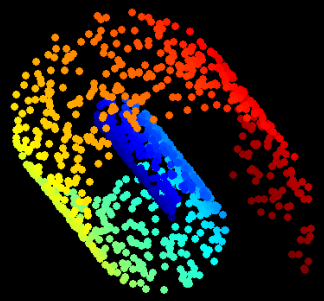
Wavelets for Graphs and their Deployment to Image Processing

Idan Ram

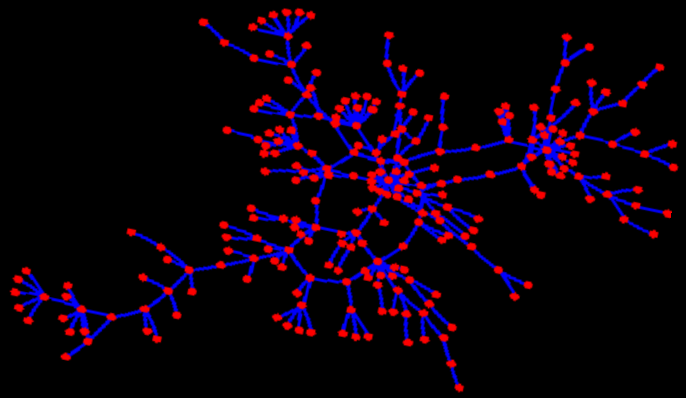
Electrical Engineering Department
Technion – Israel Institute of technology

Supervised by Prof. Israel Cohen and Prof. Michael Elad

This Talk is About …



Processing of Non-Conventionally Structured Signals



Many signal-processing tools (filters, alg., transforms, …) are desinged for uniformly sampled signals

Development of comparable methods capable of handling signals defined on graphs and point clouds is important and very much needed.

Our goal: Generalize the wavelet transform to handle this broad family of signals

The true objective: Find how to bring sparse representation to processing of such signals

We explore the use of our proposed methods for image processing

Outline

- Tree-Based Wavelet Transforms
- Image Processing using Tree-Based Wavelets
- Image Processing using Smooth Patch Ordering
- Conclusions

Tree-Based Wavelet Transforms

This part is taken from the following papers:

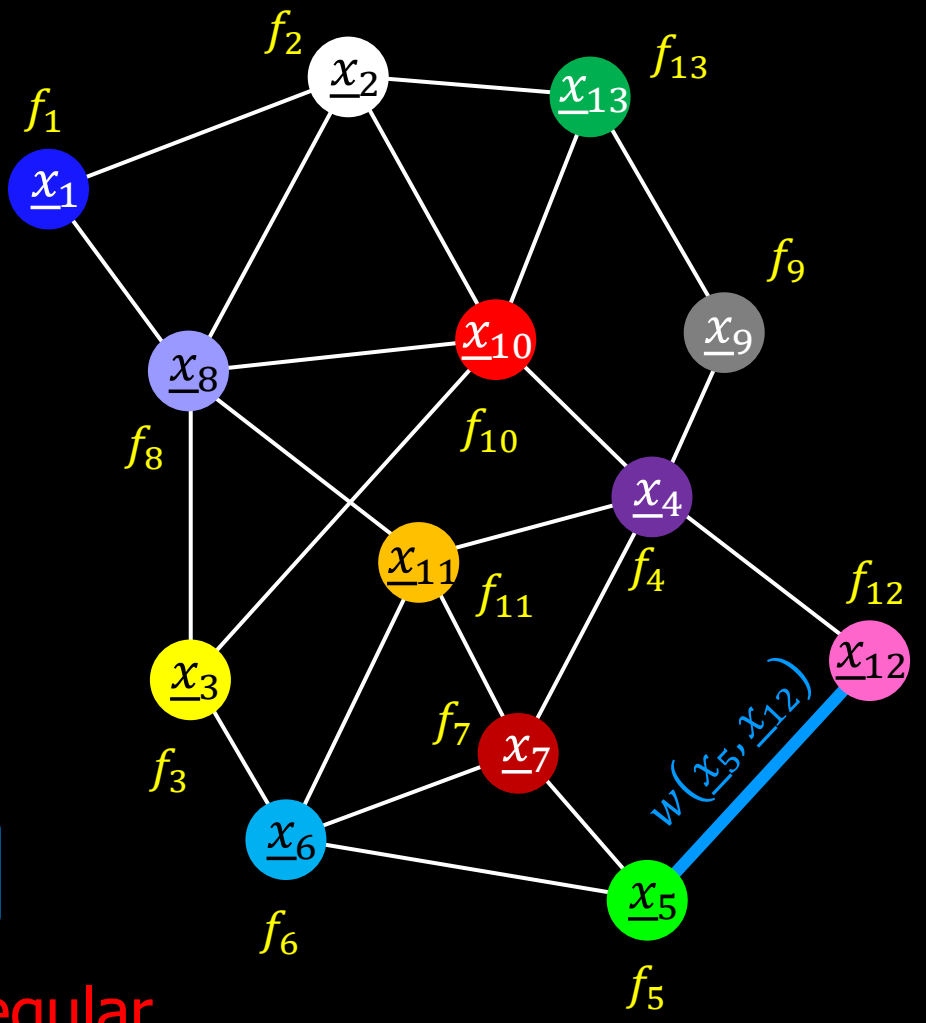
- I. Ram, M. Elad, and I. Cohen, "Generalized Tree-Based Wavelet Transform", IEEE Trans. Signal Processing, vol. 59, no. 9, pp. 4199–4209, 2011.
- I. Ram, M. Elad, and I. Cohen, "Redundant Wavelets on Graphs and High Dimensional Data Clouds", IEEE Signal Processing Letters, Vol. 19, No. 5, pp. 291–294 , May 2012.

Problem Formulation

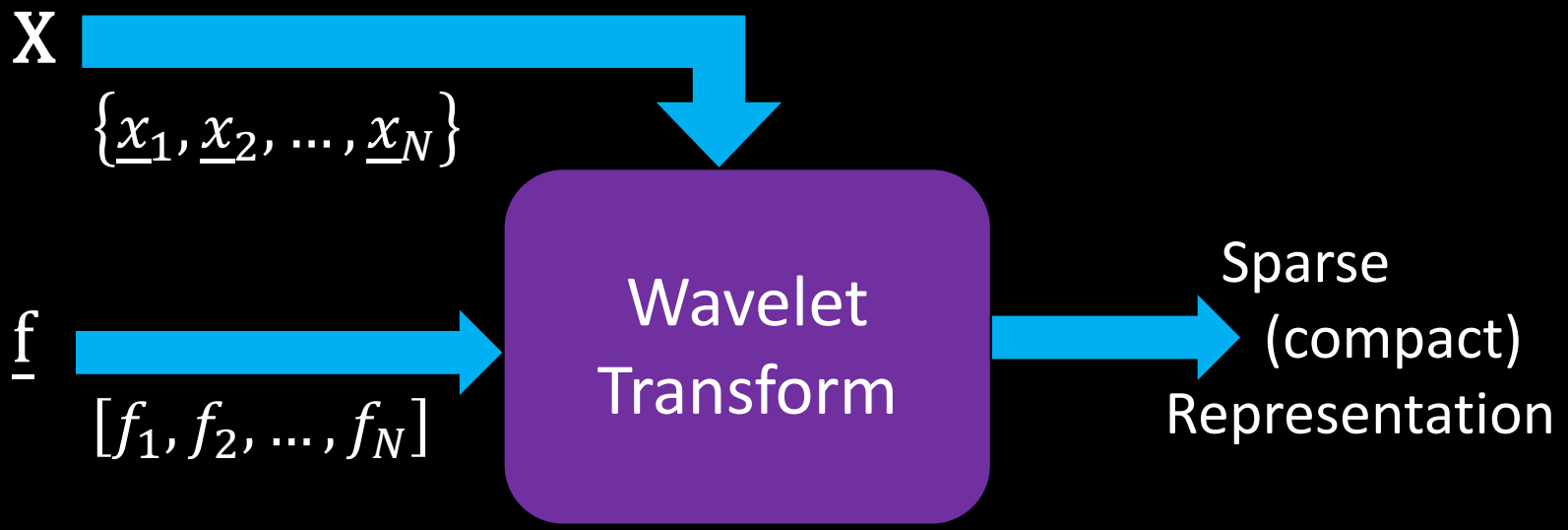
- $\mathbf{X}=\{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N\}$ - the data set, such that $\underline{x}_i \in \mathbb{R}^n$ may be
  - points in high dimension.
  - feature points associated with the nodes of a graph.
- $f: \mathbf{X} \to \mathbb{R}$ - a scalar function defined on the above coordinates, $f_i = f(\underline{x}_i)$
- Our key assumption: under a distance measure $w(\underline{x}_i, \underline{x}_j)$

$w(\underline{x}_i, \underline{x}_j)$ small $\to |f_i - f_j|$ small

for almost every pair (i, j) $\Rightarrow f$ is regular



Our Goal



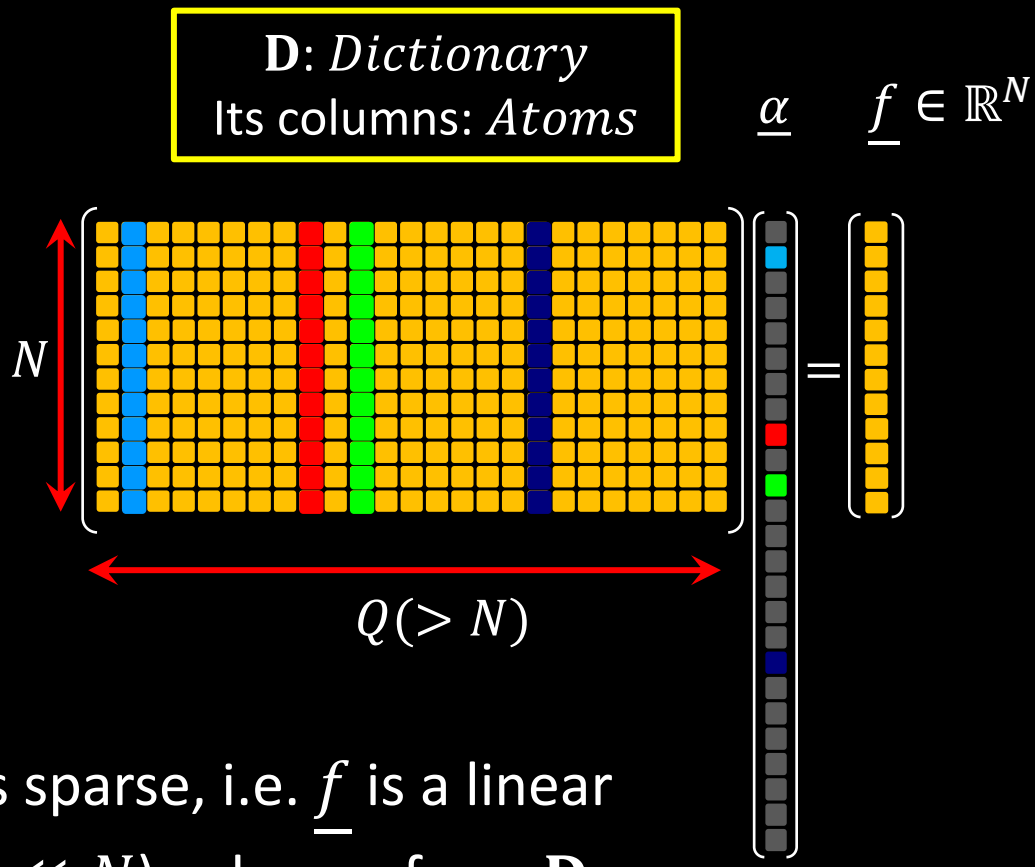
Why Wavelet?

- Wavelet is a highly effective "sparsifying transform" for regular piece-wise smooth signals.
- We would like to imitate this for our data structure.
- However, the signal (vector) $\underline{f}$ is not necessarily smooth in general.

Lets Talk About Sparsity

Sparsity: A different way to describe a signal's structure

$\mathbf{D}\underline{\alpha} = \underline{f}$



Model assumption: $\underline{\alpha}$ is sparse, i.e. $\underline{f}$ is a linear combination of FEW ($k \ll N$) columns from $\mathbf{D}$

Wavelet for Graphs – Previous Works

- "Diffusion Wavelets"

  R. R. Coifman, and M. Maggioni, 2006.

- "Multiscale Methods for Data on Graphs and Irregular Multidimensional Situations"

  M. Jansen, G. P. Nason, and B. W. Silverman, 2008.

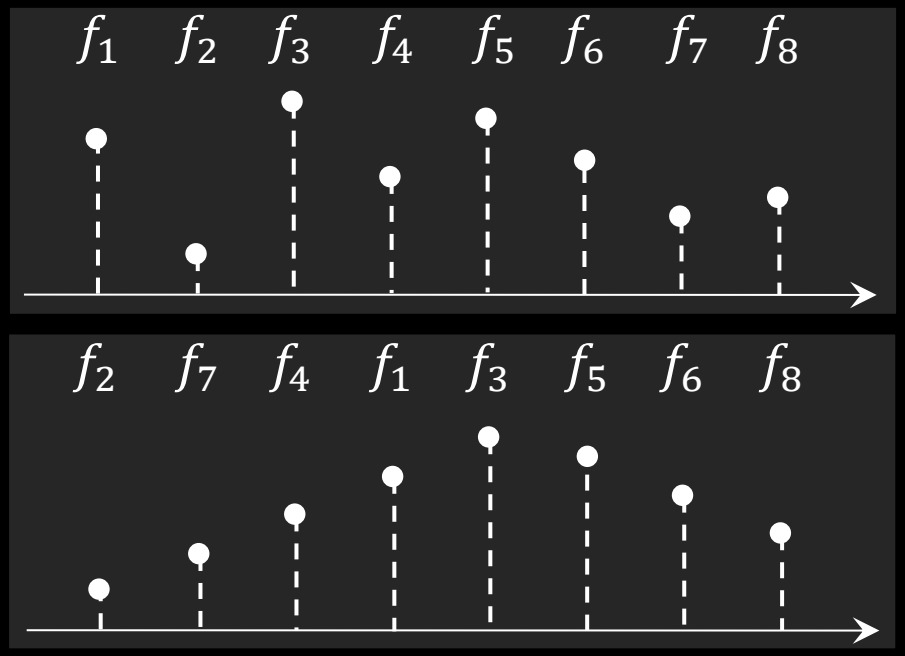
- "Wavelets on Graph via Spectral Graph Theory"

  D. K. Hammond, and P. Vandergheynst, and R. Gribonval, 2010.

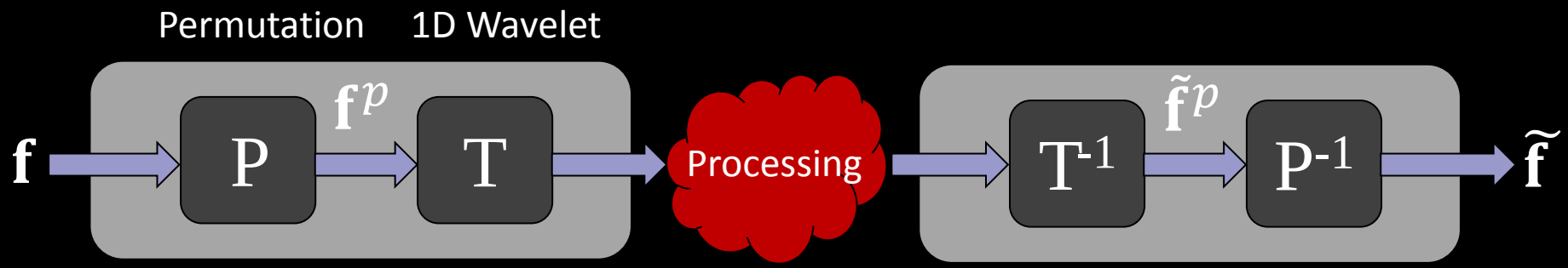
- "Multiscale Wavelets on Trees, Graphs and High Dimensional Data: Theory and Applications to Semi Supervised Learning"

  M . Gavish, and B. Nadler, and R. R. Coifman, 2010.

The Main Idea - Permutation

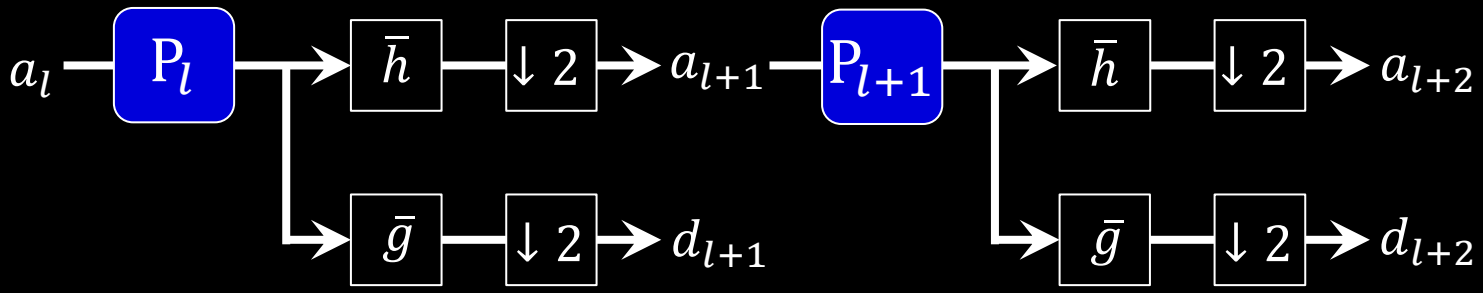


Permutation using
$\mathbf{X} = \{x_1, x_2, \dots, x_N\}$



The Main Idea – Permutation (cont.)

- In fact, we propose to perform a different permutation in each resolution level of the multi-scale pyramid:

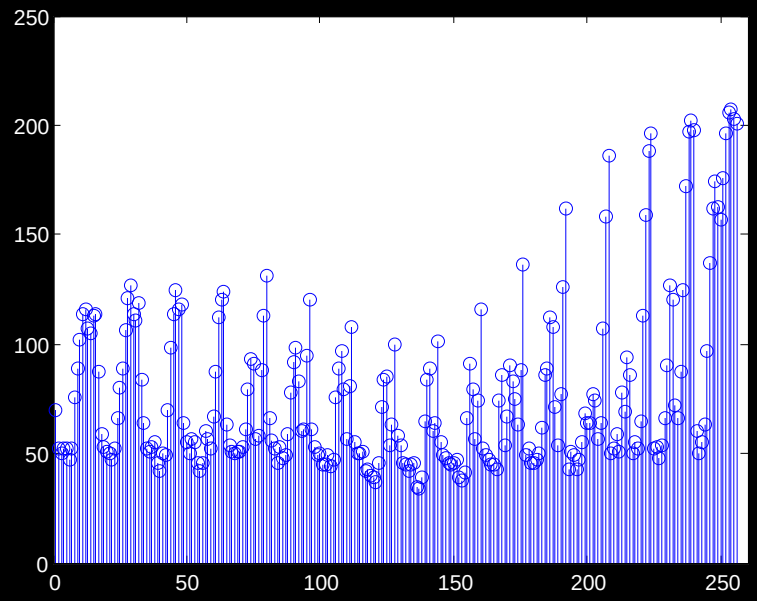


- Naturally, these permutations will be applied reversely in the inverse transform.
- The additional permutations make the difference between this and the plain 1D wavelet transform applied on $\mathbf{f}$.
- The transform adapts to the input signal while its linearity and unitarity are preserved.

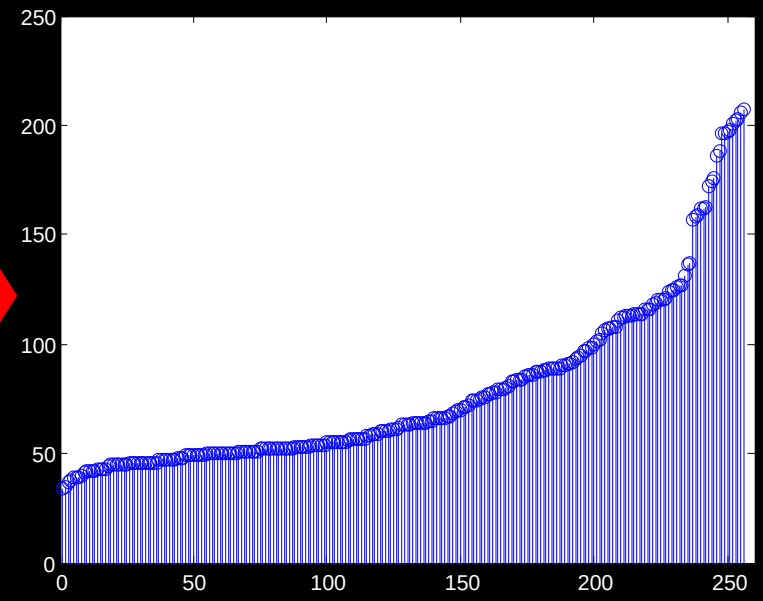
Building the Permutation $\mathbf{P}_0$

- We wish to design a permutation which produces a smooth signal when it is applied to the signal $\mathbf{f}$.
- So, … for example, we can simply permute by sorting the signal $\mathbf{f}$.

$\mathbf{f}$



$\mathbf{f}^p$



Building the Permutation $\mathbf{P}_0$ (cont.)

- **However**: we are interested in the case where $\mathbf{f}$ is corrupted (noisy, missing values, …) and thus such a sort operation is impossible.
- To our help come the feature vectors in $\mathbf{X}$, which reflect on the order of the signal values f_k. Recall:

Small $w(x_i, x_j)$ implies small $|f(x_i) - f(x_j)|$ for almost every pair (i, j)

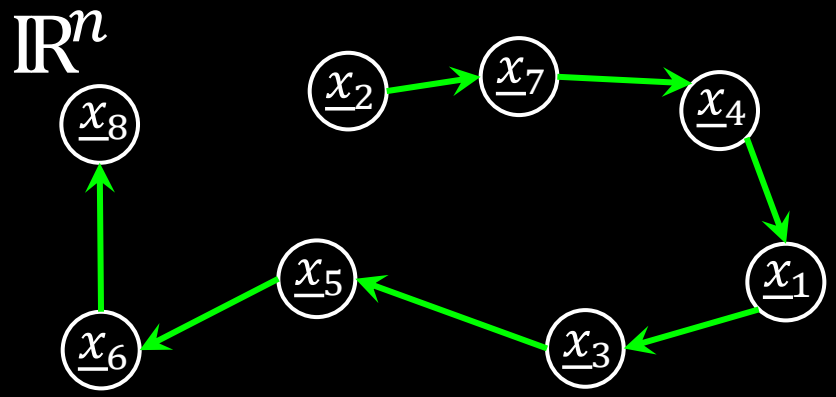
- Thus, instead of solving for the optimal permutation that "simplifies" $\mathbf{f}$, we order the features in $\mathbf{X}$ to the shortest path that visits each point once.

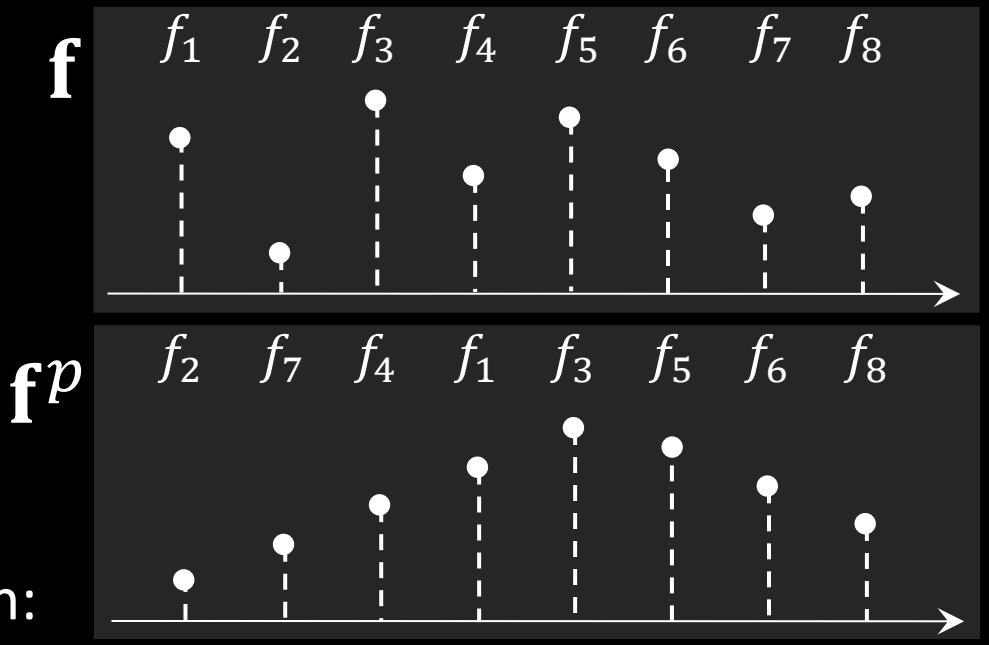
$$\min_{\mathrm{P}} \sum_{i=2}^{N} |f^p(i) - f^p(i-1)| \quad \Rightarrow \quad \min_{\mathrm{P}} \sum_{i=2}^{N} w(x_i^p, x_{i-1}^p)$$

Total variation of $\mathbf{f}$

An instance of the Traveling-Salesman-Problem (TSP)

Building the Permutation $\mathbf{P}_0$ (cont.)



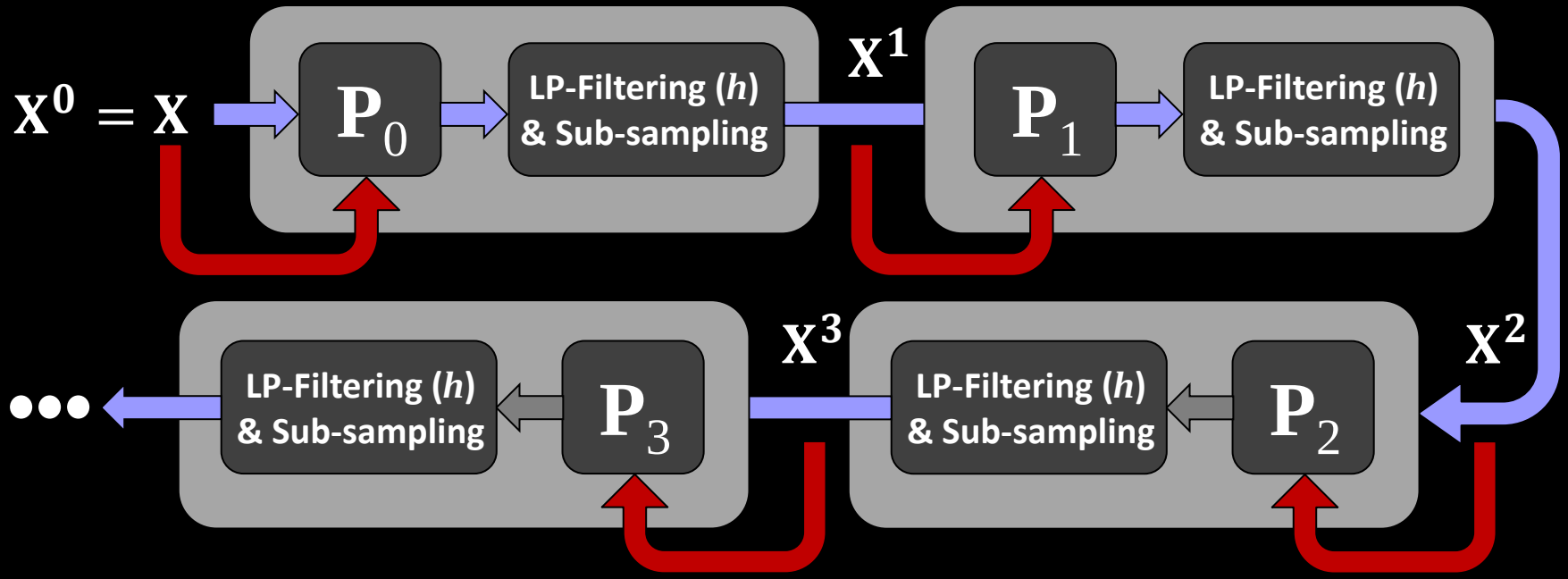


We handle the TSP task by a greedy (and crude) approximation:

- Initialize with an arbitrary index j;
- Initialize the set of chosen indices to $\Omega(1)=\{j\}$;
- Repeat $k=1{:}1{:}N\text{-}1$ times:
  - Find $\underline{x}_i$ – the nearest neighbor to $\underline{x}_{\Omega(k)}$ such that $i \notin \Omega$;
  - Set $\Omega(k+1)=\{i\}$;
- Result: the set Ω holds the proposed ordering.

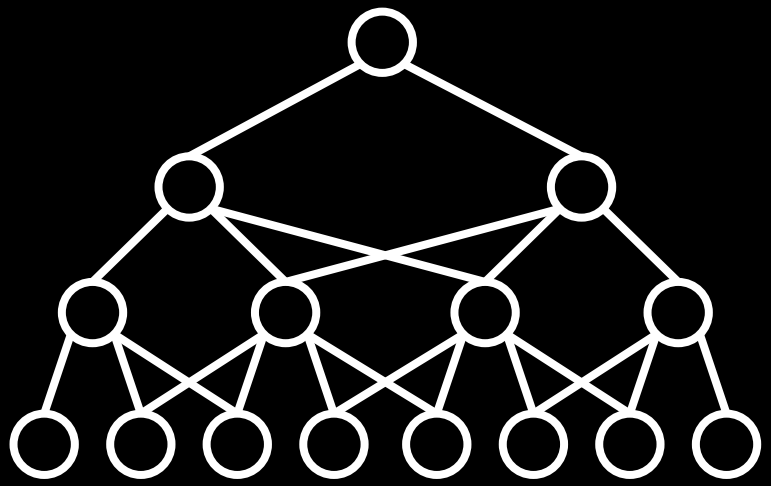
Building the Following Permutations

- In order to construct $P_1, P_2, \ldots, P_{L-1}$, the permutations at the other pyramid's levels, we use the same method, applied on propagated (reordered, filtered and sub-sampled) feature-vectors through the same wavelet pyramid:

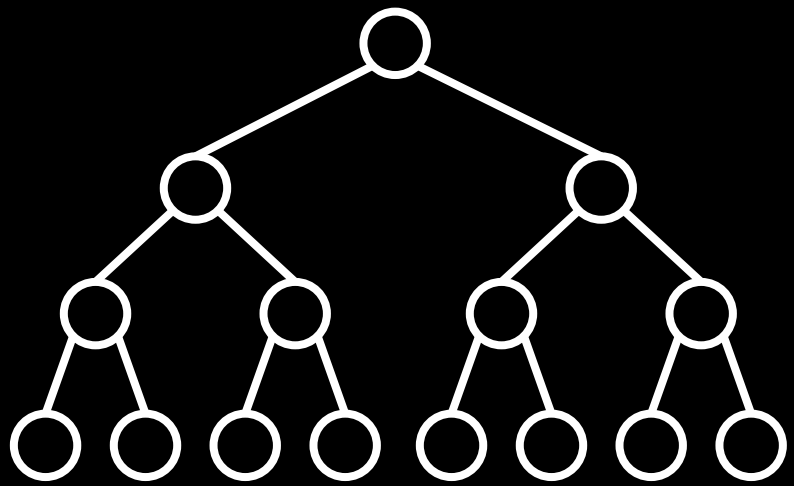


Generalized Tree

"Generalized" tree



Tree (Haar wavelet)

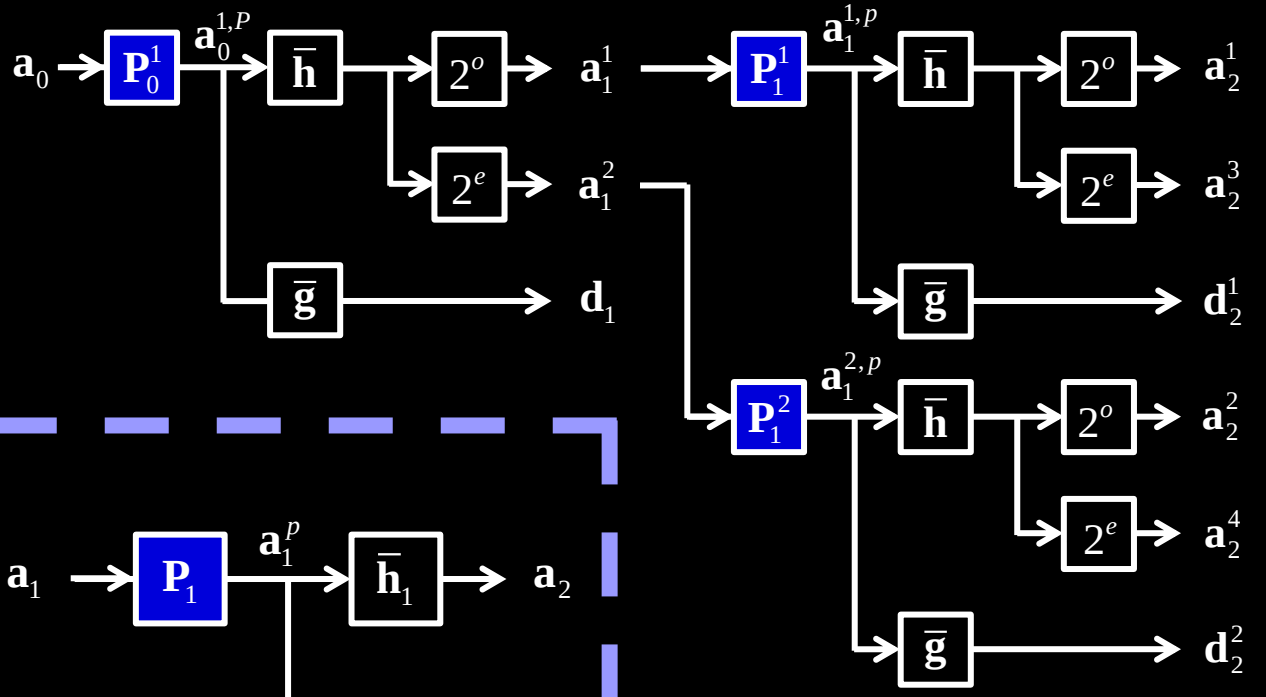


- The relation between the feature points in a full decomposition can be described using a tree-like structure.
- Our proposed transform: Generalized Tree-Based Wavelet Transform (GTBWT).

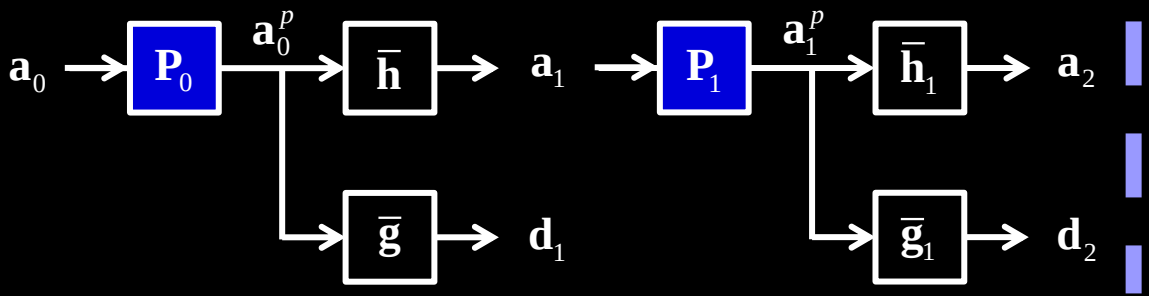
Redundant Representations

- A redundant representation may be obtained using several random variants of the GTBWT.
- A more elegant solution is to modify the redundant wavelet transform.

Redundant Tree-Based Wavelet Transform (RTBWT)



à trous algorithm



[Shensa, 1992] [Beylkin, 1992]

Image Processing using Tree-Based Wavelets

This part is taken from the following papers:

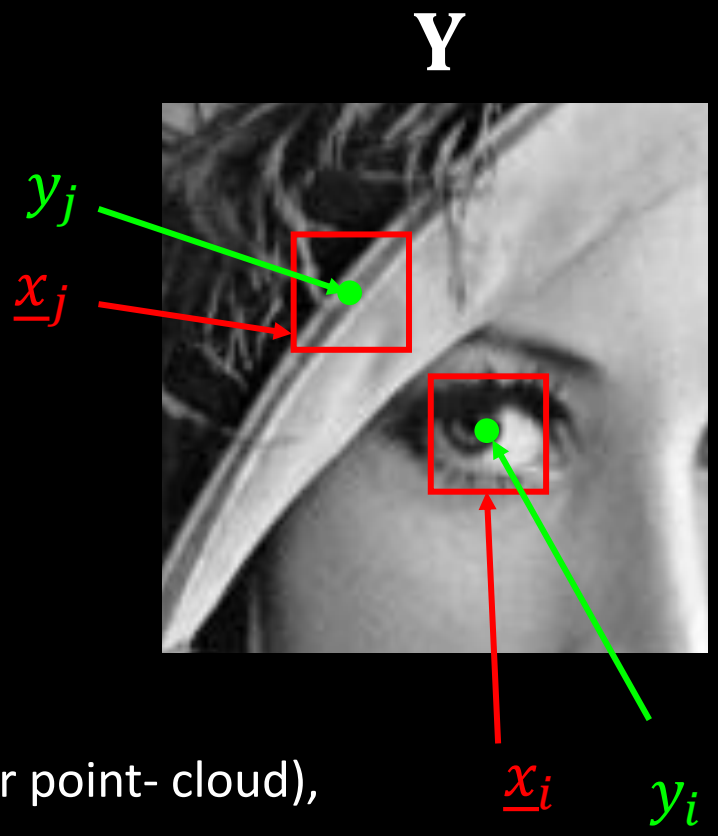
- I. Ram, M. Elad, and I. Cohen, "Generalized Tree-Based Wavelet Transform", IEEE Trans. Signal Processing, vol. 59, no. 9, pp. 4199–4209, 2011.
- I. Ram, M. Elad, and I. Cohen, "Patch-Ordering-Based Wavelet Frame and Its Use in Inverse Problems", IEEE Trans. on Image Processing, Vol. 23, No. 7, pp. 2779-2792, July 2014.
- I. Ram, I. Cohen, and M. Elad, "Facial Image Compression using Patch-Ordering-Based Adaptive Wavelet Transform", IEEE Signal Processing Letters, Vol. 21, No. 10, pp. 1270-1274, October 2014.

Turning an Image into a Graph

- $\mathbf{Y}$ – an image containing N pixels.
- $\mathbf{y}$ – column stacked version of $\mathbf{Y}$.
- y_i – i-th sample in $\mathbf{y}$.

We choose:

- $\underline{x}_i$ – column stacked version of an $\sqrt{n} \times \sqrt{n}$ patch around the location of y_i in $\mathbf{Y}$.
- $w(\underline{x}_i, \underline{x}_j)$ – the Euclidean distance $\|\underline{x}_i - \underline{x}_j\|$.

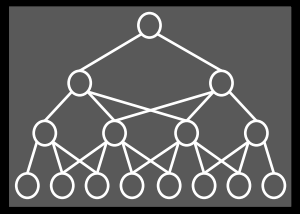


- Now, that the image is organized as a graph (or point- cloud), we can apply the developed transforms.
- After this "conversion", we forget about spatial proximities.
- The overall scheme becomes "yet another" patch-based image processing algorithm

M-Term Approximation

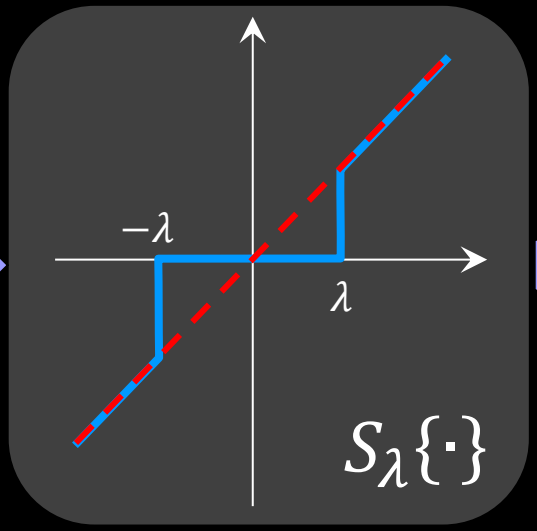
Original Image

Multiply by $\mathbf{T}$:
Forward GTBWT



$\mathbf{y}$

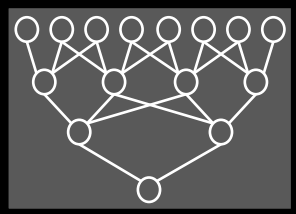
$\mathbf{Ty}$



$S_\lambda\{\mathbf{Ty}\}$

M non-zeros

Multiply by $\mathbf{T}^{-1}$:
Inverse GTBWT



Output image

$\hat{\mathbf{y}}$

Show

$$\|\mathbf{y} - \hat{\mathbf{y}}\|^2 = \|\mathbf{y} - \mathbf{T}^{-1} S_\lambda\{\mathbf{Ty}\}\|^2$$

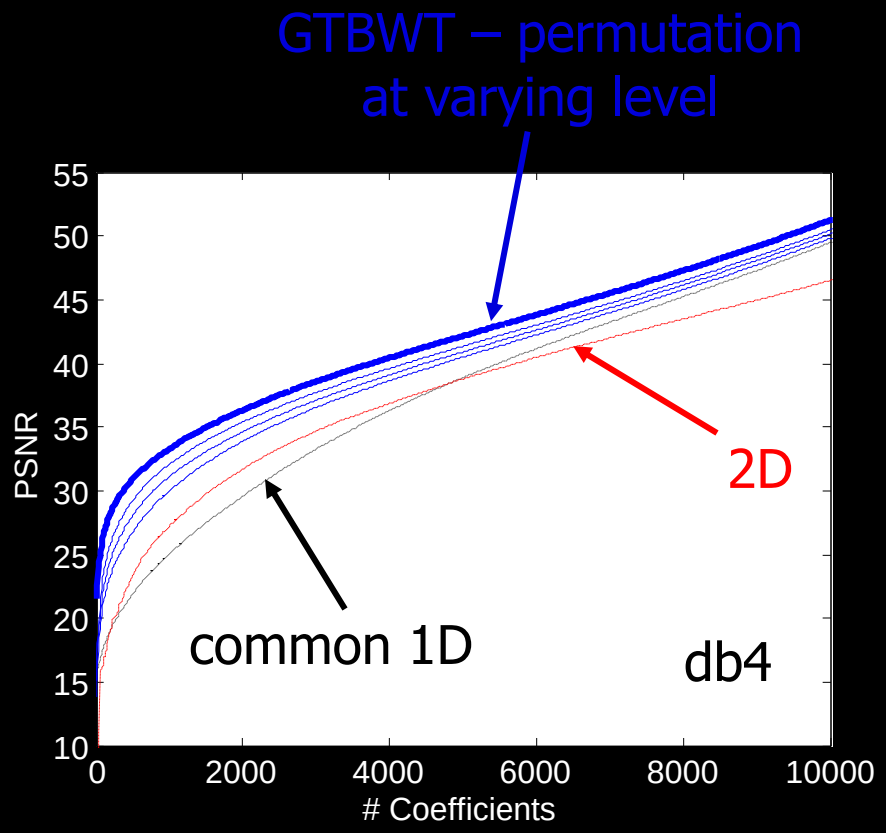
as a function of M

M-Term Approximation (cont.)

For a 128×128 center portion of the image Lena

we compare the image representation efficiency of the

- GTBWT
- A common 1D wavelet transform
- 2D wavelet transform



The Representation Basis Functions

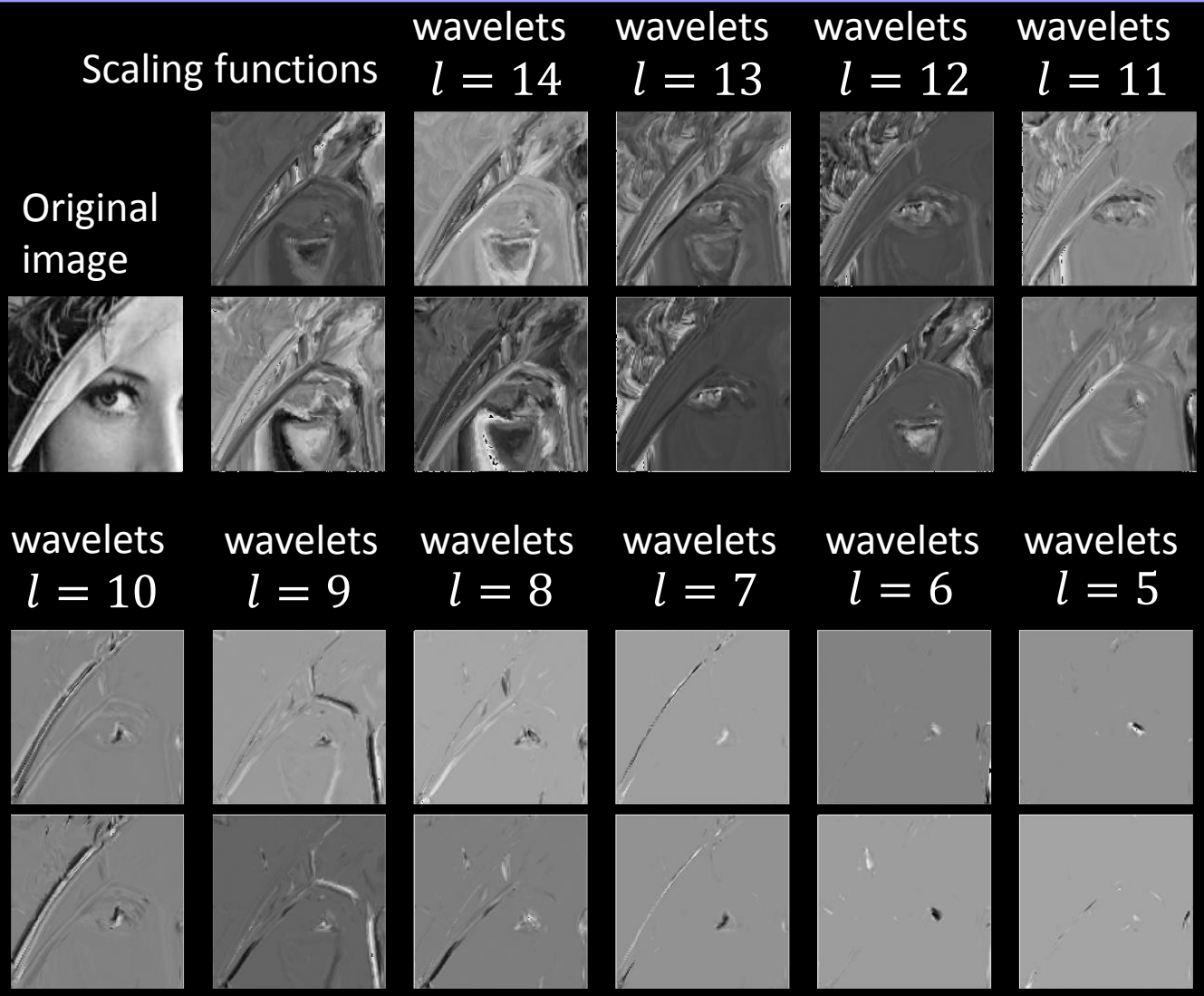
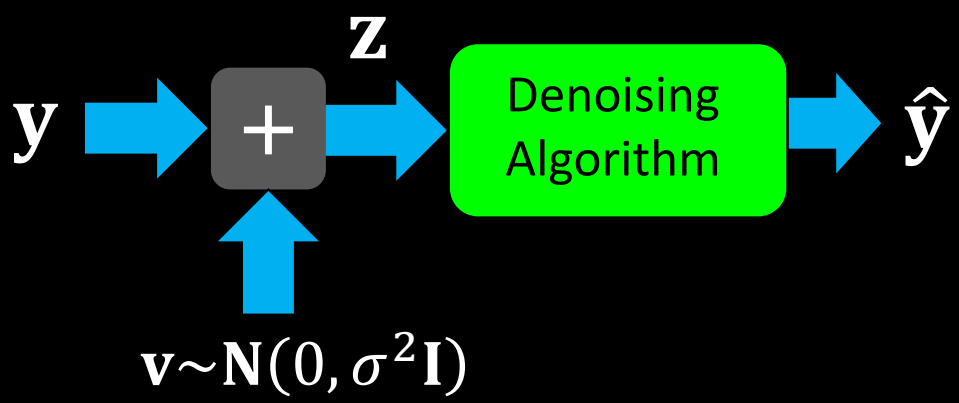


Image Denoising – The Basic Scheme



Approximation by the THR algorithm:

$$\hat{\mathbf{y}} = \mathbf{T}^{-1} S_\lambda\{\mathbf{T}\mathbf{z}\}$$

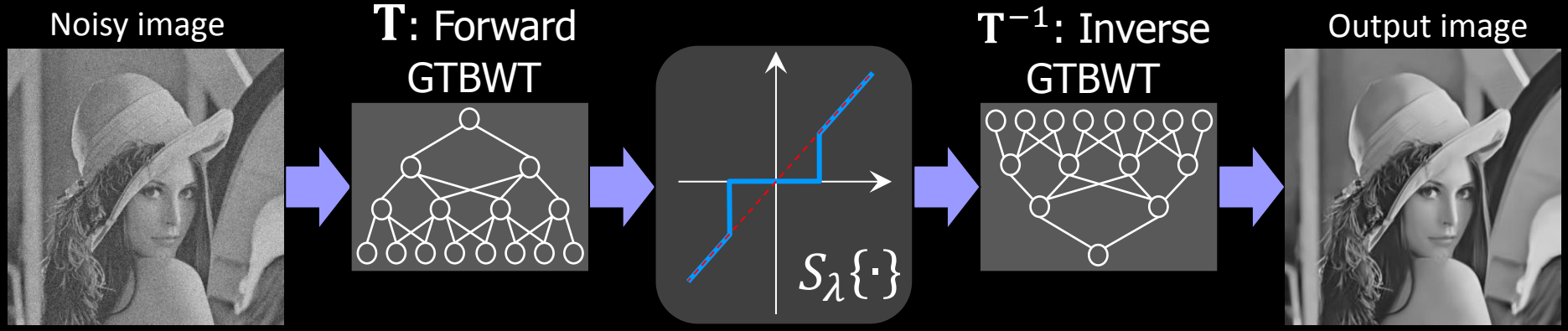
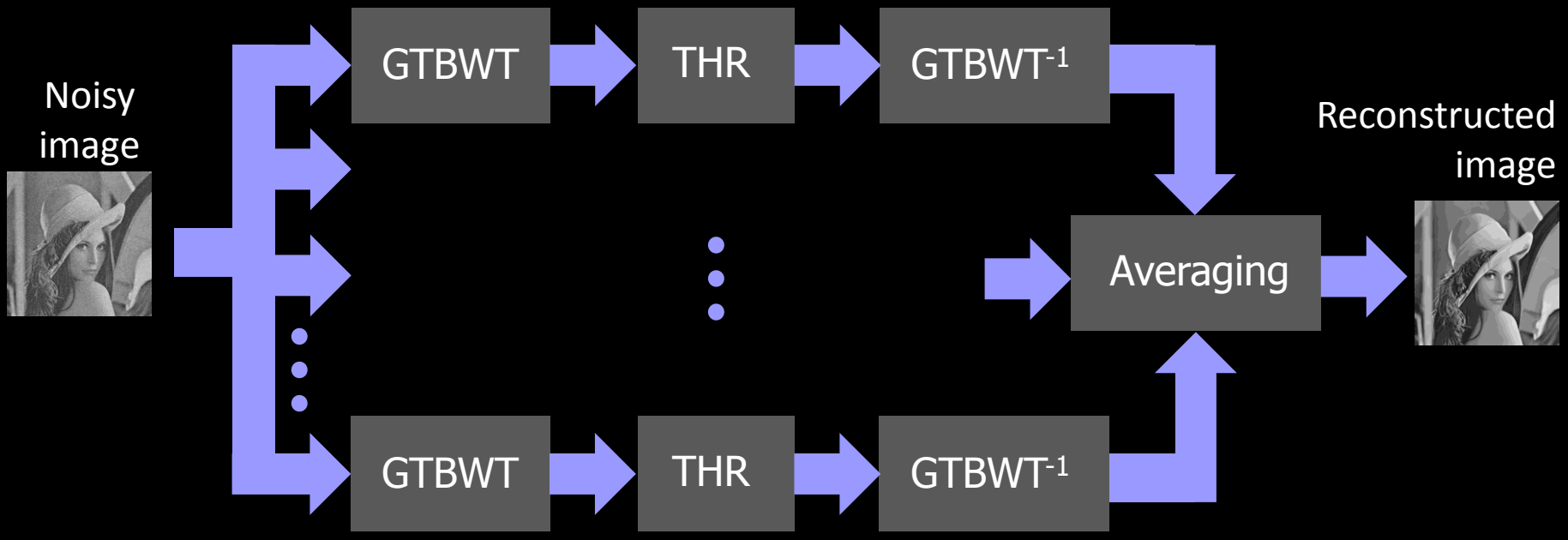


Image Denoising - Improvements

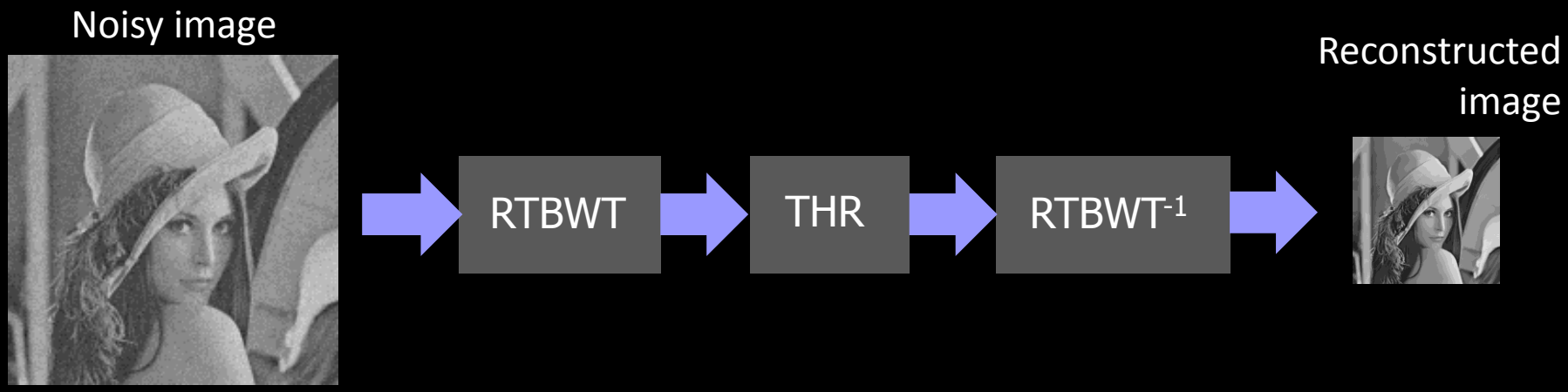
Cycle-spinning: Apply the above scheme several (10) times, with different random variations of the GTBWT, and average.



Equivalent to using a redundant transform

Image Denoising – Improvements (cont.)

Alternatively, we can simply apply our proposed Redundant Tree-Based Wavelet Transform.



- Preliminary tests showed that the GTBWT and RTBWT perform similarly.

Image Denoising – Improvements (cont.)

Sub-image averaging : Apply the above scheme to n sub-images and reconstruct the image from the obtained results.

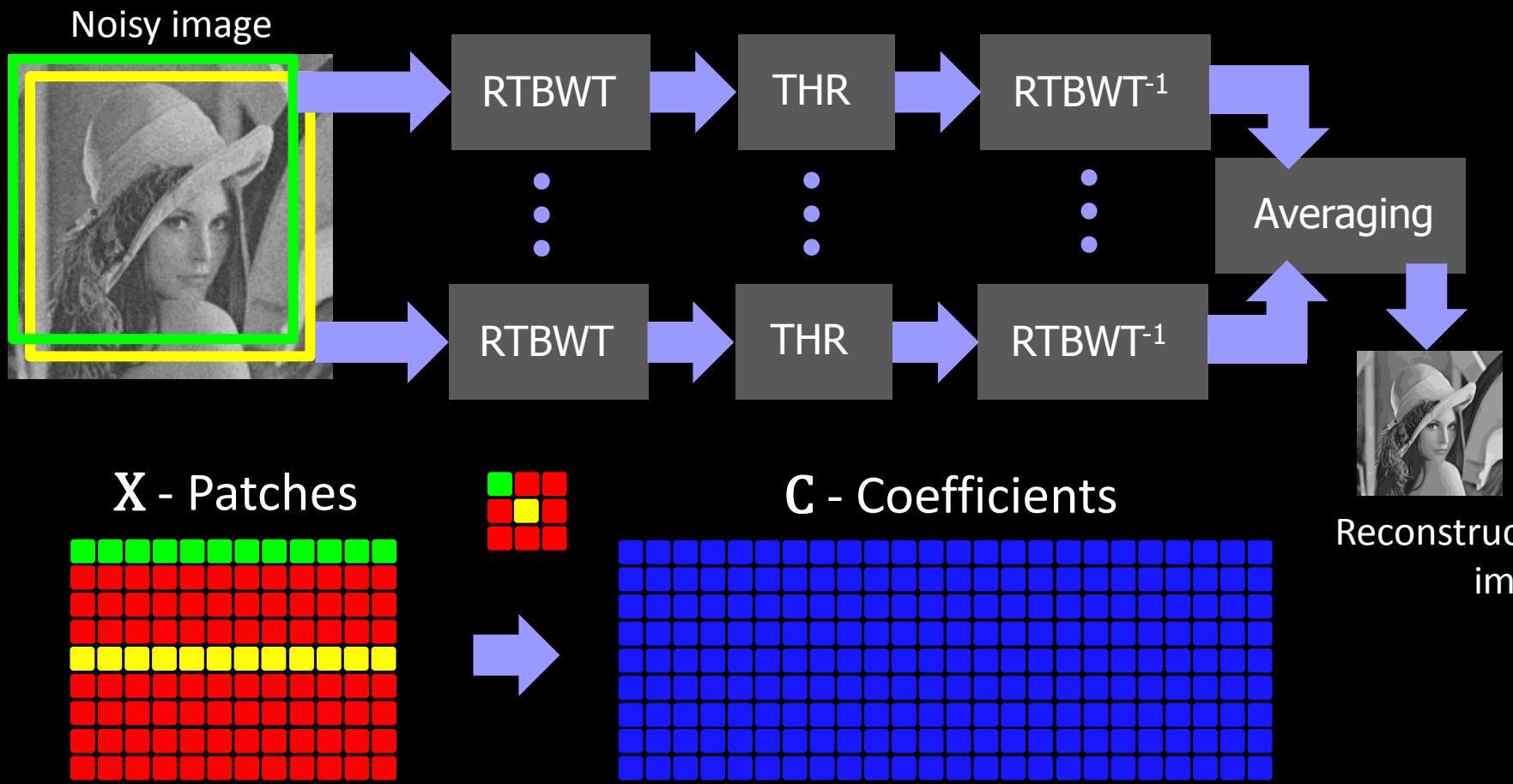


Image Denoising – Improvements (cont.)

Improved thresholding: thresholding the wavelet coefficients based on the norm of the (transformed) vector they belong to:

- Classical thresholding: every coefficient within **C** is passed through the function:

$$c_{i,j} = \begin{cases} c_{i,j} & \left|c_{i,j}\right| \geq T \\ 0 & \left|c_{i,j}\right| < T \end{cases}$$

- Instead we propose to force "joint-sparsity" on the above array of coefficients, forcing all rows to share the same support:

$$c_{i,j} = \begin{cases} c_{i,j} & \left\|c_{*,j}\right\|_2 \geq T \\ 0 & \left\|c_{*,j}\right\|_2 < T \end{cases}$$

C - Coefficients

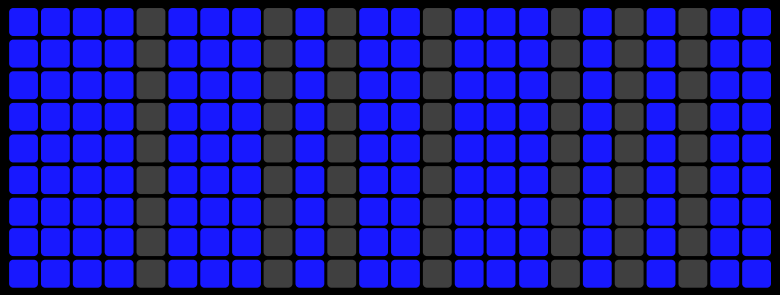


Image Denoising – Improvements (cont.)

Restricting the NN: When searching the nearest-neighbor for the ordering, restriction to near-by area is helpful, both computationally and in terms of the output quality.

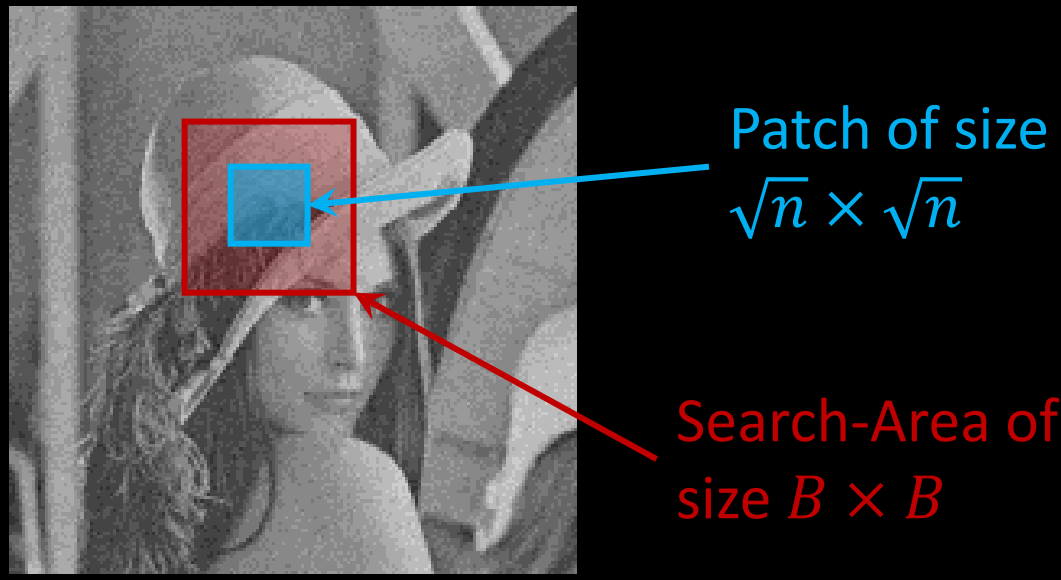


Image Denoising – Results

- We explore the results obtained with two iterations of our scheme.
- The RTBWT PSNR results are good and competitive, especially for $\sigma \geq 25$.

Image		**σ/PSNR [dB]**		
		10 / 28.14	25 / 20.18	50 / 14.16
Lena	BM3D	35.93	32.08	29.05
	1 iteration	35.70	32.06	28.97
	2 iterations	35.45	32.11	29.18
	1 iteration + Wiener	35.75	32.26	29.30
Barbara	BM3D	34.98	30.72	27.23
	1 iteration	34.50	30.73	27.39
	2 iterations	34.55	30.76	27.65
	1 iteration + Wiener	34.55	30.90	27.78
House	BM3D	36.71	32.86	29.69
	1 iteration	36.41	32.74	29.55
	2 iterations	35.68	32.46	29.49
	1 iteration + Wiener	35.86	32.37	29.56

Image Deblurring

$\mathbf{y}$

Lexicographic ordering of the N pixels

$\mathbf{X}$: Array of overlapped patches of size nN

Applying a J redundant tree-based wavelet transform

We obtain an array of nNJ transform coefficients

- All these operations could be described as one linear operation: multiplication of $\mathbf{y}$ by a huge matrix $\boldsymbol{\Omega}$.
- This transform is adaptive to the specific image.

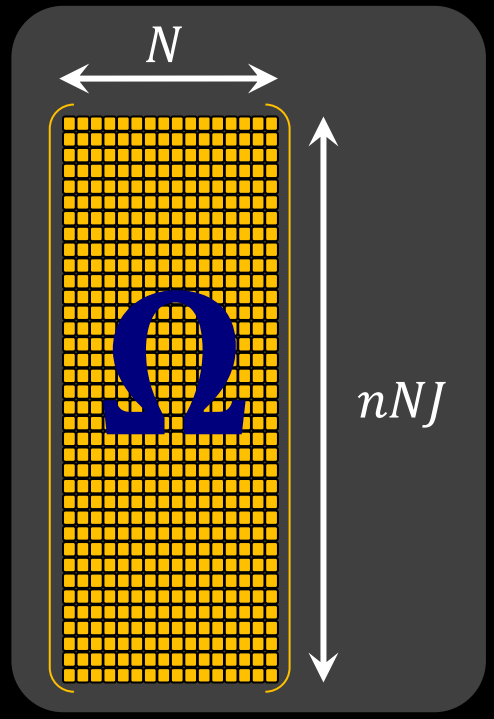
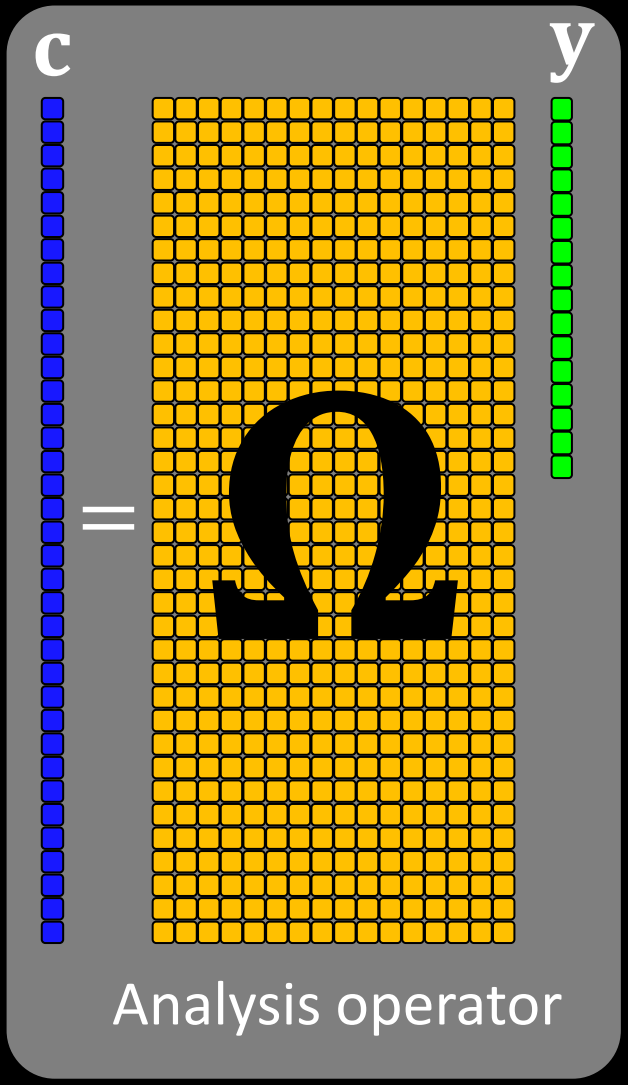
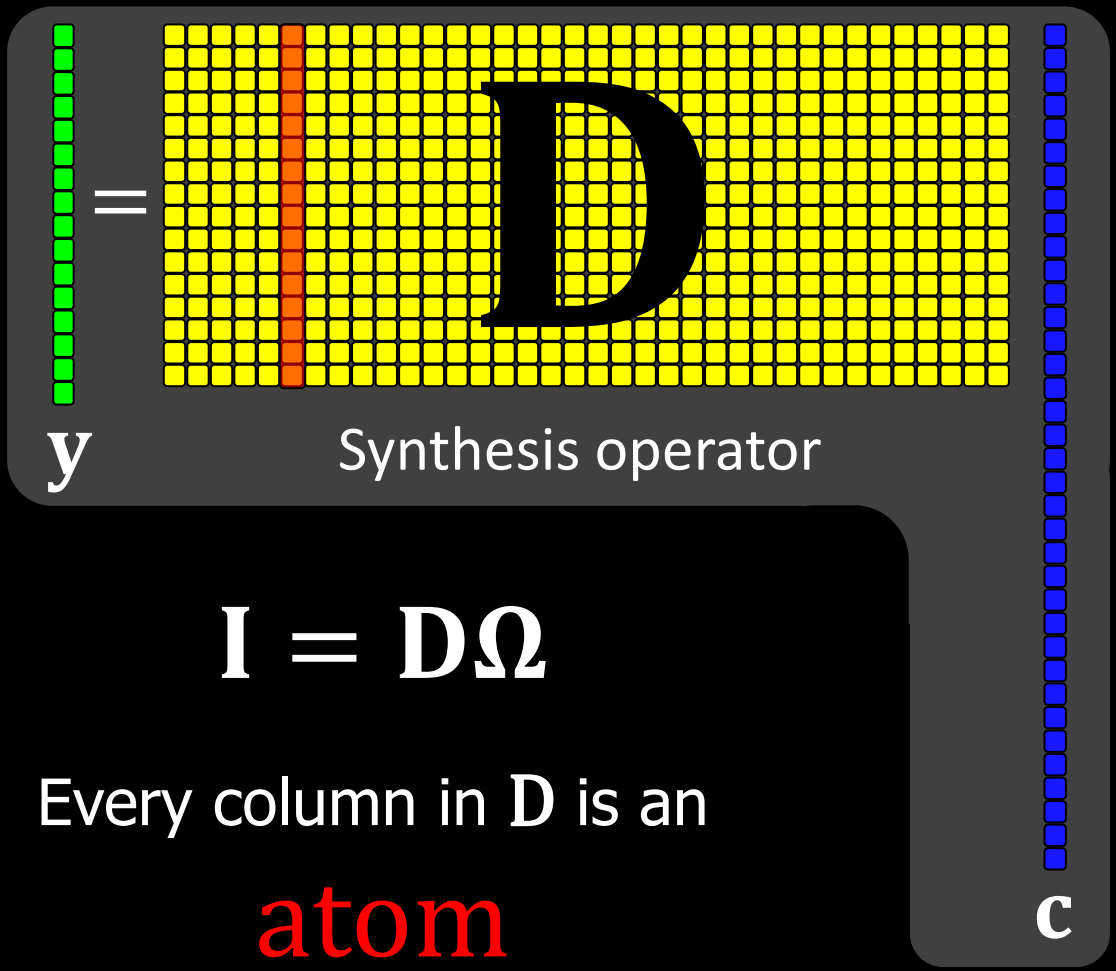


Image Deblurring (cont.)





$$\mathbf{I} = \mathbf{D}\boldsymbol{\Omega}$$

Every column in $\mathbf{D}$ is an

atom

Image Deblurring (cont.)

We can use these operators to solve various inverse problems of the form:

$$\mathbf{z} = \mathbf{H}\mathbf{y} + \mathbf{v}$$

where: $\mathbf{y}$ is the original image
$\mathbf{v}$ is an AWGN, and
$\mathbf{H}$ is a degradation operator of any sort

We could consider solving a synthesis or analysis problems, or their combination:

$$\{\hat{\mathbf{y}}, \hat{\mathbf{c}}\} = \underset{\mathbf{y},\mathbf{c}}{\text{Argmin}} \; \|\mathbf{z} - \mathbf{H}\mathbf{y}\|_2^2 + \eta\|\mathbf{y} - \mathbf{D}\mathbf{c}\|_2^2 + \lambda\|\mathbf{c}\|_p^p + \mu\|\mathbf{c} - \boldsymbol{\Omega}\mathbf{y}\|_2^2$$

$\eta \to \infty$, $\mu = 0$ $\Rightarrow$ Synthesis

$\eta = 0$, $\mu \to \infty$ $\Rightarrow$ Analysis

Generalized Nash Equilibrium*

Instead of minimizing the joint analysis/synthesis problem:

$$\{\hat{\mathbf{y}}, \hat{\mathbf{c}}\} = \underset{\mathbf{y},\mathbf{c}}{\text{Argmin}} \ \|\mathbf{z} - \mathbf{H}\mathbf{y}\|_2^2 + \eta\|\mathbf{y} - \mathbf{D}\mathbf{c}\|_2^2 + \lambda\|\mathbf{c}\|_p^p + \mu\|\mathbf{c} - \mathbf{\Omega}\mathbf{y}\|_2^2$$

we handle it as a generalized Nash equilibrium process, and break it down into two separate and easy to handle parts:

and solve iteratively

Inversion: $\mathbf{y}_{k+1} = \underset{\mathbf{y}}{\text{Argmin}} \ \|\mathbf{z} - \mathbf{H}\mathbf{y}\|_2^2 + \eta\|\mathbf{y} - \mathbf{D}\mathbf{c}\|_2^2$

Denoising: $\mathbf{c}_{k+1} = \underset{\mathbf{c}}{\text{Argmin}} \ \lambda\|\mathbf{c}\|_p^p + \mu\|\mathbf{c} - \mathbf{\Omega}\mathbf{y}\|_2^2$

* Danielyan, Katkovnik, and Eigiazarian, "BM3D frames and Variational Image Deblurring", IEEE Trans. on Image Processing, Vol. 21, No. 4, pp. 1715-1728, April 2012.

Image Deblurring - Results

Image	BM3D-DEB ISNR	IDD-BM3D ISNR init. with BM3D-DEB	Ours ISNR Init. with BM3D-DEB	Ours ISNR 1 iteration with simple initialization	Ours ISNR 3 iterations with simple initialization
Lena	6.55	6.61	6.92	5.99	6.34
Barbara	4.13	3.96	4.57	2.54	2.76
House	8.22	8.55	8.79	7.58	8.01
Cameraman	6.46	7.12	7.38	6.18	6.69

Blur PSF = $\frac{1}{1 + i^2 + j^2} \quad -7 \leq i, j \leq 7$

$\sigma^2=8$

Image Deblurring - Results

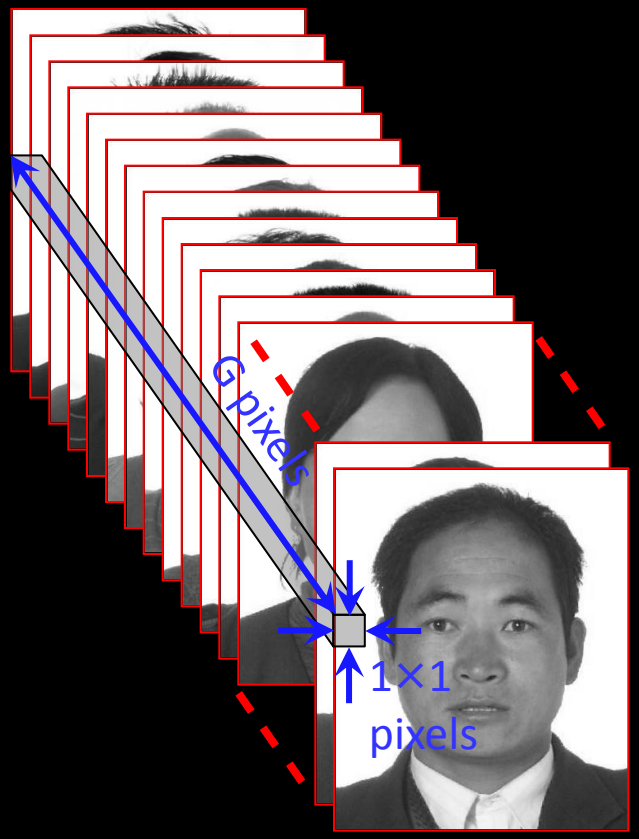
Original

Blurred+Noisy

Restored

Image Compression

- The problem: Compressing photo-ID images.
- General purpose methods (JPEG, JPEG2000) do not take into account the specific family.
- By adapting to the image-content (e.g. adaptive transform), better results could be obtained.
- We perform Geometric alignment of the images [Goldenberg, Kimmel, & E. ('05)] as it is very helpful.
- For our technique to operate well, we find the best common Redundant Tree-Based Wavelet Transform fitting a training set of facial images.
- Our pixel ordering is therefore designed on patches of size 1×1×G pixels from the training volume.



Compression by Pixel-Ordering

Detect main features and warp the images (20 bytes)

Compute the mean image and subtract it

Find the common RTBWT and calculate Huffman Tables for entropy coding

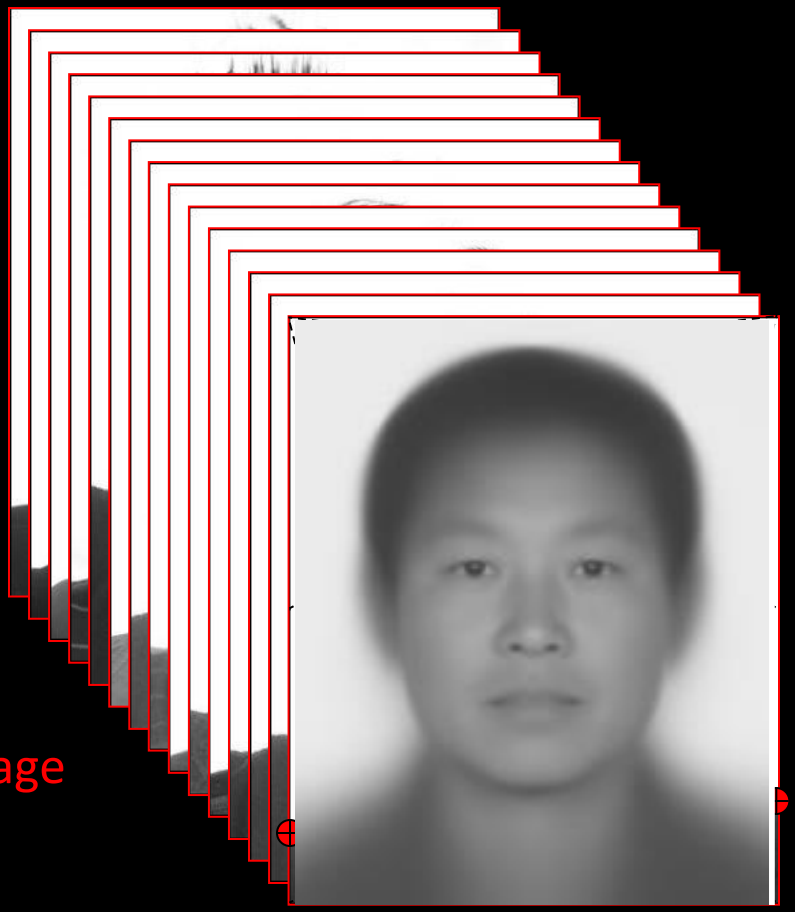
On the training set

Warp, remove the mean, and apply sparse coding using the RTBWT

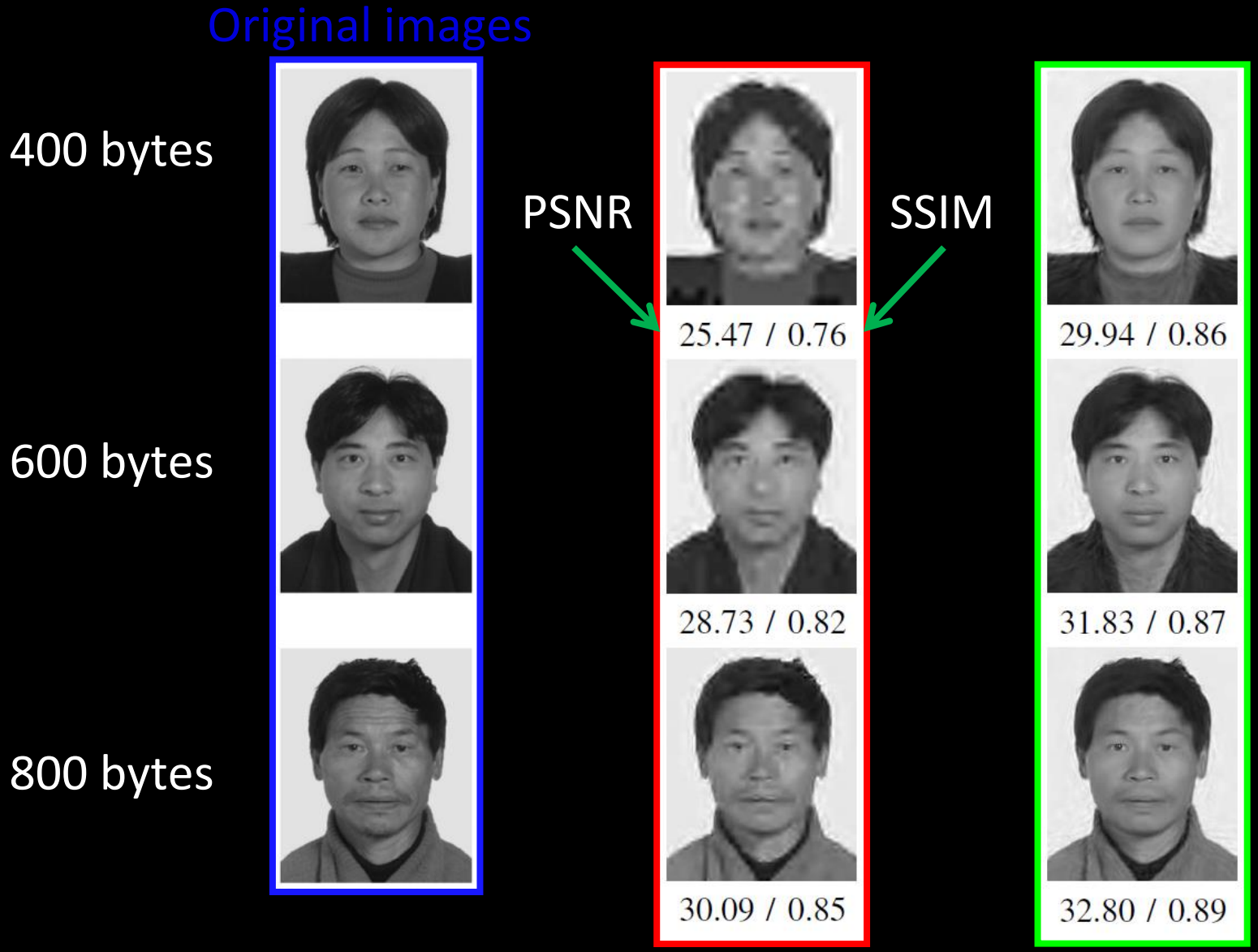
Apply quantization and entropy coding

On the test image

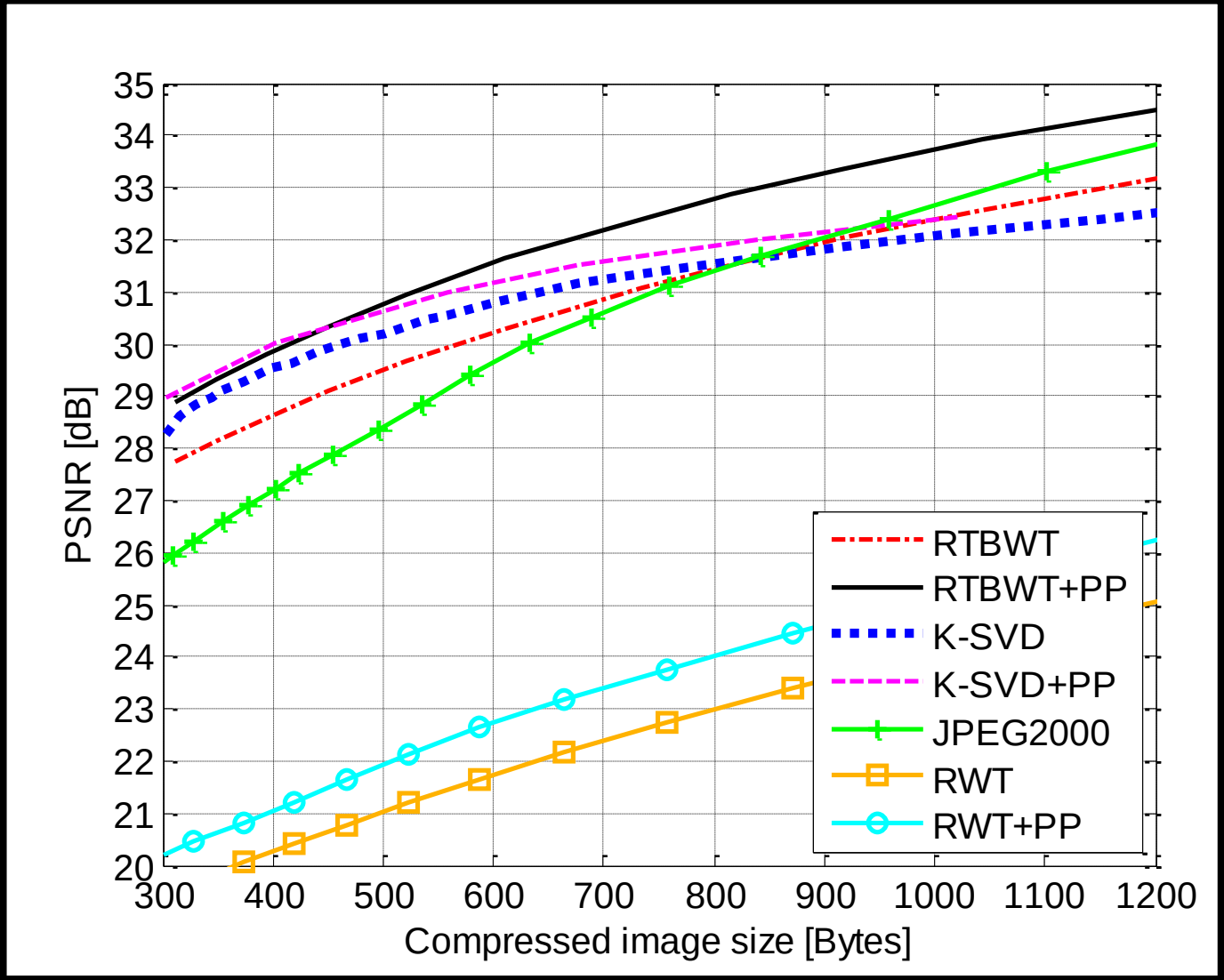
Training set (4415 images)



Face Compression - Results



Rate-Distortion Curves



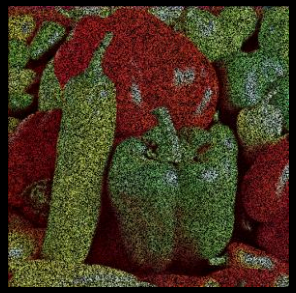
[Bryt & Elad, 2008]

Image Processing using Smooth Patch Ordering

This part is based on the paper:

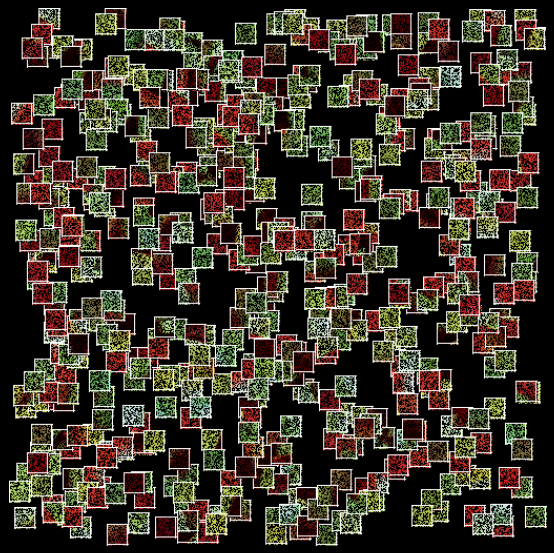
- I. Ram, M. Elad, and I. Cohen, "Image Processing using Smooth Ordering of its Patches", IEEE Transactions on Image Processing, Vol. 22, No. 7, pp. 2764–2774 , July 2013.

Returning to the Basics

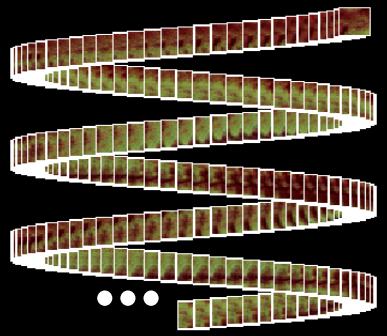


Suppose we start with a corrupted image

We extract all patches with overlaps



Then we order these patches to form the shortest path, as before

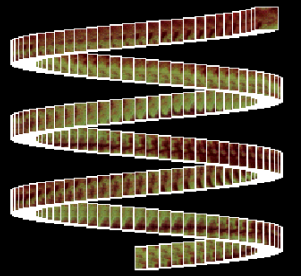


This reordering induces a permutation on the image pixels

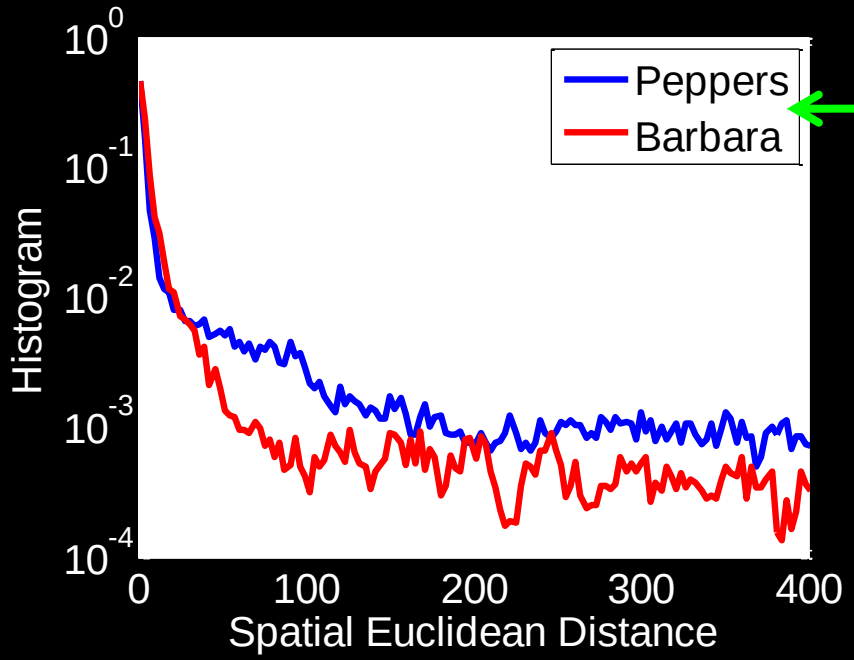
What should we expect from this permutation?

Spatial Neighbor ≠ Euclidean Neighbor

What should we expect?



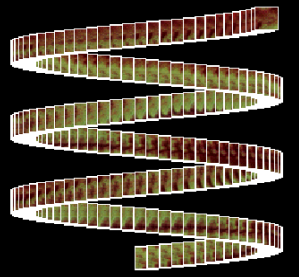
Spatial neighbors are not necessarily expected to remain neighbors in the new ordering



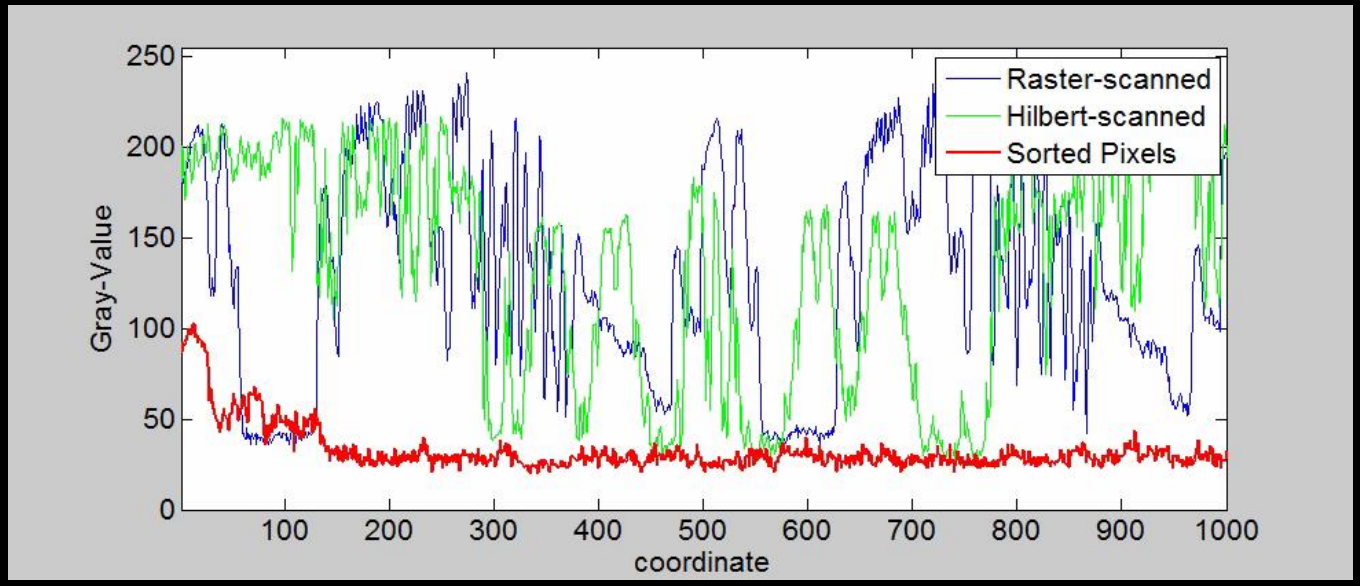
Noisy images with σ=10

The Reordered Signal is More Regular

What should we expect?



- The ordering defined by the new path is expected to
  - be robust to noise and degradations.
  - lead to a smooth (or at least, piece-wise smooth) 1D version of the target signal.*



* Measure of smoothness:

$$\frac{1}{L}\sum_{k=2}^{L}|\mathrm{y}[k]-\mathrm{y}[k-1]|$$

1. Raster scan: 9.57
2. Hilbert curve: 11.77
3. Sorted (ours): 5.63

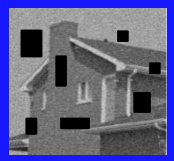
Processing the Permuted Pixels

- Given a corrupted image of the form:

$$\mathbf{z} = \mathbf{M}\mathbf{y} + \mathbf{v}$$

where: $\mathbf{y}$ is the original image
$\mathbf{v}$ is an AWGN, and
$\mathbf{M}$ is a point-wise degradation operator

- We reorder the pixels of the corrupted image using the permutation calculated from its patches.
- We can take advantage of our prior knowledge that the reordered target image should be smooth, and apply the following process



Corrupted Image $\tilde{\mathbf{f}}$

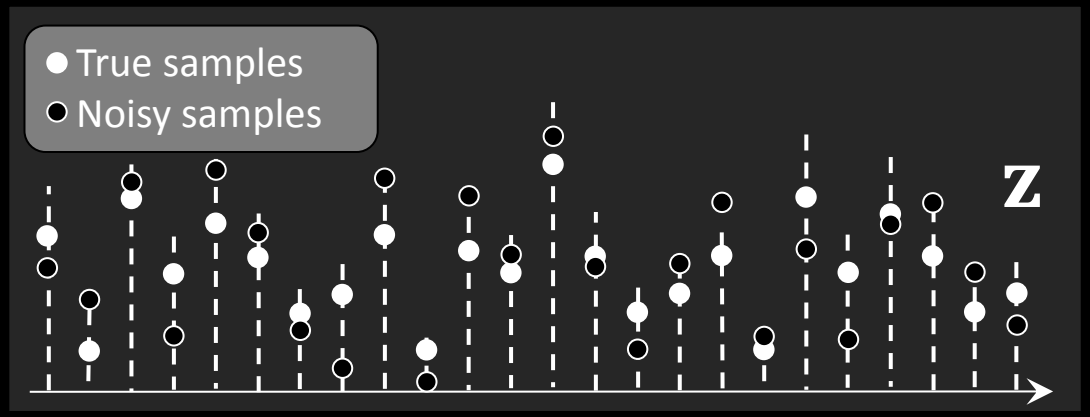
Re-order the pixels to a 1D signal

Apply the 1D signal a smoothing operator

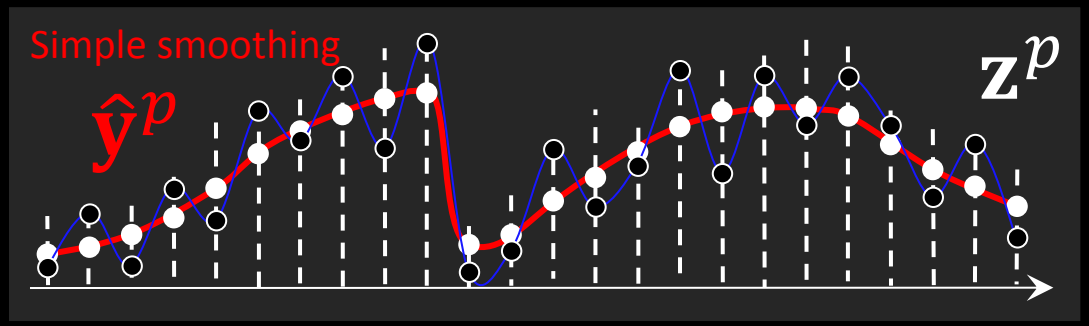
Re-order the pixels back to their location

Image Denoising

Noisy with σ=25 (20.18dB)



Ordering based on the noisy patches



Reconstruction: 32.65dB *

* This result is obtained with (i) cycle-spinning, (ii) sub-image averaging, (iii) two iterations, (iv) learning the filter , and (v) switched smoothing.

The “Simple Smoothing” We Do

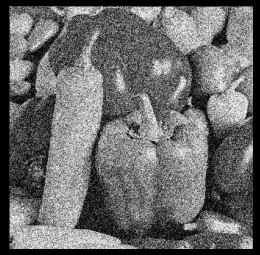
Simple smoothing works fine

but

We can do better by a training phase

Training image

Original image



Noisy image

Apply the permutation on the pixels

Compute the TSP permutation

Apply the permutation on the pixels

Apply a 1D filter h

+

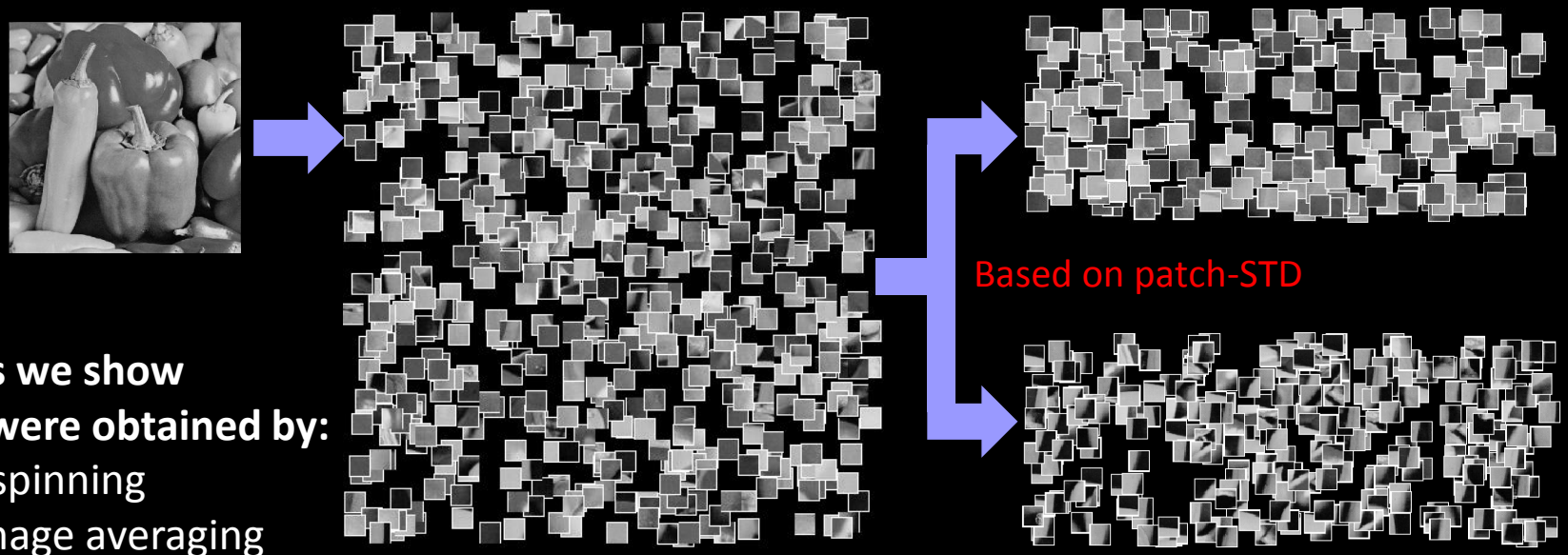
-

optimize h to minimize the reconstruction MSE

Naturally, this is done off-line and on other images

Filtering – A Further Improvement

Cluster the patches to smooth and textured sets, and train a filter per each separately



The results we show hereafter were obtained by:

(i) Cycle-spinning
(ii) Sub-image averaging
(iii) Two iterations
(iv) Learning the filter , and
(v) Switched smoothing.

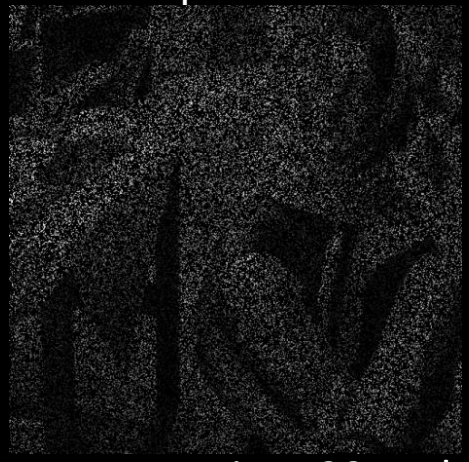
Denoising Results Using Patch-Reordering

Image		σ/PSNR [dB]		
		10 / 28.14	25 / 20.18	50 / 14.16
Lena	K-SVD	35.52	31.35	27.85
	BM3D	35.93	32.05	28.96
	1st iteration	35.26	31.55	28.66
	2nd iteration	35.39	31.80	28.96
Barbara	K-SVD	34.40	29.54	25.43
	BM3D	34.93	30.61	27.16
	1st iteration	34.29	30.36	27.19
	2nd iteration	34.39	30.47	27.35
House	K-SVD	35.90	31.97	28.01
	BM3D	36.63	32.79	29.54
	1st iteration	35.61	32.34	29.28
	2nd iteration	35.80	32.54	29.64

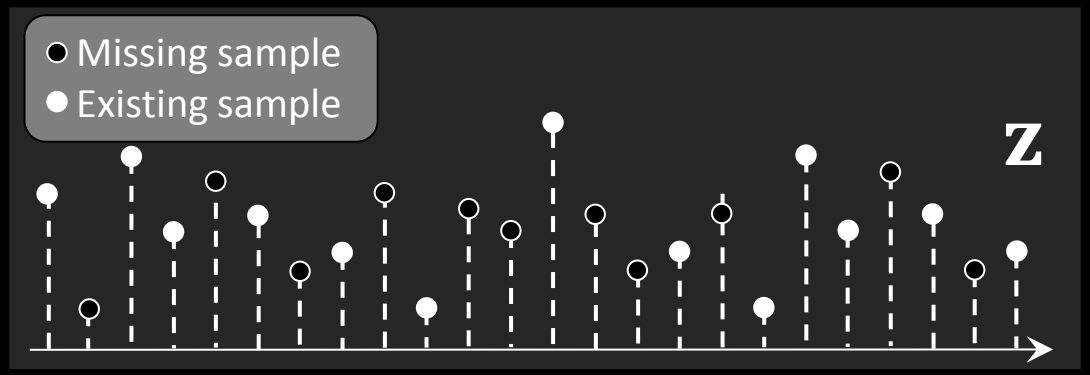
Bottom line: (1) This idea works very well;
(2) It is especially competitive for high noise levels; and
(3) A second iteration always pays off.

Image Inpainting

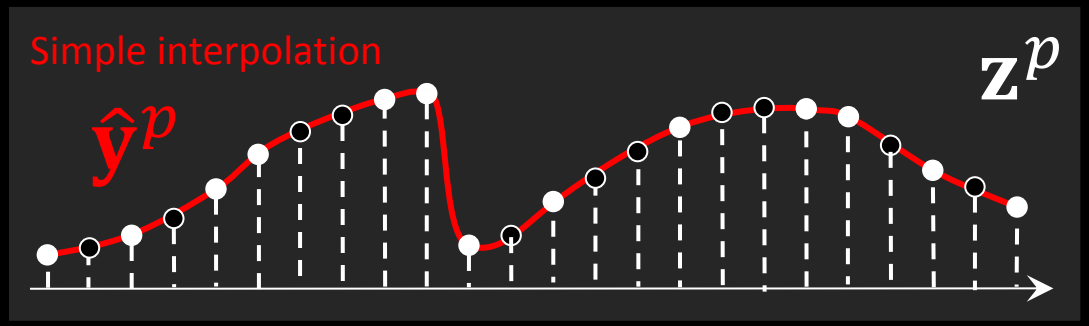
0.8 of the pixels are missing



Reconstruction: 29.71dB *



Ordering, distance uses only EXISTING pixels



* This result is obtained with (i) cycle-spinning, (ii) sub-image averaging, and (iii) three iterations.

Inpainting Results – Examples

Given data 80% missing pixels | Bi-Cubic interpolation | DCT and OMP recovery | 1st iteration of the proposed alg. | 3rd iteration of the proposed alg.

Inpainting Results

Reconstruction results from 80% missing pixels using various methods:

Image	Method	PSNR [dB]
Lena	Bi-Cubic	30.25
	DCT + OMP	29.97
	Proposed (1st iter.)	30.25
	Proposed (2nd iter.)	31.80
	Proposed (3rd iter.)	31.96
Barbara	Bi-Cubic	22.88
	DCT + OMP	27.15
	Proposed (1st iter.)	27.56
	Proposed (2nd iter.)	29.34
	Proposed (3rd iter.)	29.71
House	Bi-Cubic	29.21
	DCT + OMP	29.69
	Proposed (1st iter.)	29.03
	Proposed (2nd iter.)	32.10
	Proposed (3rd iter.)	32.71

Bottom line:

(1) This idea works very well;

(2) It is operating much better than the classic sparse-rep. approach; and

(3) Using more iterations always pays off, and substantially so.

Conclusions

We propose new wavelet transforms for scalar functions defined on graphs or high dimensional data clouds

The proposed transform extends the classical orthonormal and redundant wavelet transforms

We demonstrate the ability of these transforms to efficiently represent and denoise images

We show that the obtained transform can be used as a regularizer in classical image processing Inverse-Problems

We also show that the obtained transform can be used as a learned dictionary for the compression of facial images

Finally, we show that using the ordering of the patches only, quite effective processing of images can be obtained

Future Directions

- Finding new image processing applications.
- Finding new applications for the proposed method that involve processing data on graphs and point clouds.
- Improving the ordering scheme, for example by allowing patches/features to be revisited.
- Compressing facial images using more than one dictionary and more advanced entropy coding methods.
- Using pixel permutations as regularizes in image processing problems.

Thank you for your time,

Questions?