Learning Sparse Representations for Image and Signal Modeling

PhD Course, DEIB 2023

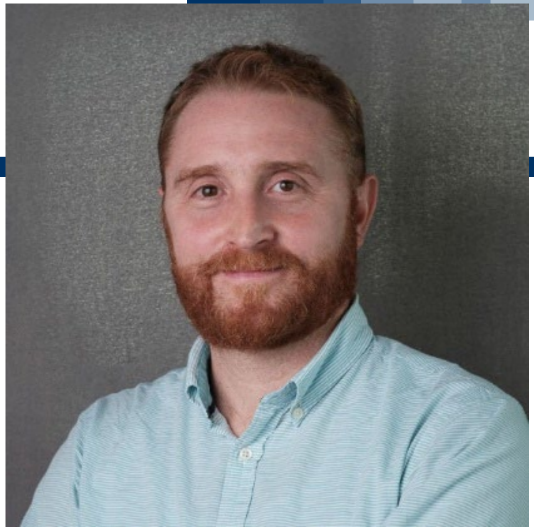
Giacomo Boracchi

April 12, 2023

giacomo.boracchi@polimi.it

https://boracchi.faculty.polimi.it

Giacomo Boracchi



- Mathematician (Università Statale degli Studi di Milano 2004),
- PhD in Information Technology (DEIB, Politecnico di Milano 2008)
- Associate Professor since 2019 at DEIB (Computer Science), Polimi

- Research Interests are mathematical and statistical methods for:
- Image / Signal analysis and processing
- Unsupervised learning, change / anomaly detection

The teaching materials, the exam

Teaching Materials

The course builds upon a previous class held by Dr. Brendt Wohlberg from Los Alamos National Laboratory, (NM, USA) in 2013

- An hard copy of his notes will be provided

Before/after each class I will upload:

- Slides used during lectures
- Codes to be used in the “Computer lab” part of the lectures
- Assignments

Please, check the website

https://boracchi.faculty.polimi.it/teaching/LearningSparse.htm

Teaching

Most of the classes will be given on the backboard,

- please take your own notes, there is not a perfect match between this course and notes from Brendt Wohlberg

During classes I will provide Python / Matlab codes where you will be asked to add few lines implementing the presented algorithms

- Please, try it yourself! This is necessary to understand the presented algorithms

After each class I will provide few assignments (homeworks)

- It is very important to solve these assignments yourself, these often anticipate examples/materials that will be used in the next classes.

Final Assessment

Course Assessment

Final assessment will be based on:

- class participation
- a final discussion of the assignments.

Thus, participate actively to our classes: answer/ask questions, interact with others (including myself) during lab exercises!

- I really appreciate to receive a constant feedback from you

Projects are available upon requests for those students that are particularly interested in the course materials or that want to improve their grade.

Sparsity

From “**Sparse Modeling for Image and Vision Processing**”

J. Mairal, F.Bach, J.Ponce

Now Publisher 2012

Sparsity and Parsimony

The principle of sparsity or "parsimony" consists in *representing some phenomenon with as few variable as possible*

Stretch back to philosopher William Ockham in 14th Century

Wrinch and Jeffreys [1921] relate simplicity to parsimony:

The ***existence of simple laws*** *is, then, apparently, to be regarded as* ***a quality of nature****; and accordingly we may infer that it is justifiable to* ***prefer a simple law to a more complex one that fits our observations slightly better.***

Simplicity <-> number of learning parameters

Sparsity in Statistics

Statistics: simple models are preferred.

Sparsity is used to **prevent overfitting and improve interpretability of learned models.**

In model fitting, the number of parameters is typically used as a criterion to perform model selection.

See Bayes Information Criterion (BIC), Akaike Information Criterion (AIC), …., Lasso.

Signal Processing: similar concepts but different terminology. **Vectors corresponds to signals** and **data modeling is crucial** for performing various operations such as **restoration, compression, solving inverse problems.**

Signals are approximated by sparse linear combinations of **prototypes** (basis elements / atoms of a dictionary), resulting in simpler and compact model.

Best subset selection <-> computing the sparse representation of a signal w.r.t. a give basis/dictionary

Neuroscience: Olshausen and Field [1996], learning the from a training set of data dictionaries yielding sparse representations

Denoising via Thresholding in Transform Domain

Denoising by Thresholding

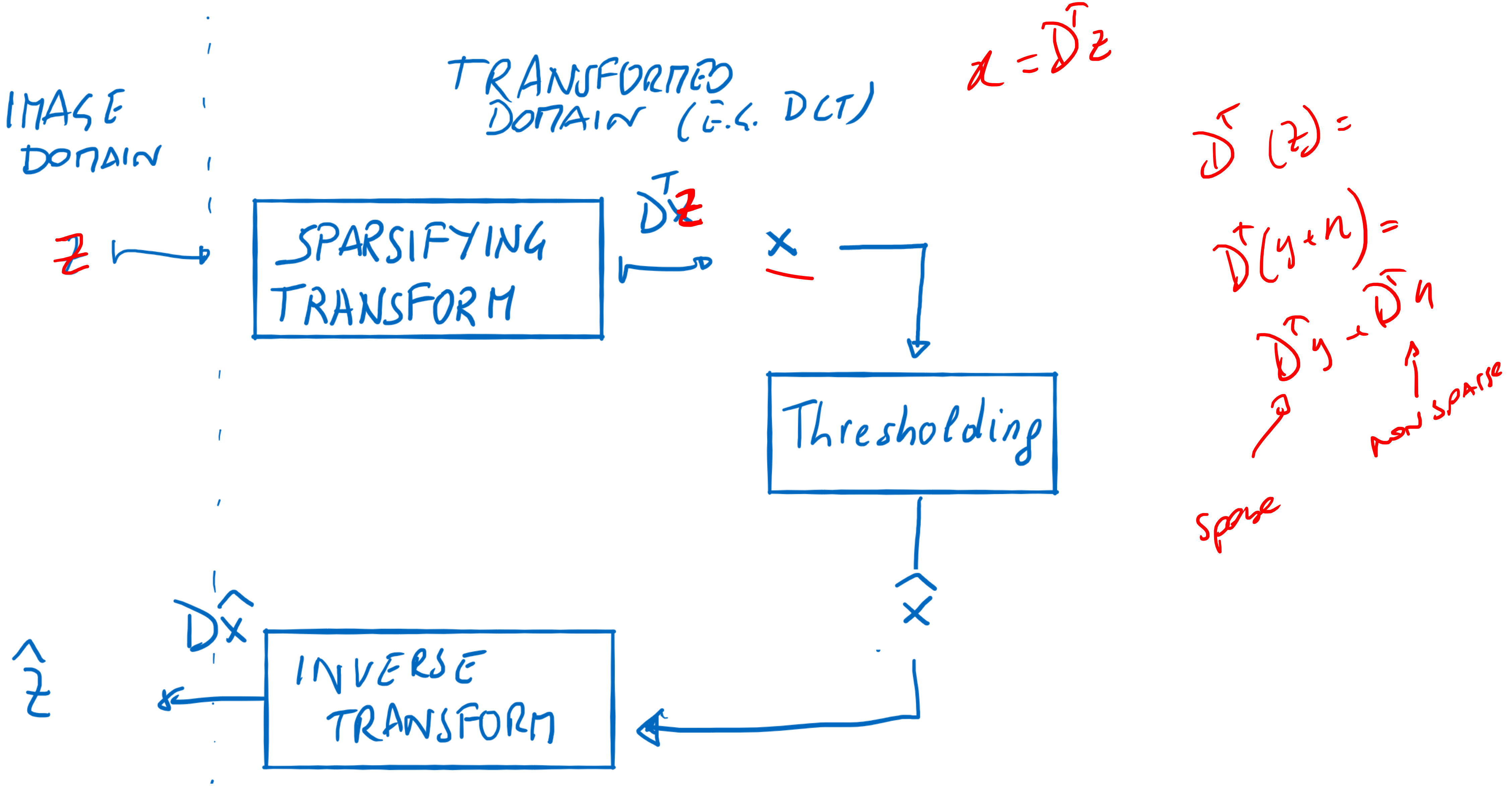
The underlying assumption is that a clean signal admits **a sparse representation w.r.t.** a suitable basis $\{\boldsymbol{e_i}\}_{i=1,\dots,d}$ (or set of generators)

Noise, being unstructred, spreads in all the coefficients

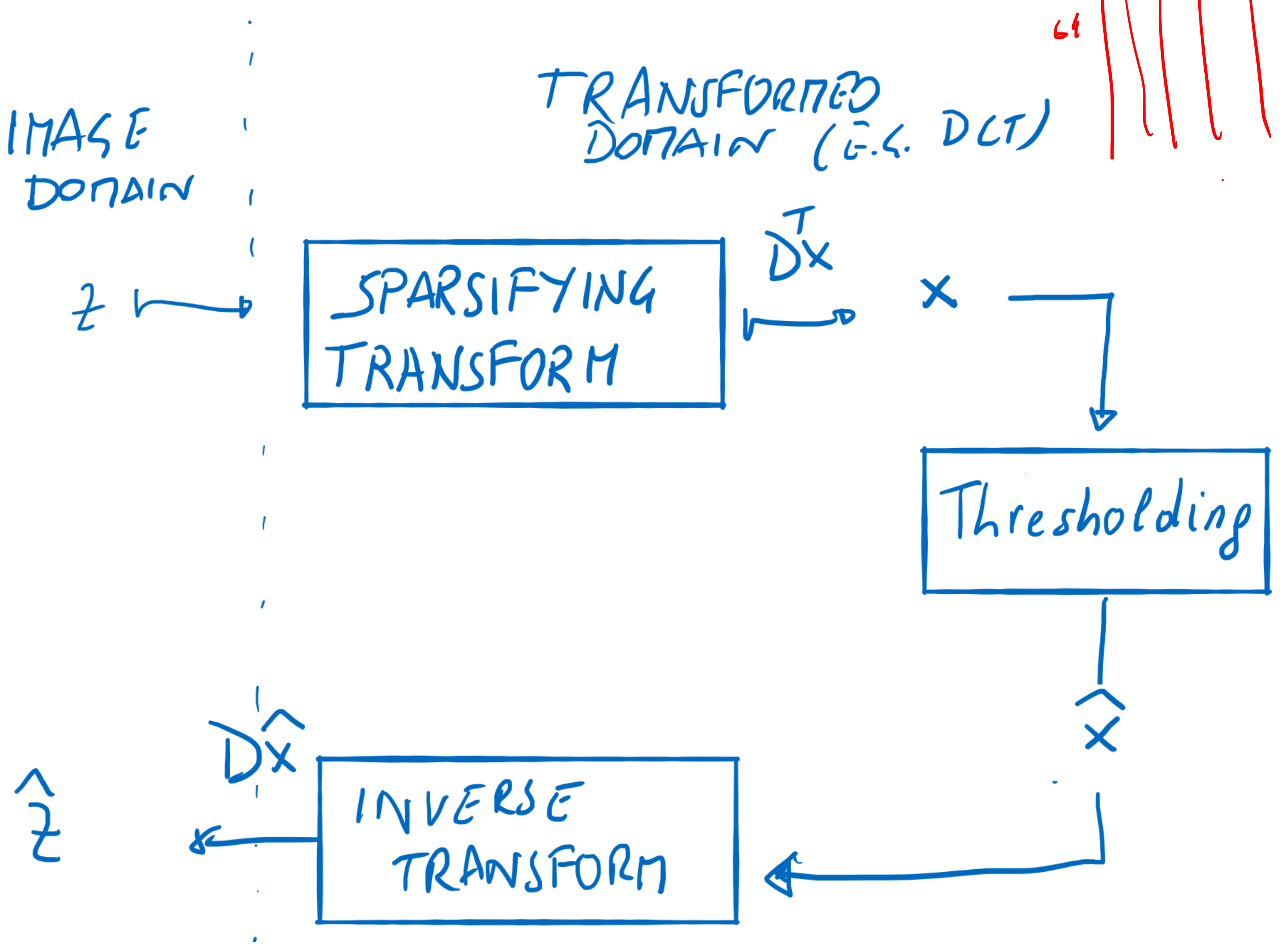
A general denoising appraoch consists in

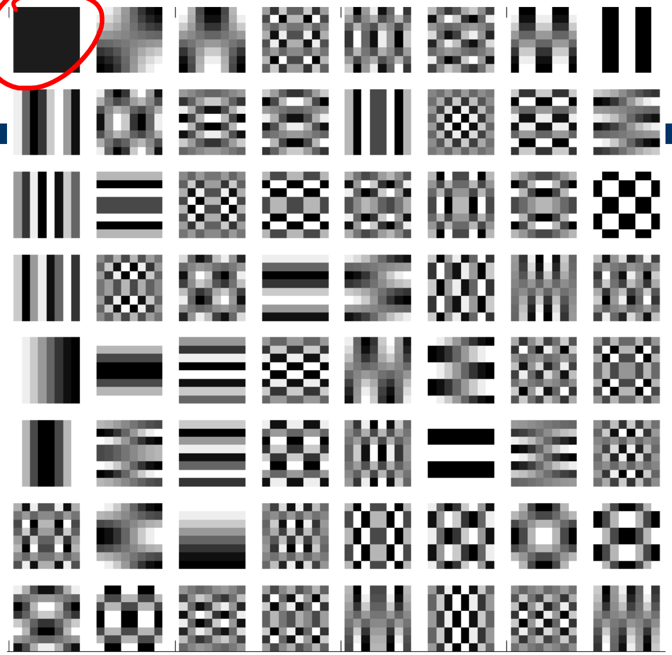
- Cropping the image patch-wise
- Transform each patch according to a sparsitying transformation
- Perform Thresholding
- Invert Transformation

Sparsity Promoting Denoising



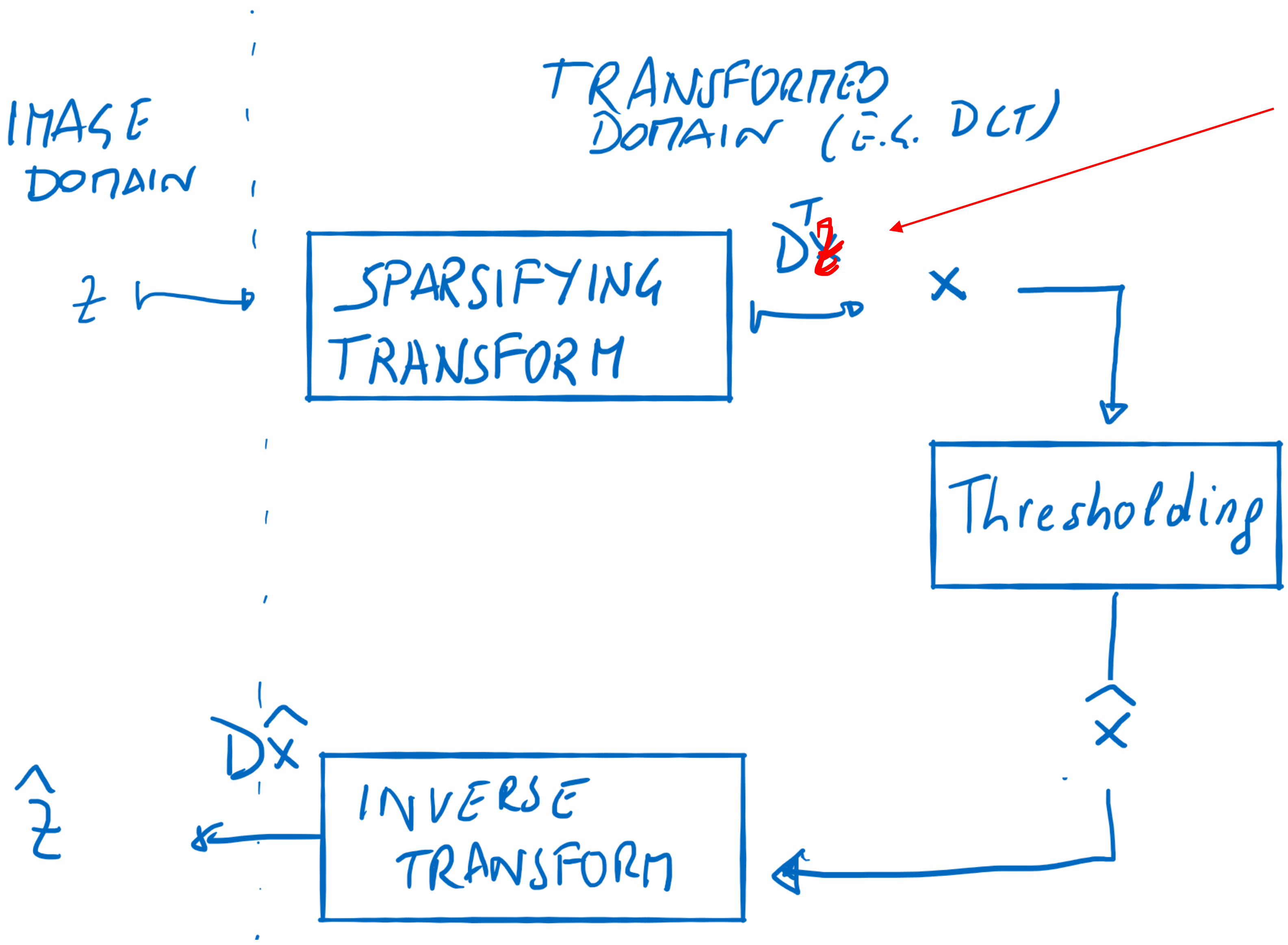
Sparsity Promoting Denoising





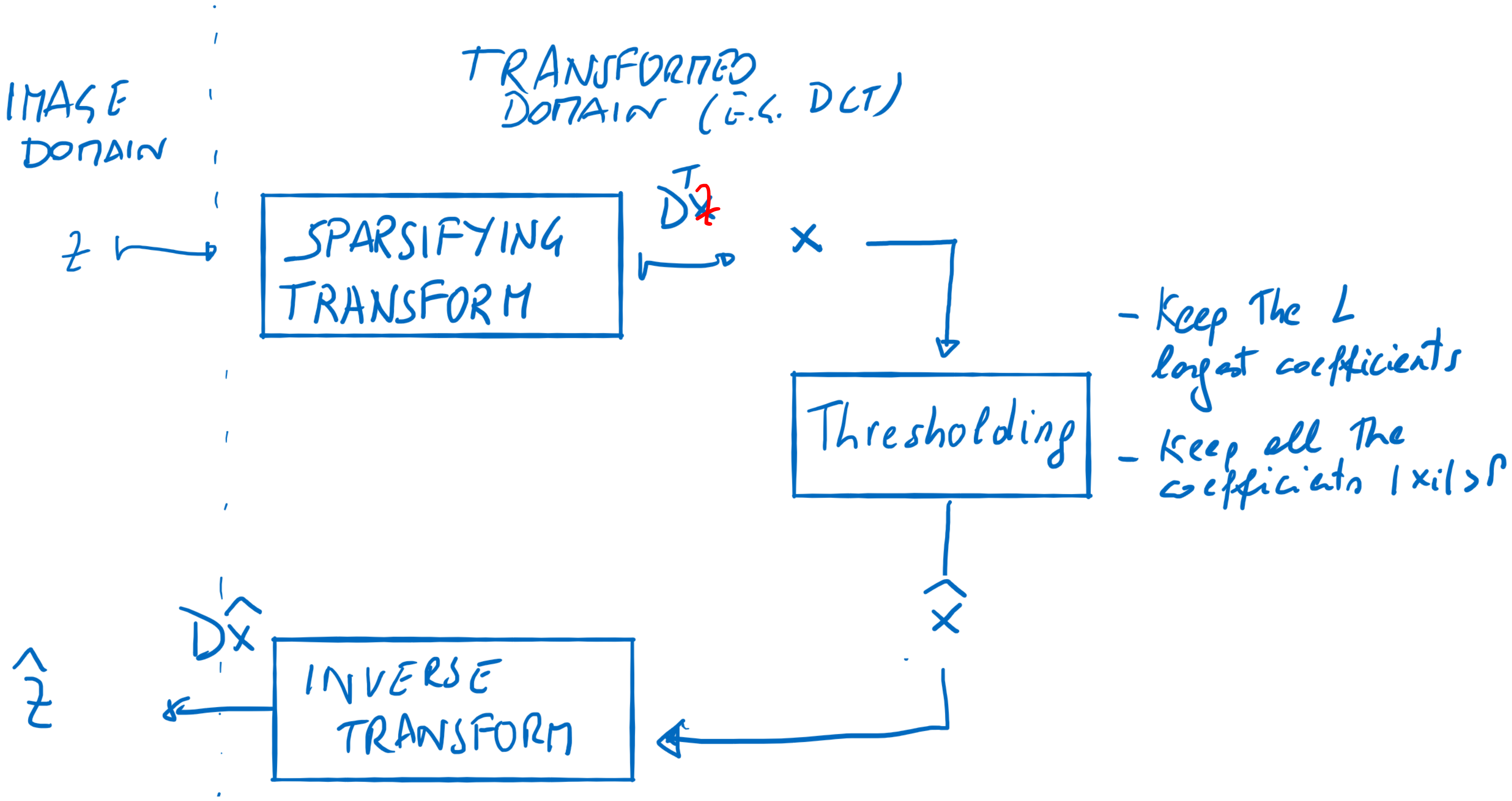
DCT: Discrete Cosine Transform is often used at patch-level. These are the elements of the basis, which are then arranged column-wise in the matrix D

Sparsity Promoting Denoising

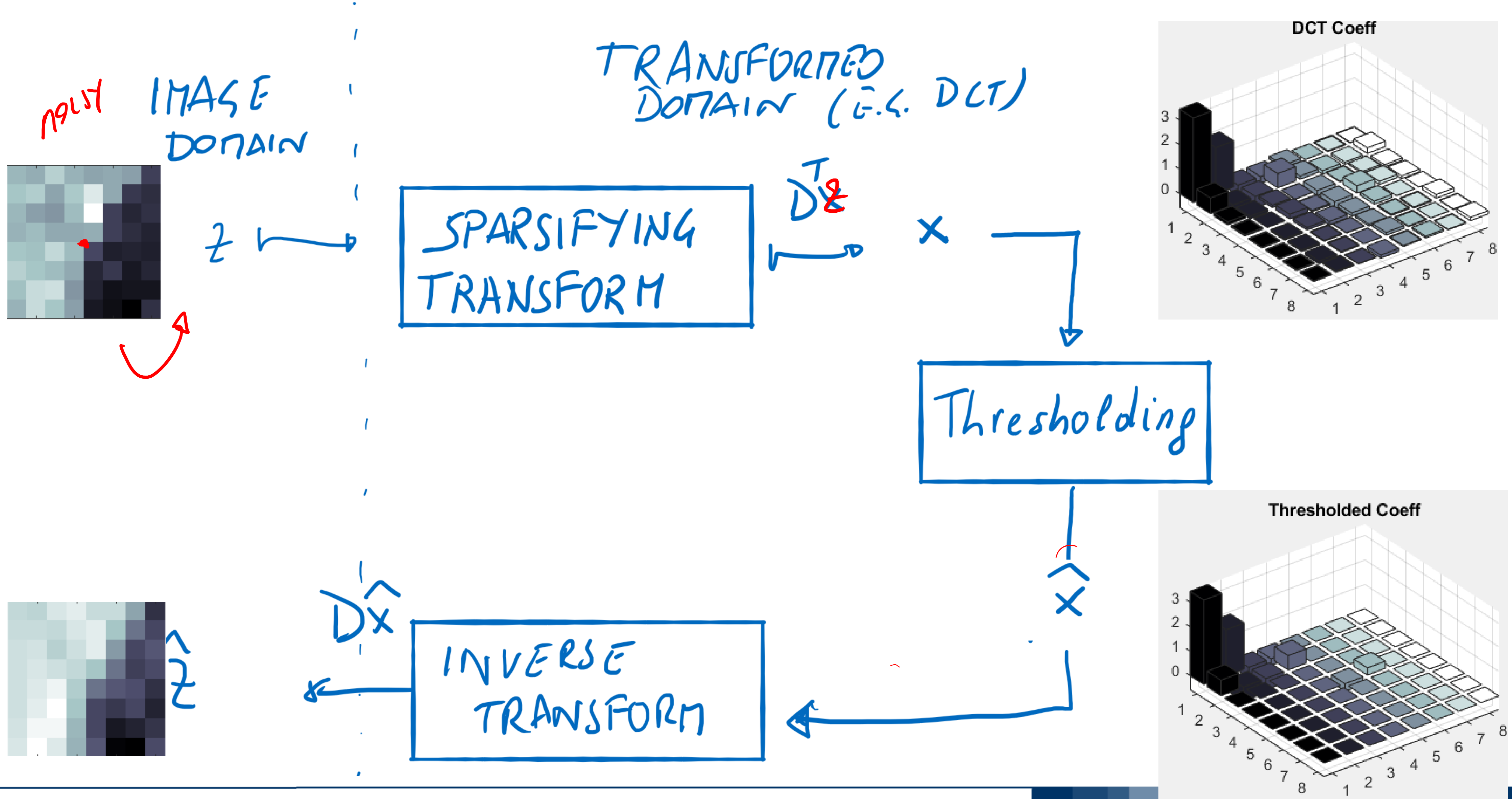


When the transformation is not w.r.t. an orthonormal basis, decomposition equation is not as simple. In case or redundant set of generators the representation is not unique and it has to «be pursued»

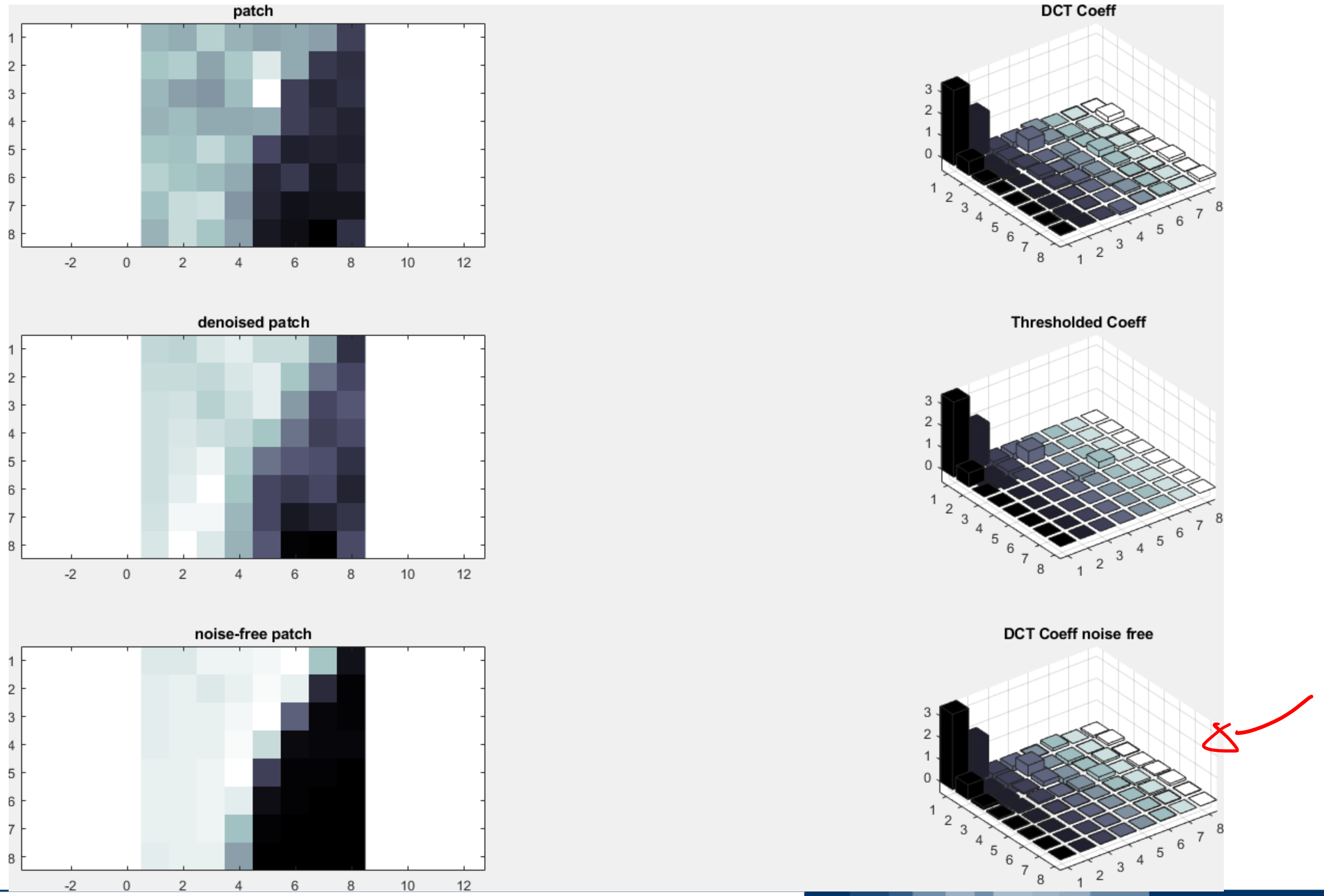
Sparsity Promoting Denoising



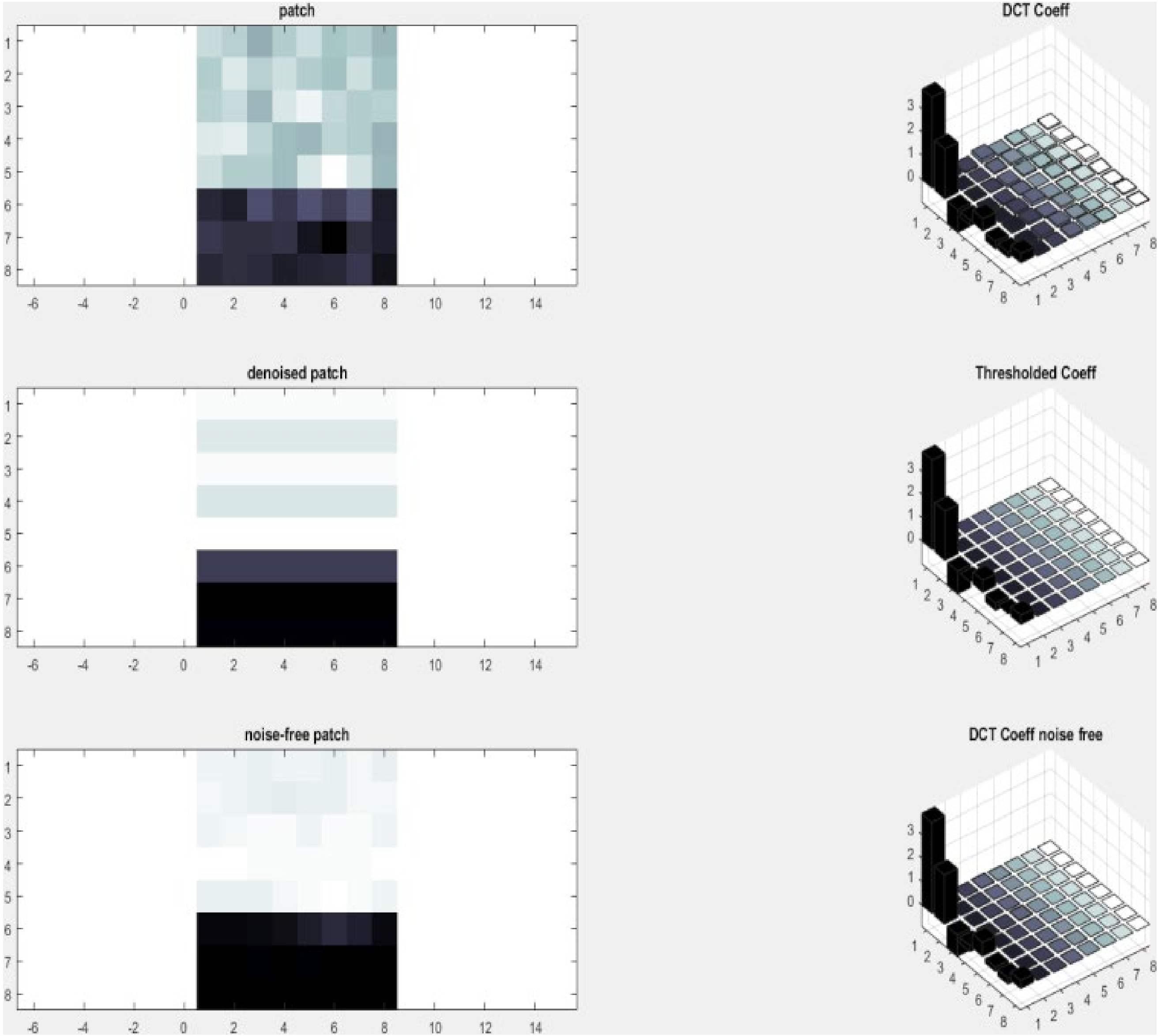
Sparsity Promoting Denoising



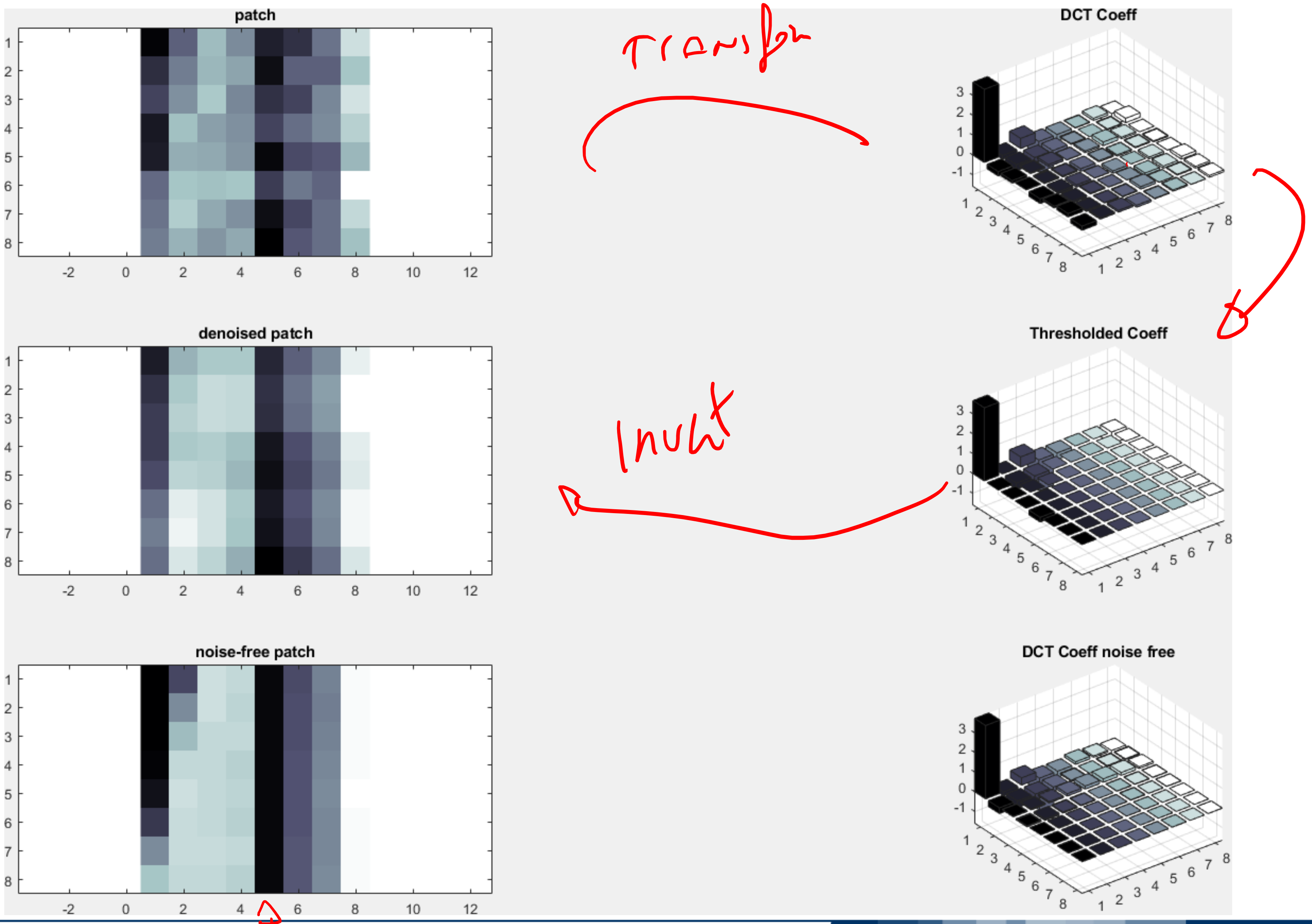
Sparsity Promoting Denoising



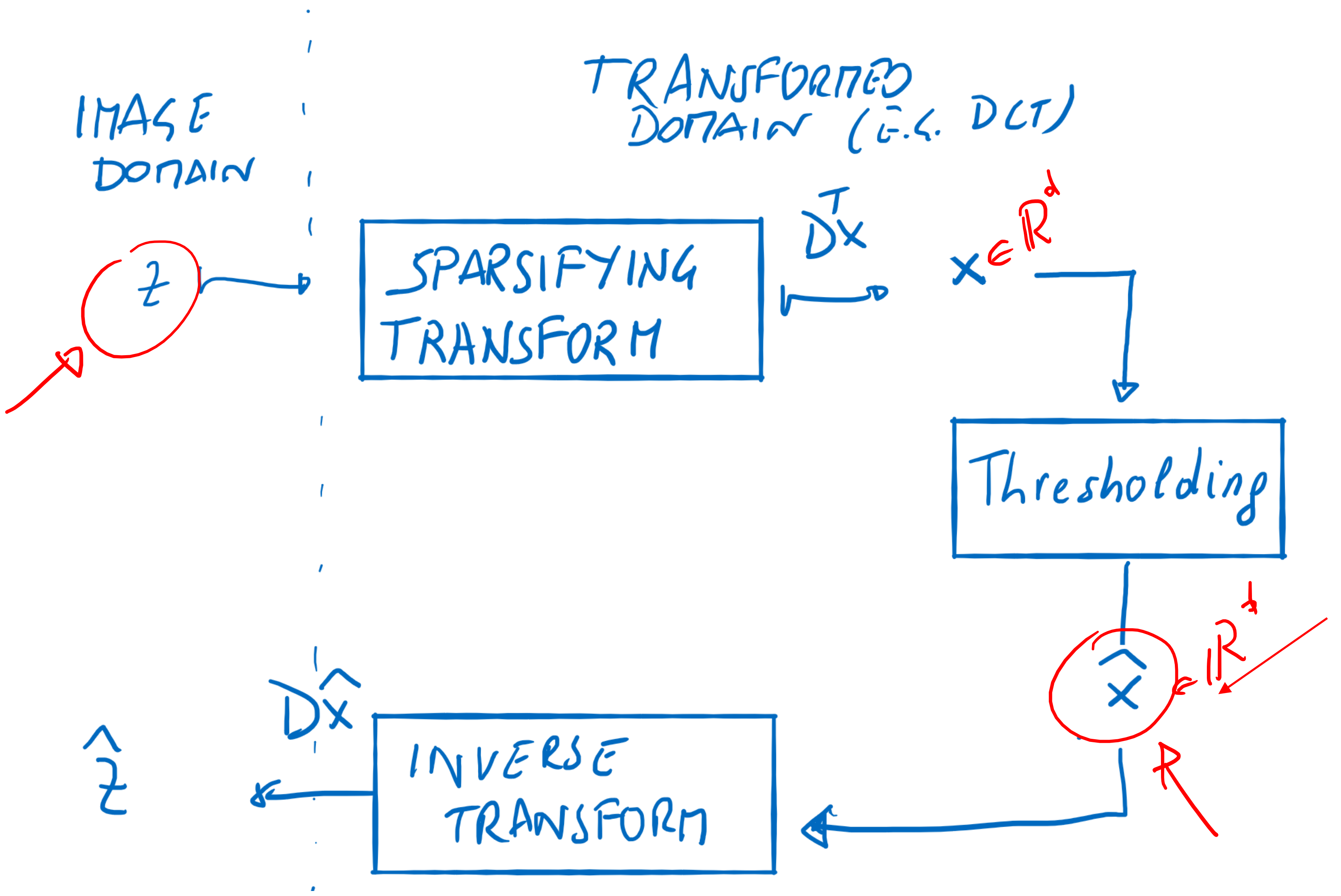
Sparsity Promoting Denoising



Sparsity Promoting Denoising



Sparsity Promoting Denoising

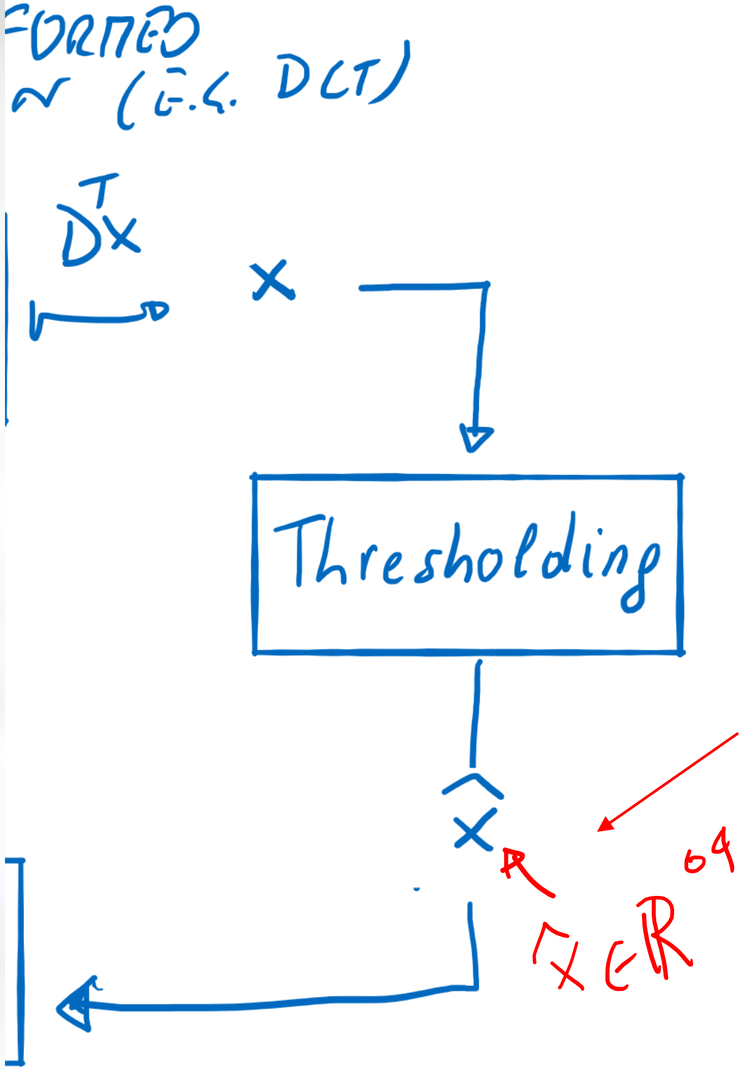


This is a much more compressed representation than z.

Encoding $\hat{\boldsymbol{x}}$ **instead of** $\boldsymbol{z}$ can significantly reduce the size of the image

JPEG Compression performs encoding of $\hat{x}$

Sparsity Promoting Denoising



This is a much more compressed representation than z.

Encoding $\hat{x}$ **instead of** z can significantly reduce the size of the image

JPEG Compression performs encoding of $\hat{x}$

The Course Outline

What you'll learn

The Outline (first half, sparse coding)

- Basics on linear orthonormal representations
  - Transform-domain representation and sparsity.
  - Mathematical Background.
- Greedy Sparse Coding Algorithms for minimum ℓ^0 norm
  - Matching Pursuit, Orthogonal Matching Pursuit.
- Convex Relaxation, minimum ℓ^1 norm sparse coding algorithm
  - Sparse coding as a convex optimization problem, BPDN (LASSO), connections with ℓ^0 solutions.
  - Notes on other norms, visual intuition.
  - Iterative Reweighted Least Squares, Proximal Methods, Iterative Soft Thresholding.

The outline (second half, dictionary learning)

- Dictionary Learning
  - Dictionary Learning Algorithms: Gradient Descent, MOD, KSVD

- Extended sparse models
  - Joint Sparsity, Group Sparsity, Sparse Coding Algorithms
  - Convolutional Sparsity.

- Sparsity in engineering applications
  - Major imaging problems involving sparsity as a regularization prior: Denoising, Inpainting, Superresolution, Deblurring.
  - Dictionary Learning and Sparse Coding for Classification and Anomaly Detection

Assignments on 1D signals

You can implement these codes in the programming language you prefer, if you do not feel comfortable using Matlab or Python

Today's Assignment: Generate the Basis

Generate the DCT basis according to the following formula (DCT type II) the k-th atom of the DCT basis in dimension M is defined as

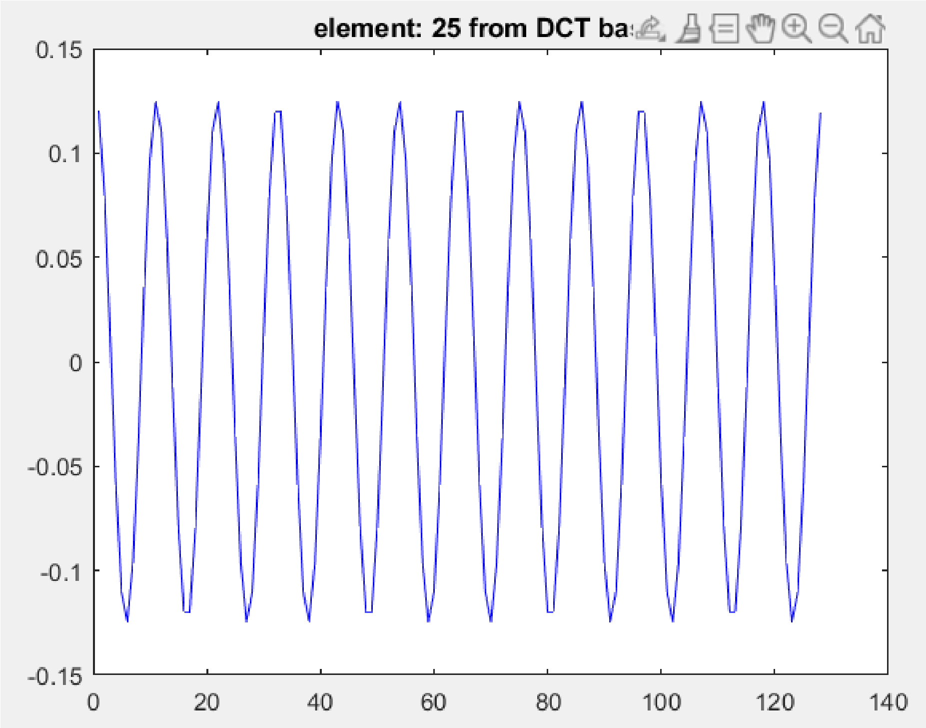
$$DCT_k(n) = c_k \cos\left(k\pi \frac{2n+1}{2M}\right) \quad n, k = 0, \dots, M-1$$

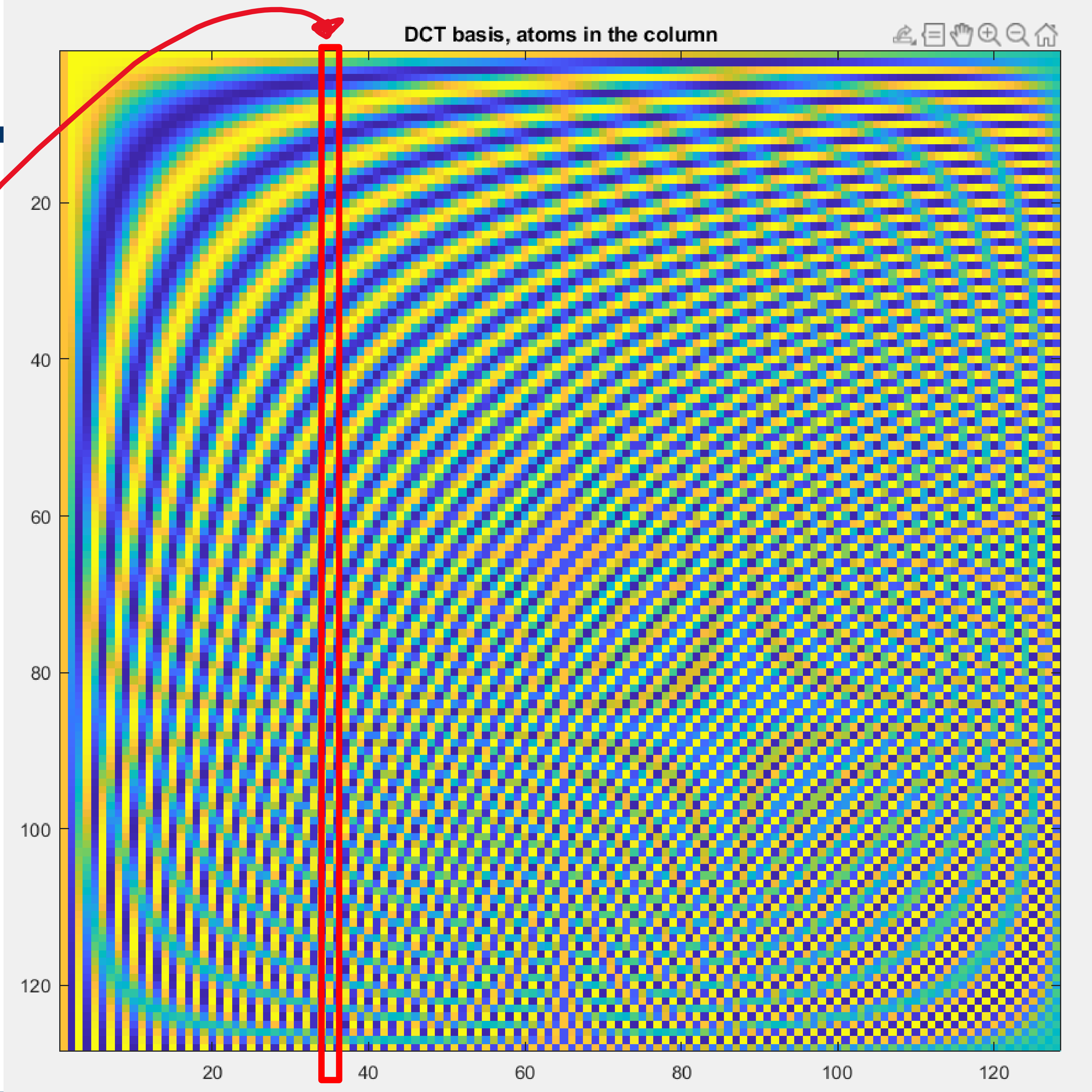
where $c_0 = \sqrt{1/M}$ and $c_k = \sqrt{2/M}$ for $k \neq 0$.

For each $k = 0, \dots, M-1$, just sample each function $\cos\left(k\pi \frac{2n+1}{2M}\right)$ at $n = 0, \dots, M-1$, obtain a vector. Ignore the normalization coefficient. Divide each vector by its ℓ_2 norm.

How can you use the function dct and it's inverse idct to define the DCT matrix?

The Matrix Should Look Like



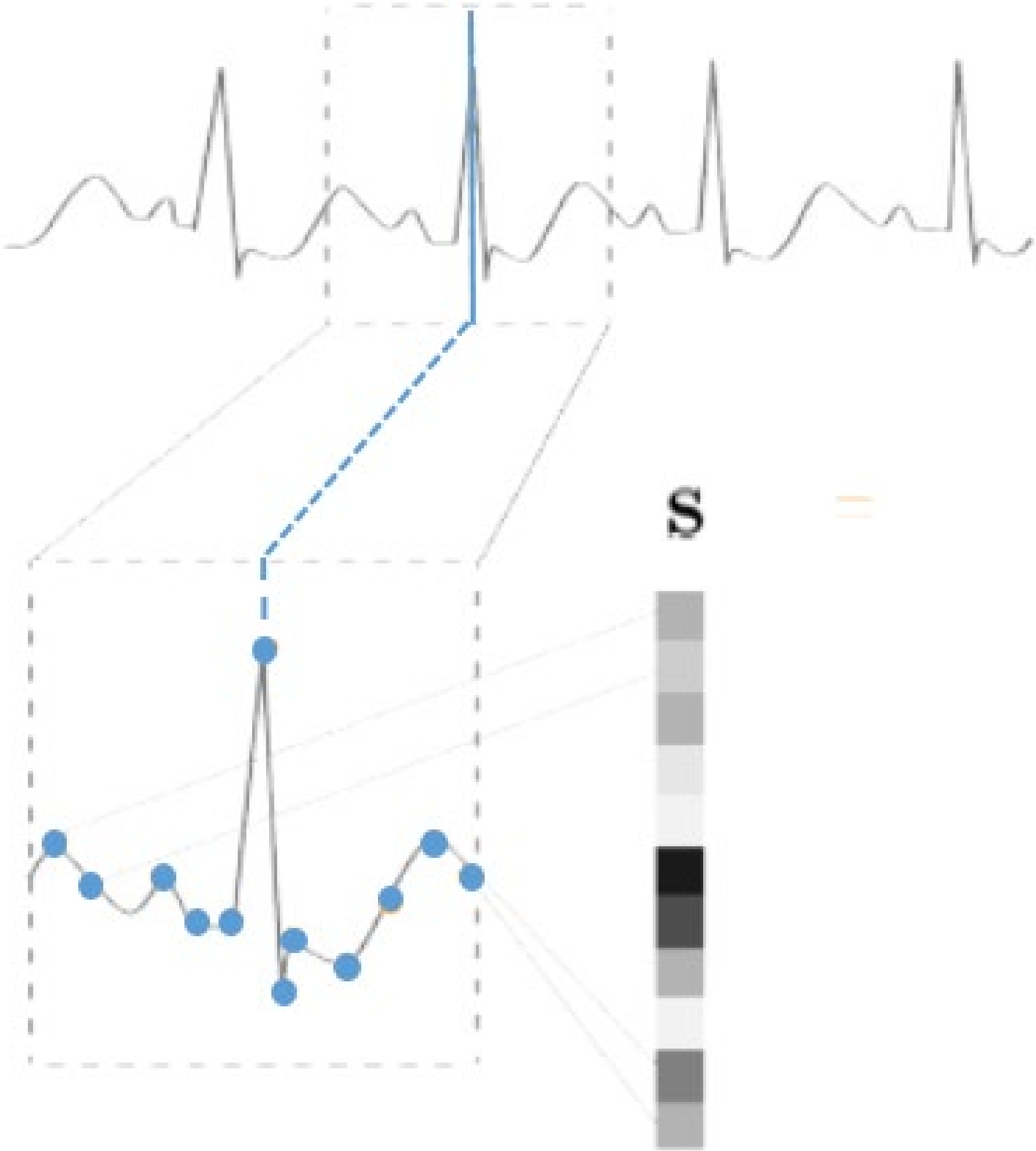


Today's Assignment: Analysis and Synthesis

- Load the ECG traces
- Analysis: Use the DCT basis you have defined to compute the representation of each signal $\boldsymbol{s}$ w.r.t the basis
- Display the coefficients and check whether they are sparse
- Synthesis: Reconstruct the signal from the coefficients. Check whether the reconstruction is perfect

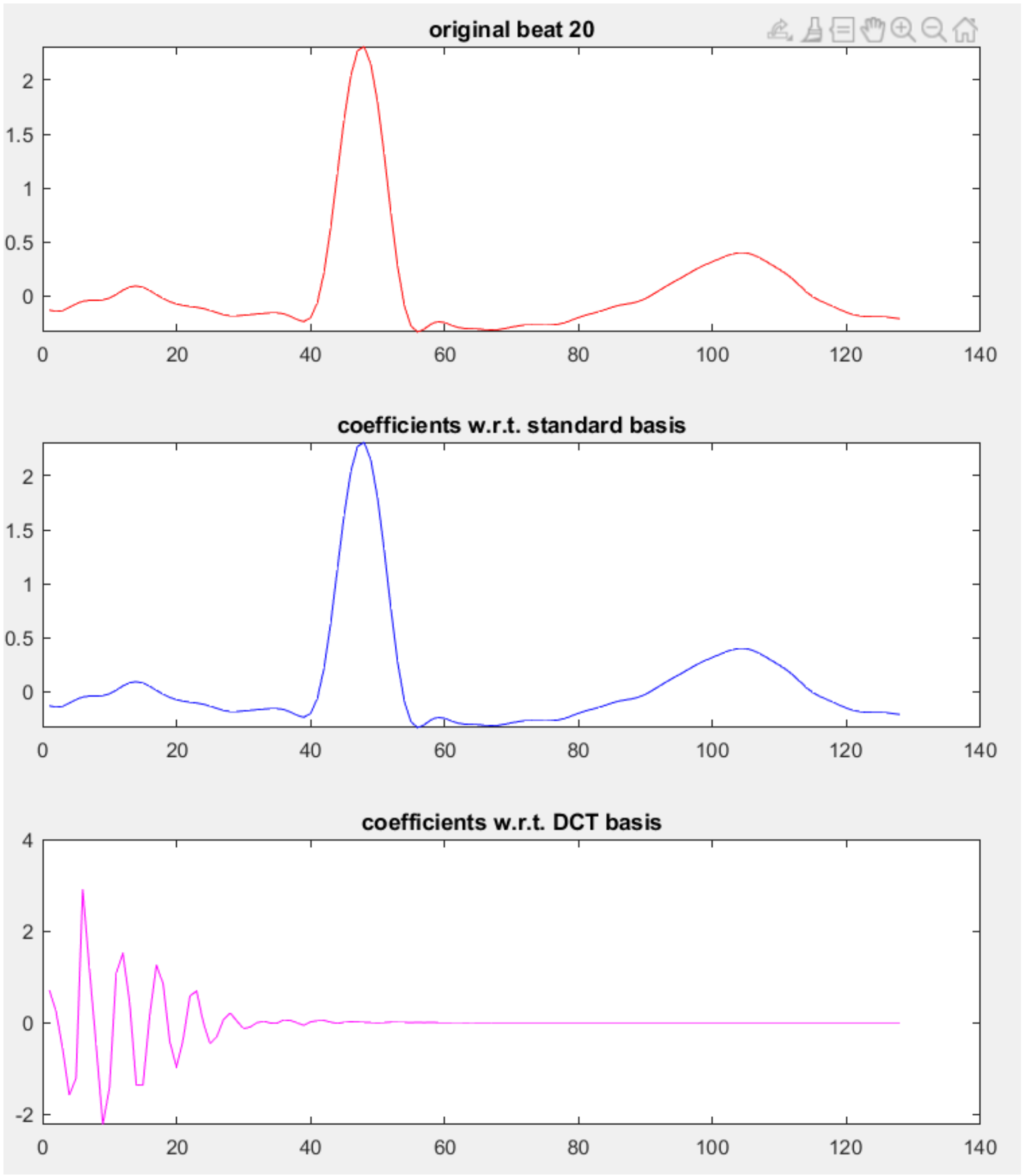
Repeat the process by first adding noise to the traces

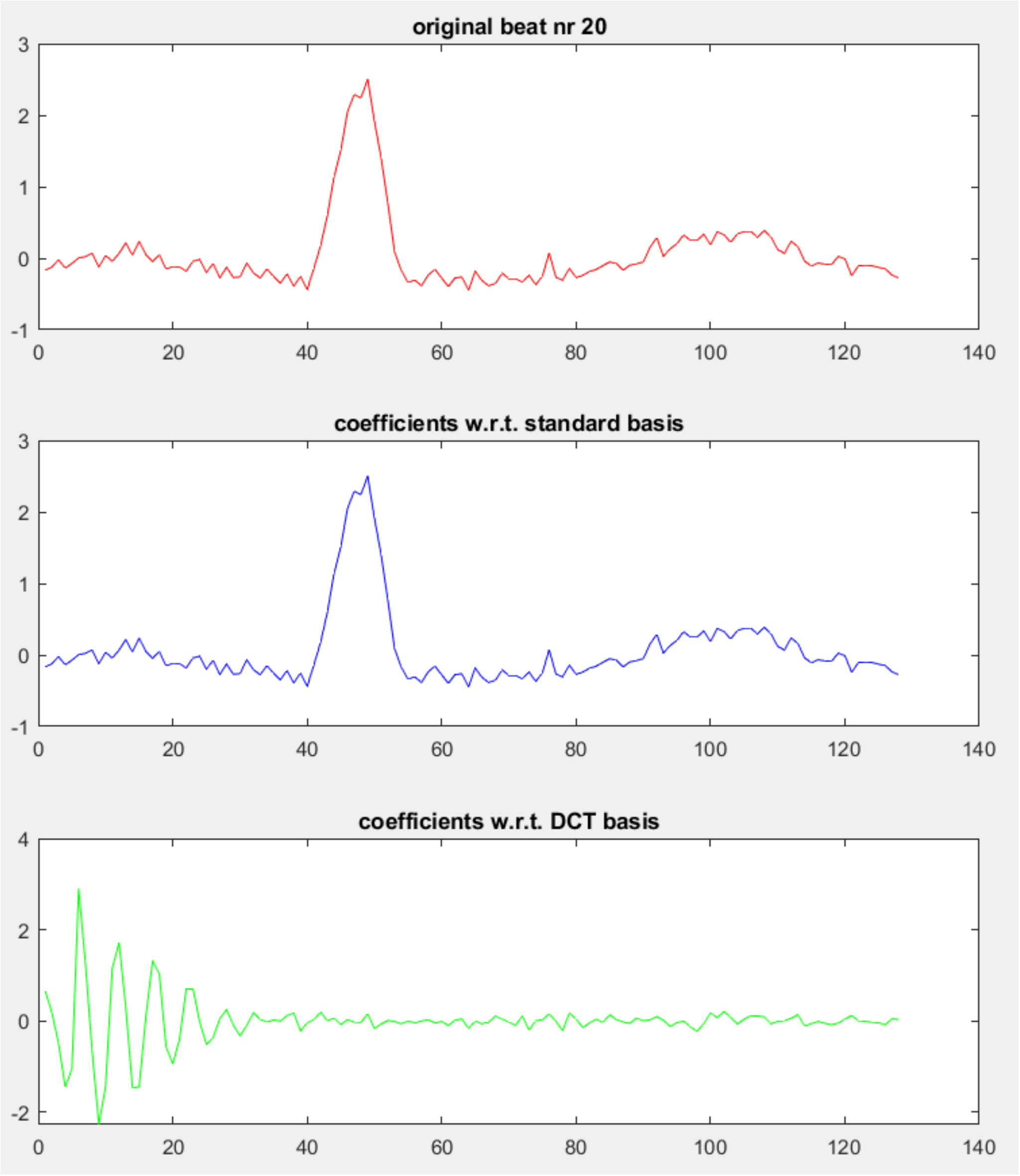
Modeling Scheme



w/o noise

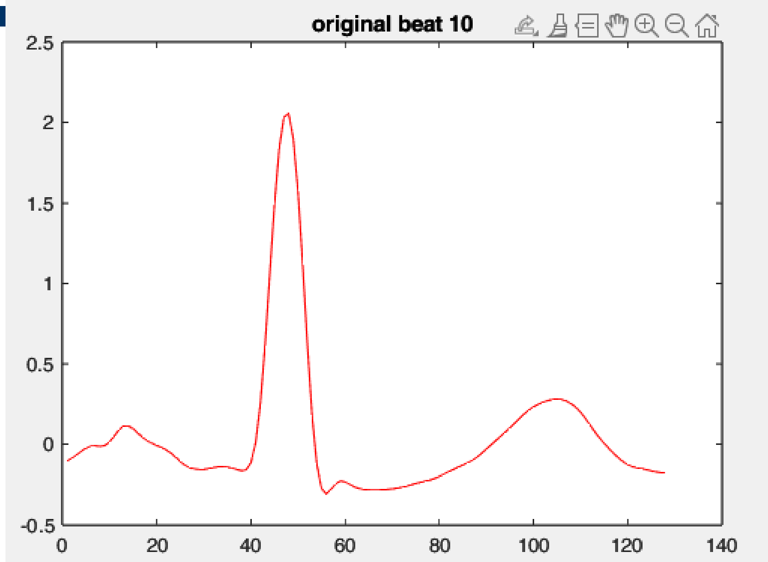
w/ noise

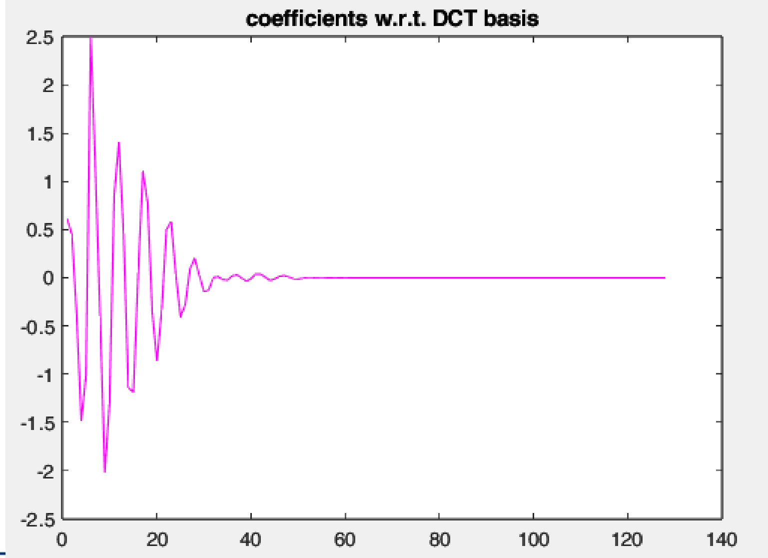




w/o noise

w/ noise





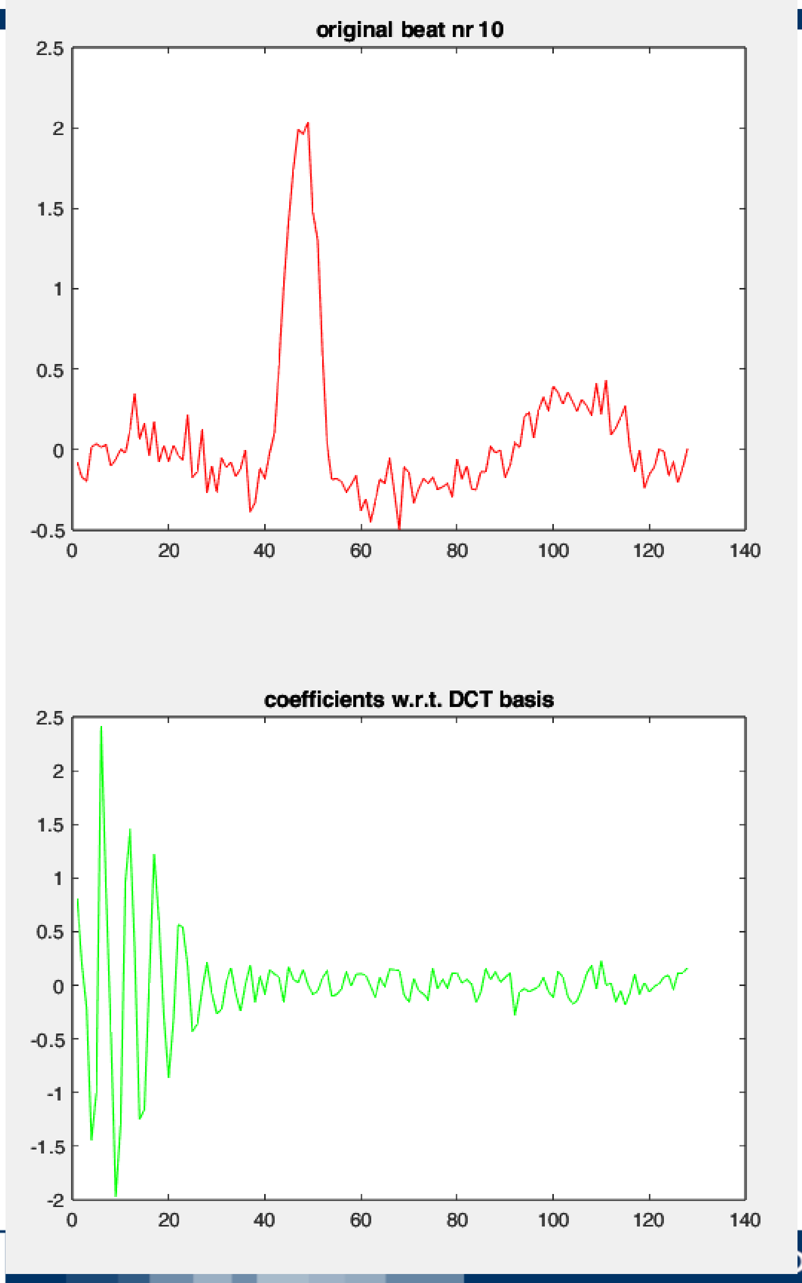
The sparsity prior seems to be effective on this type of signals. There are only few nonzero coefficients in here.

DCT bases yields a sparse representation of heartbeats

w/o noise w/ noise

Still, the coefficients referring to the noise-free signal have a magnitude that is larger than those coefficients that are only affected by noise

However, the large coefficients are also affected by noise (therefore getting rid of the smallest coefficients won’t return the noise-free signal)

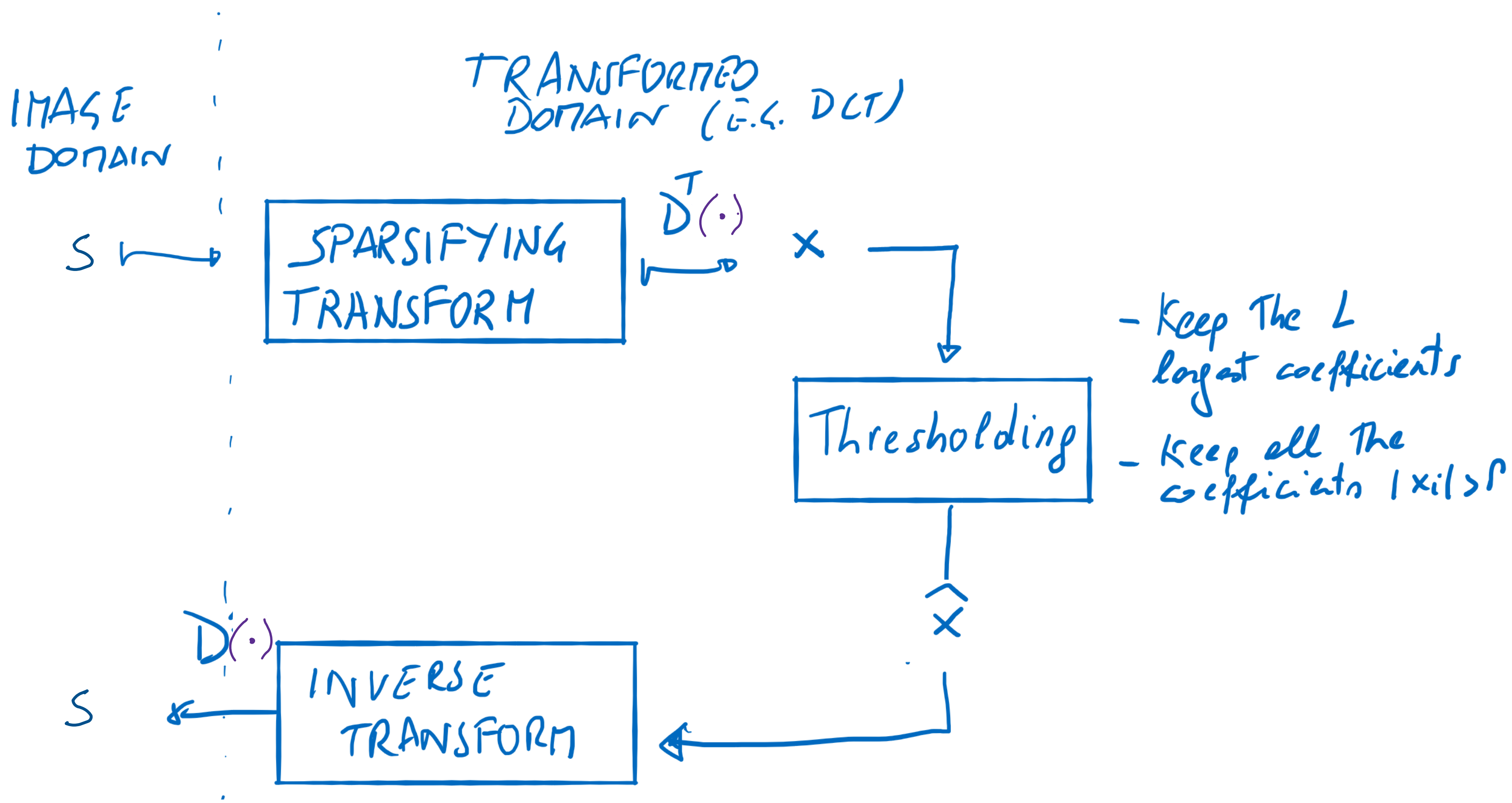


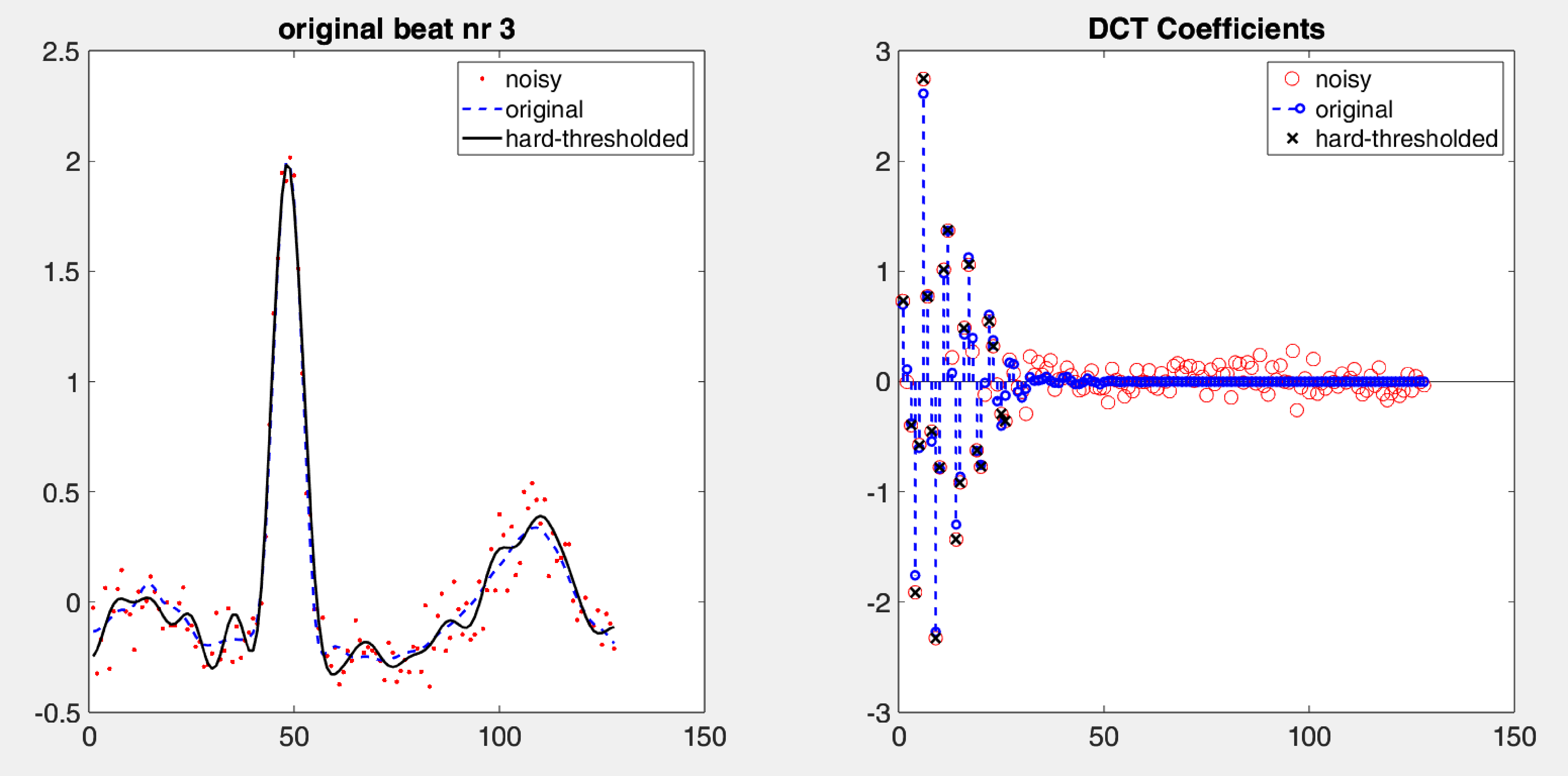
Programming Exercise

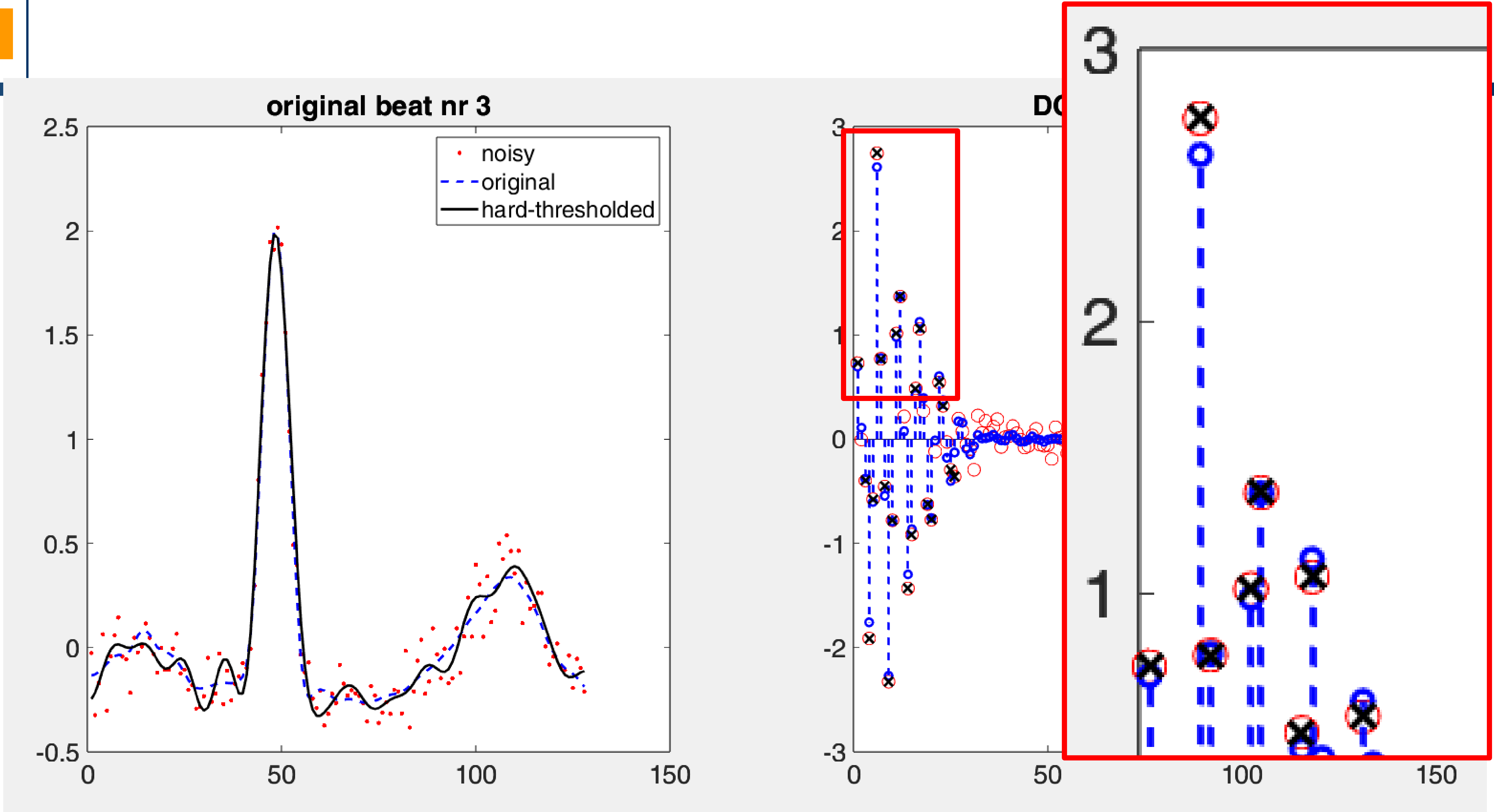
- Implement a function that performs hard thresholding w.r.t the 1D DCT basis for ECG signals
- Run the Hard Thresholding Denoising on the ECG signals + noise
- Run the Hard Thresholding Denoising on the ECG signals + spike (see next)

Today's Assignment: Enforce Sparsity

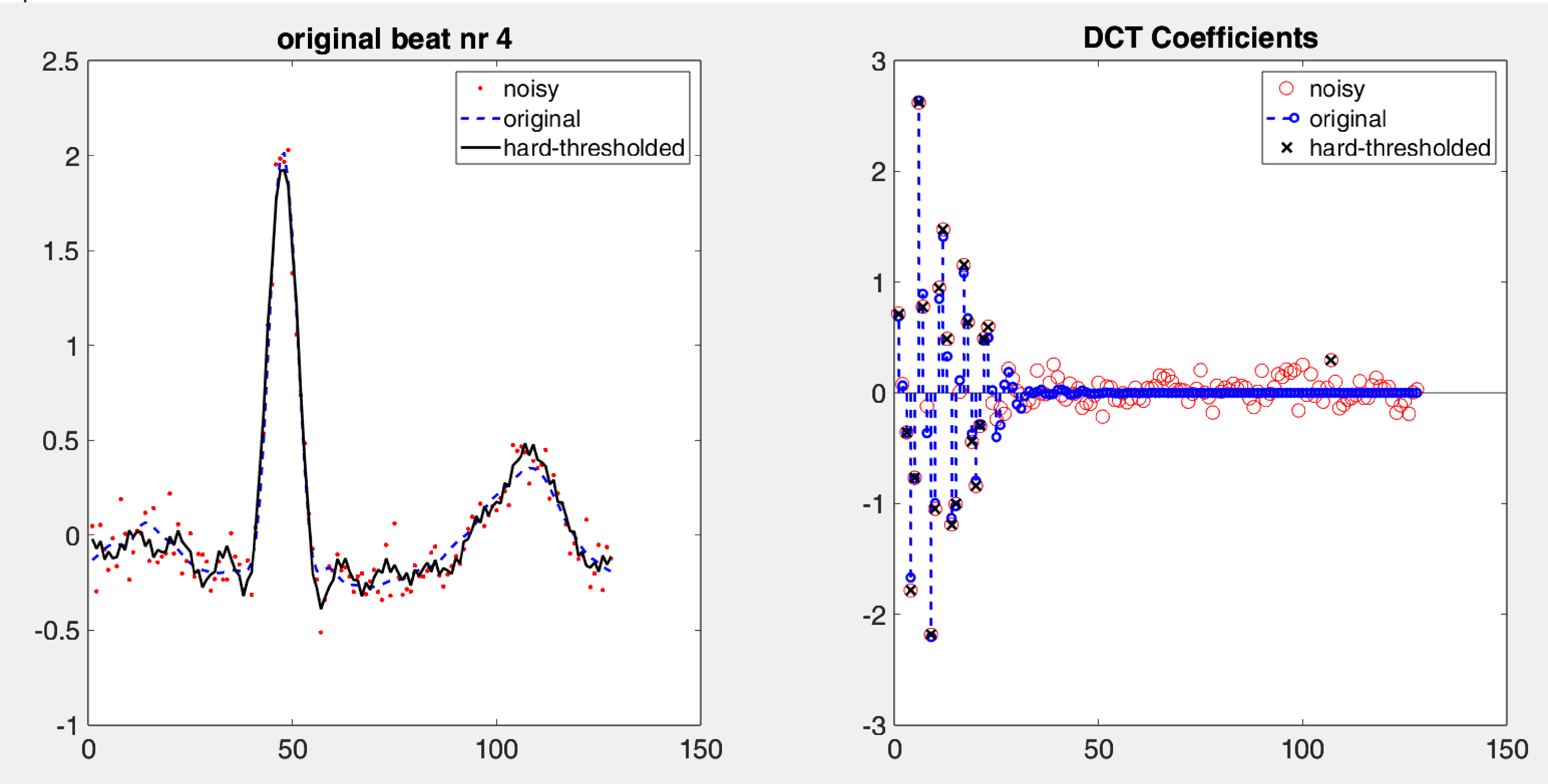
- Enforce Sparsity to get rid of noise

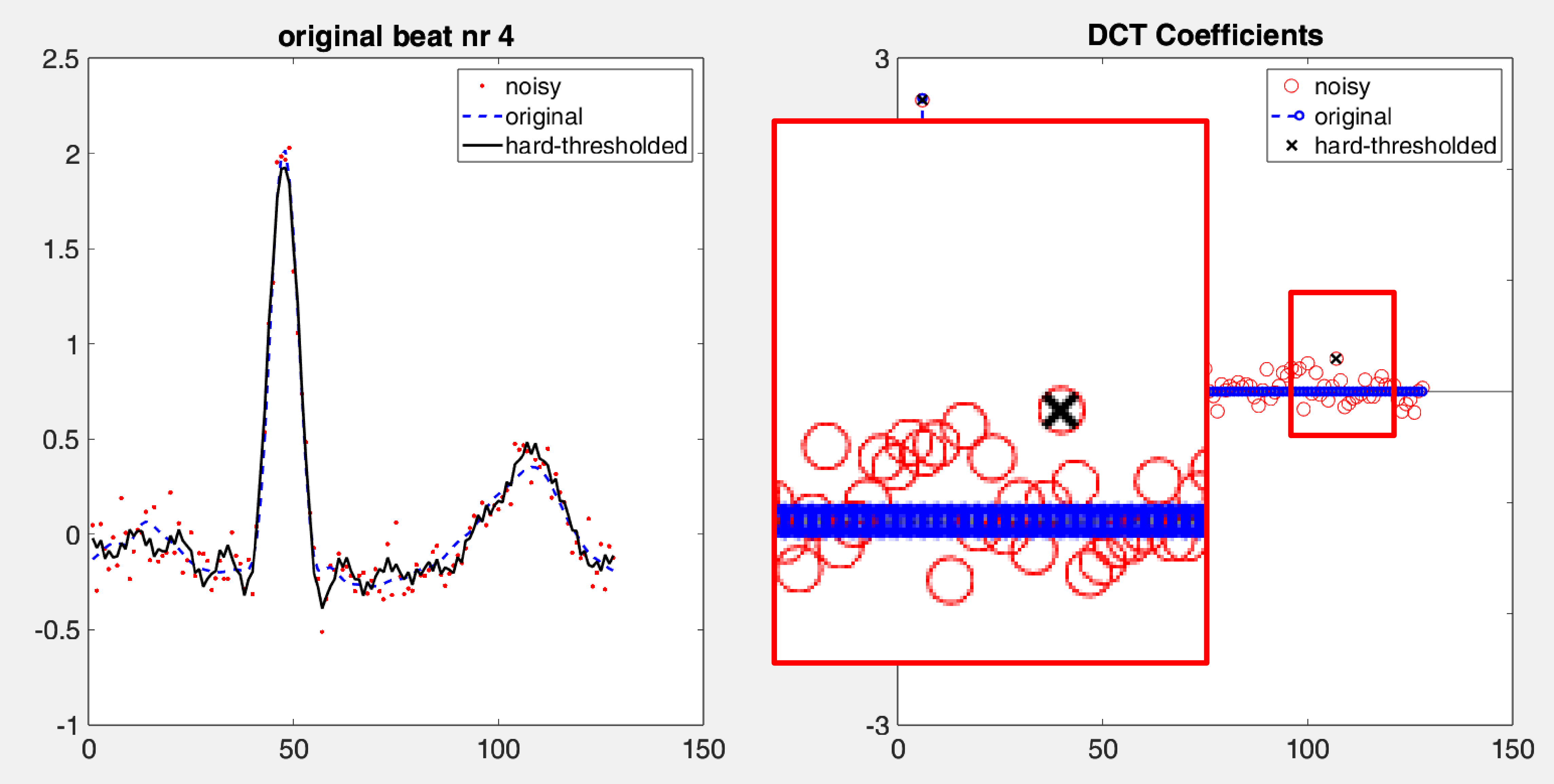



original beat nr 3
noisy
original
hard-thresholded
2.5
2
1.5
1
0.5
0
-0.5
0
50
100
150
DCT Coefficients
noisy
original
hard-thresholded
3
2
1
0
-1
-2
-3
0
50
100
150



original beat nr 3
noisy
original
hard-thresholded
DC


original beat nr 4
noisy
original
hard-thresholded
2.5
2
1.5
1
0.5
0
-0.5
-1
0
50
100
150
DCT Coefficients
noisy
original
hard-thresholded
3
2
1
0
-1
-2
-3
0
50
100
150


original beat nr 4
noisy
original
hard-thresholded
2.5
2
1.5
1
0.5
0
-0.5
-1
0
50
100
150
DCT Coefficients
noisy
original
hard-thresholded
3
-3
0
50
100
150

Assignment: Generate a truly sparse 1D signal w.r.t. D

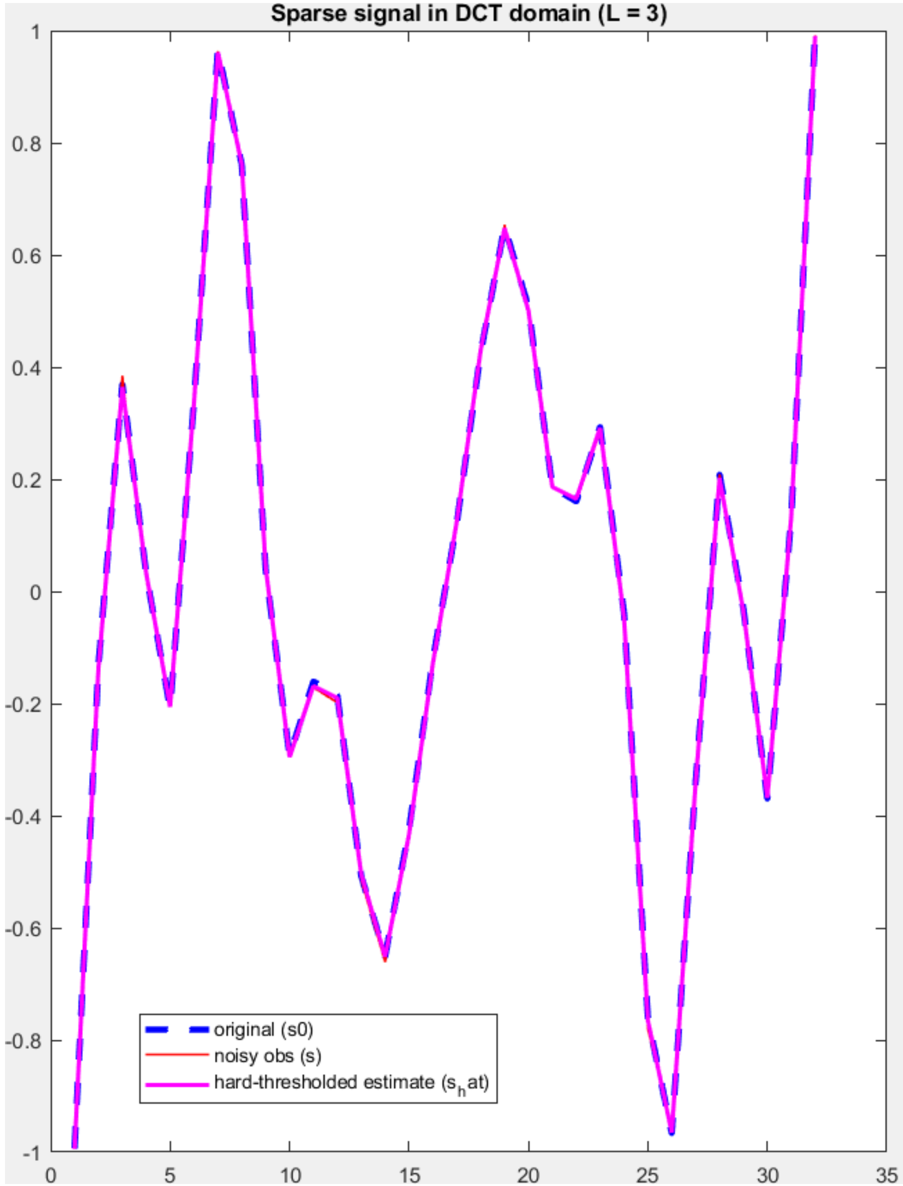
Idea:

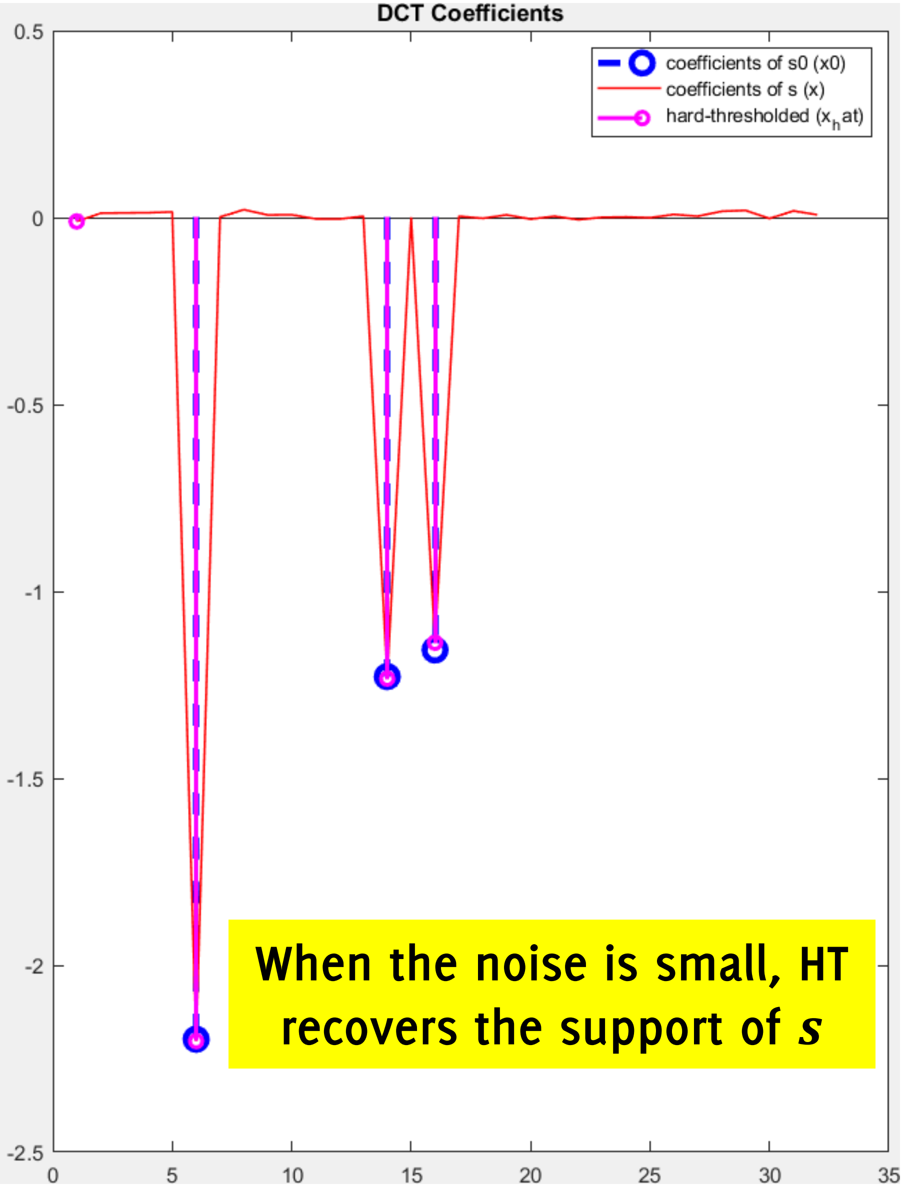
1. Randomly define sparse coefficients x_0 of size M
2. Synthesis w.r.t. a DCT dictionary, i.e. compute $s_0 = Dx_0$
3. Add white Gaussian noise η: $s = s_0 + \eta$

Rmk:

s might not look very realistic, but this is truly sparse w.r.t. D

Generate a truly sparse signal w.r.t. D

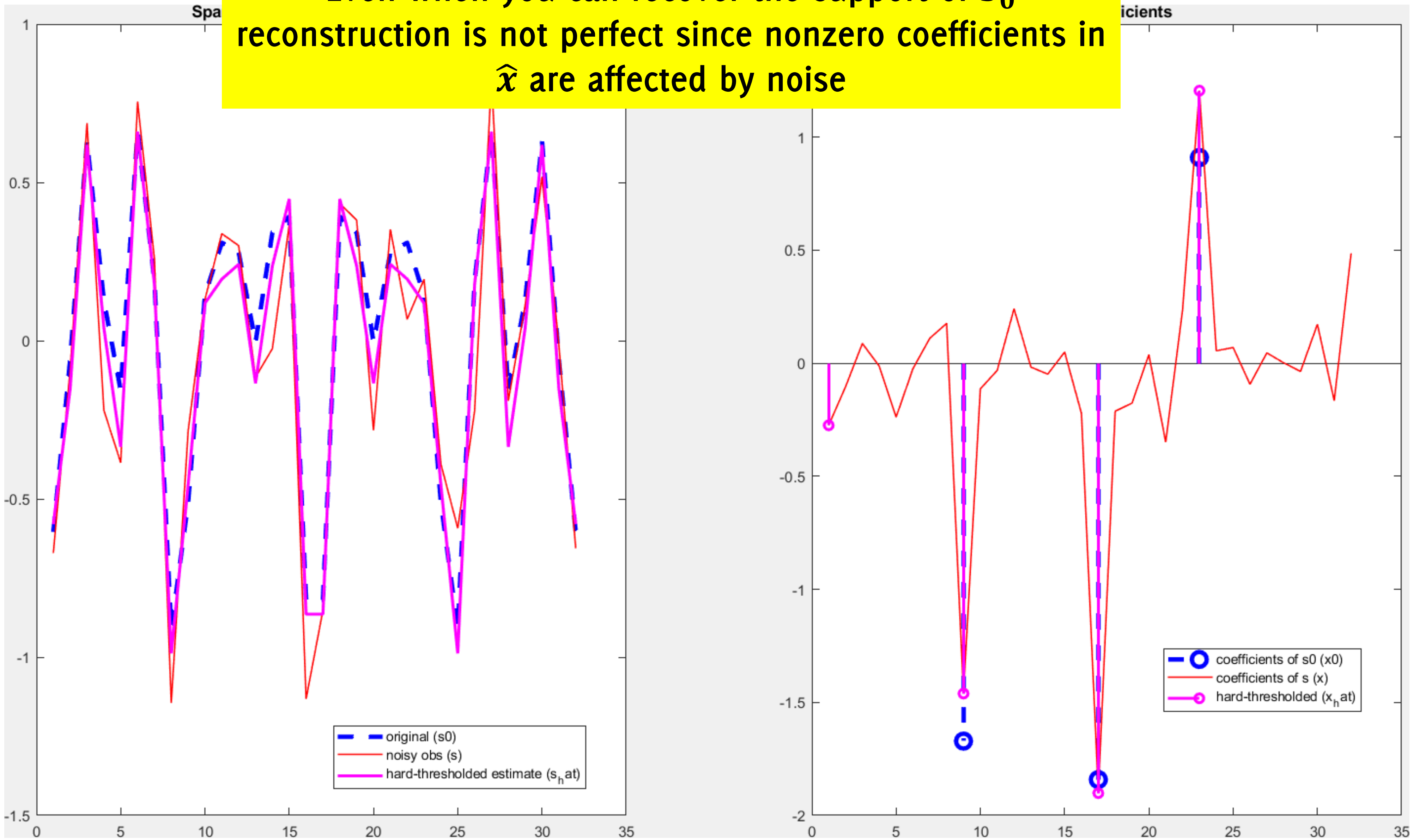




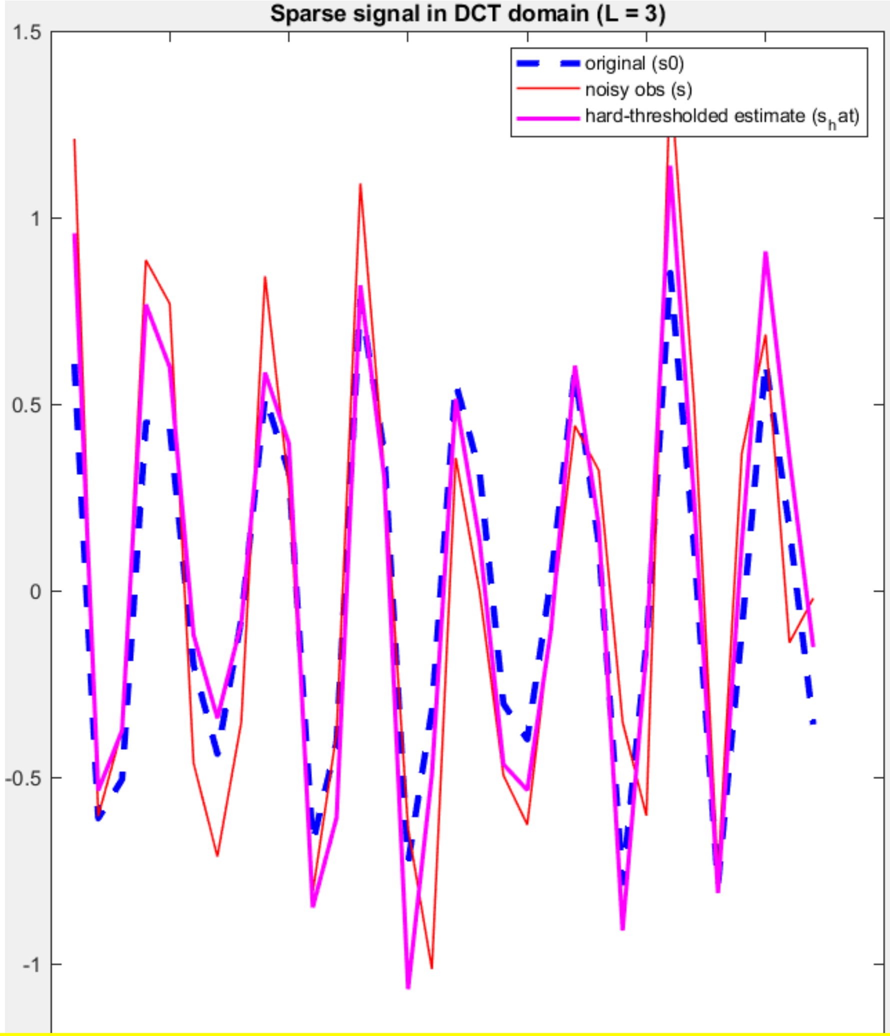
When the noise is small, HT recovers the support of s

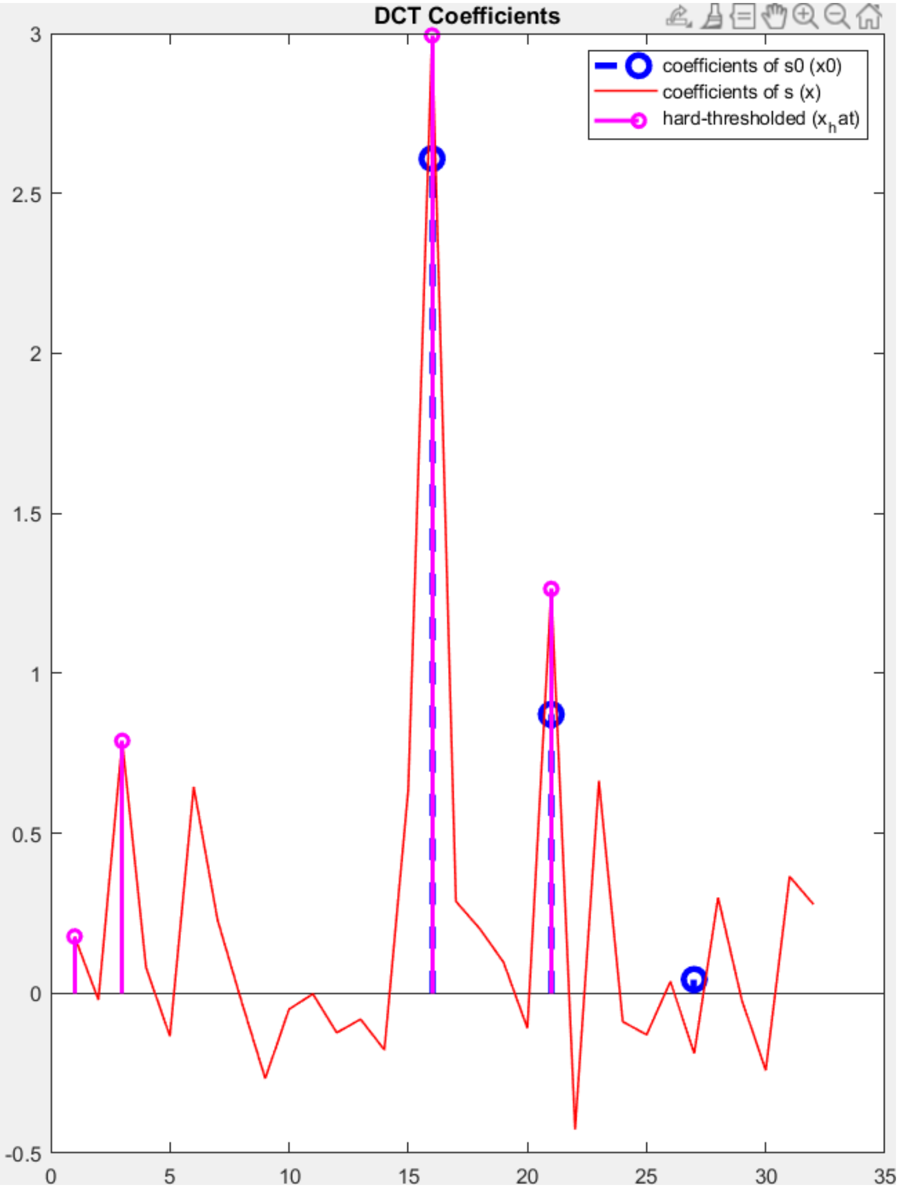
Generate a truly sparse signal w.r.t. D

Even when you can recover the support of s_0 reconstruction is not perfect since nonzero coefficients in $\hat{x}$ are affected by noise



Generate a truly sparse signal w.r.t. D





When the noise is large, HT might fail even at recovering the support of x_0

Now, assume your signal is sparse w.r.t. $[D, C]$

Idea:

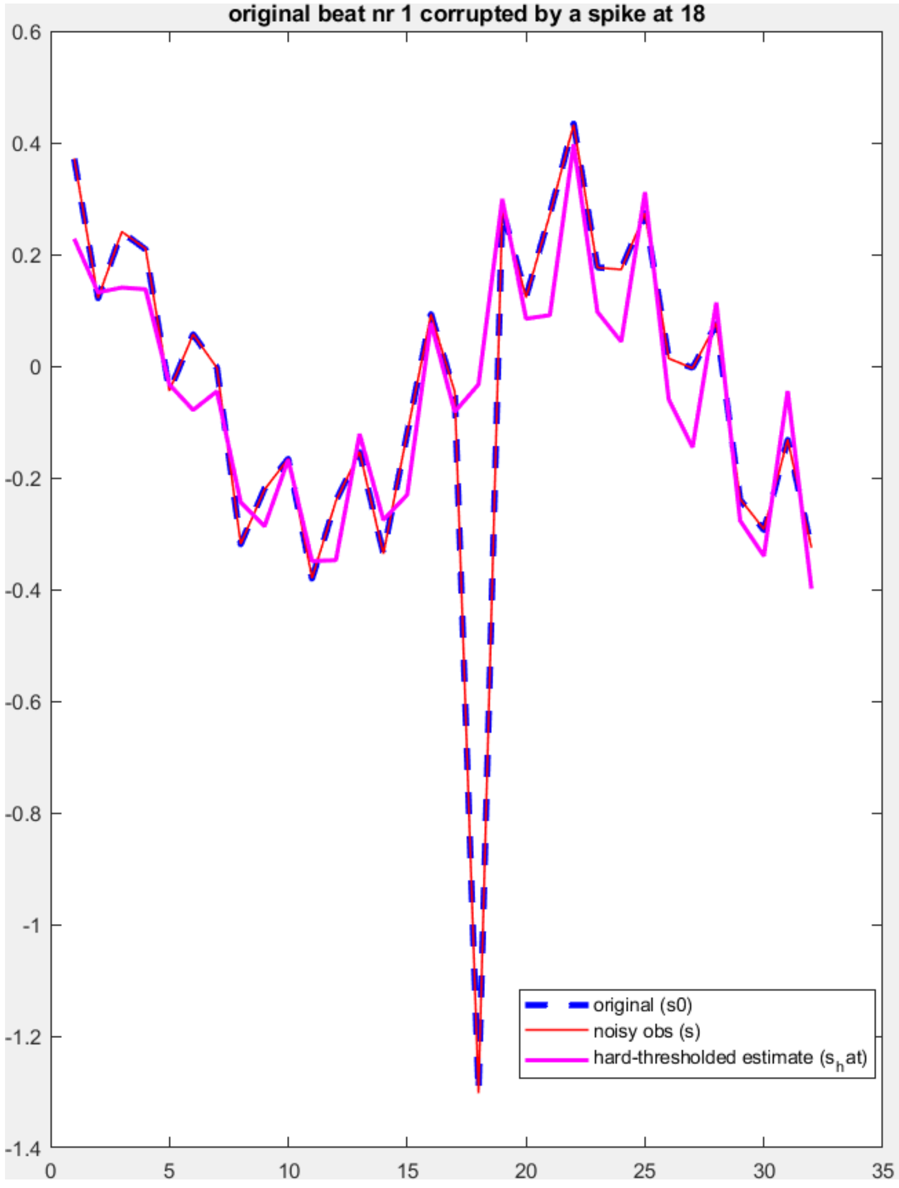
1. Randomly define sparse coefficients x_0
2. Synthesis w.r.t. a DCT dictionary, i.e. compute $s_0 = Dx_0$
3. Add a spike δ_c at location c, which is a sparse element w.r.t. C

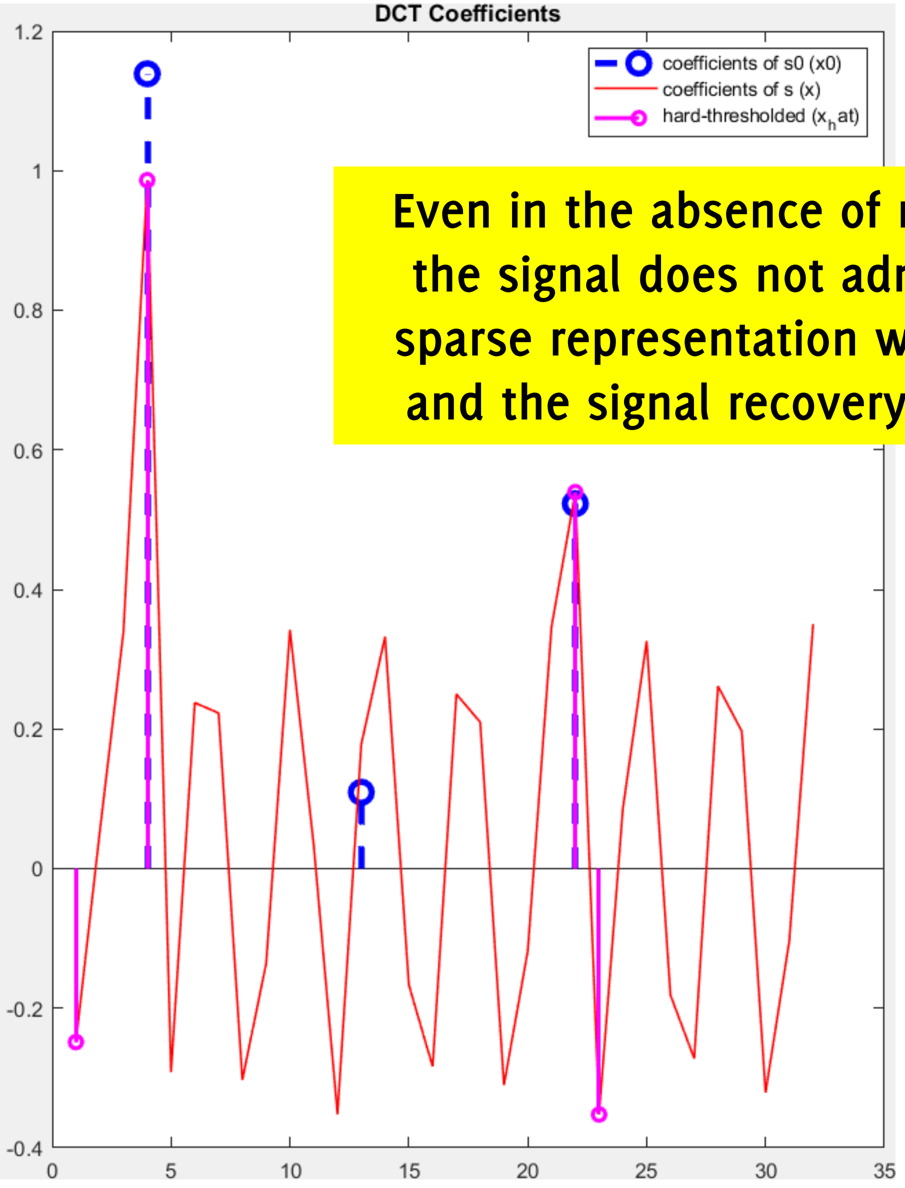
$$s_0 = s_0 + \lambda\delta_c$$

   where λ and c are randomly defined

4. Add noise: $s = s_0 + \eta$

Truly sparse signals w.r.t. $[D, C]$

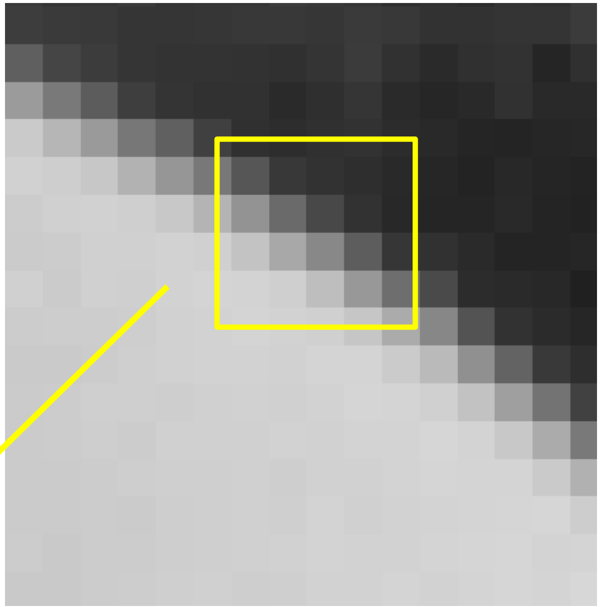




Even in the absence of noise, the signal does not admit a sparse representation w.r.t. D and the signal recovery fails

Assignments on Image Denoising

Image Denoising and Patch-wise processing



123	102	94	82	77
133	110	94	78	67
121	116	101	83	71
127	128	131	96	69
119	127	137	119	89

a 5 ×5 patch

Sliding DCT Denoising

- A very powerful, yet simple denoising algorithm that can pair much more sophisticated alternatives

- A description of the algorithm steps can be found here

Yu, Guoshen, and Guillermo Sapiro. "DCT image denoising: a simple and effective image denoising algorithm." *Image Processing On Line* 1 (2011): 292-296.

https://www.ipol.im/pub/art/2011/ys-dct/article.pdf

Assignments on Image Denoising

- Build the matrix containing 2D-DCT basis (each column is a basis element, stacked in a vector.)
- Implement an image denoising algorithm by performing Hard Thresholding in DCT domain on each patch. In particular
  - Divide image in a tessellation of patches (e.g. 8x8 pixels)
  - Transform in DCT domain each patch and perform HT
  - Return estimated patches to their original location
- What happens when HT is applied to a noise-free signal/image?
- What if you change the sparsity level L? What if $L = M$?

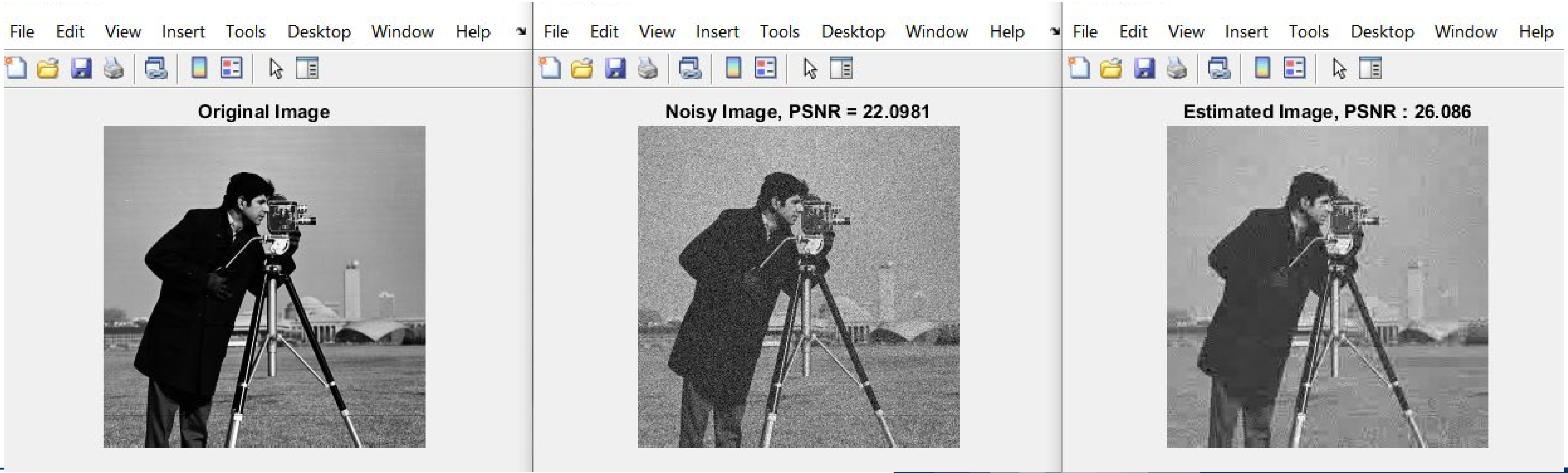
Implement Sliding DCT denoising algorithm.

- Modify the `im2patch` function to extract and rearrange each patch in the image (i.e. extract a patch centered in each pixel).
- Perform denoising by Hard Thresholding in 2D-DCT domain
- Modify the `patch2im` function to average estimates of overlapping patches. In particular, return each patch to its original location and sum to the other estimates. Keep track of the number of estimates you have been summing in each pixel and divide by this number to obtain aggregation using uniform weights.
- You can download standard test images from here:
  - http://www.cs.tut.fi/~foi/GCF-BM3D/BM3D_images.zip

Assess Denoising Performance

- Measure the PSNR of the denoised image
- $PSNR(\hat{y}, y) = 10\ \log_{10} \frac{1}{MSE(\hat{y},y)}$
- Where 1 stands for the signal peak (image is assumed to be in [0,1])

```
sigma_noise = 20/255; img = im2double(imread('cameraman.tif'));
```



Theory Excercises

A Simple Proof

Proof that if a set of vectors $\{\boldsymbol{e_i}\}$, $\boldsymbol{e}_i \in \mathbb{R}^M$ are linearly independent

and if

$$\boldsymbol{v} = \sum_i x_i \boldsymbol{e_i} \ , x_i \in \mathbb{R}$$

Then the representation $\{x_i\}$ is unique

Other exercises

- *What are the 1-sparse subspaces of* $\mathbb{R}^3$ *?*
  What are the 2-sparse subspaces of $\mathbb{R}^3$ *?*
- *Show that if a set of vectors* $\{d_i\}_{i=1,\dots,M}$ *is linearly independent, then a vector* $v \in < d_i >_{i=1,\dots,M}$ *admits a unique representation*
- *Show that if a set of vectors* $\{d_i\}$ *is linearly dependent, then any vector* $v \in < d_i >_{i=1,\dots,M}$ *has an infinite number of representation w.r.t.* $\{d_i\}_{i=1,\dots,M}$

- Define a set of 3 linearly independent vectors $a, b, c \in \mathbb{R}^3$ which does not form an orthonormal basis.
- Compute the representation of the standard basis vectors e_1, e_2, e_3 of $\mathbb{R}^3$ by solving a linear system.
- How are 1-sparse subspaces w.r.t a, b, c ?
- How are 2-sparse subspaces w.r.t a, b, c ?
- Do they differ from 1-sparse and 2-sparse subspaces w.r.t. e_1, e_2, e_3?