with Cholesky update test
SMALL.solver(i)=SMALL_init_solver;
SMALL.solver(i).toolbox='SMALL';
SMALL.solver(i).name='SMALL_chol';
SMALL.solver(i).param='200, 1e-14';

SMALL.solver(i)=SMALL_solve(SMALL.Problem, SMALL.solver(i));

i=i+1;
% SolveOMP from SparseLab test
SMALL.solver(i)=SMALL_init_solver;
SMALL.solver(i).toolbox='SparseLab';
SMALL.solver(i).name='SolveOMP';
SMALL.solver(i).param='200, 0, 0, 0, 1e-14';

SMALL.solver(i)=SMALL_solve(SMALL.Problem, SMALL.solver(i));

%% Plot results
SMALL_plot(SMALL);
```

Examples (add)

- **small_solver_test.m**



Examples (add)

•Image Denoising KSVD vs SPAMS (Mairal 2009) comparison – Time and PSNR for different numbers of patches used for training