This research was supported by the European Research Council under EU's 7th Framework Program, ERC Grant agreement no. 320649, and by the Intel Collaborative Research Institute for Computational Intelligence.

Facial Image Compression using Patch-Ordering-Based Adaptive Wavelet Transform

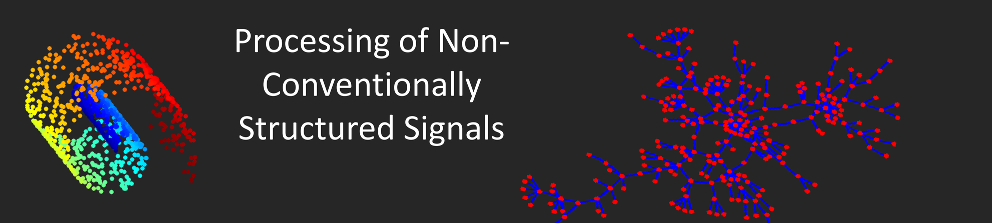
Michael Elad
The Computer Science Department

Idan Ram & Israel Cohen
The Electrical Engineering Department

Technion – Israel Institute of Technology

ICASSP 2015 Brisbane Australia
40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2015
19 - 24 April 2015, Brisbane, Australia

1. Motivation and Goals

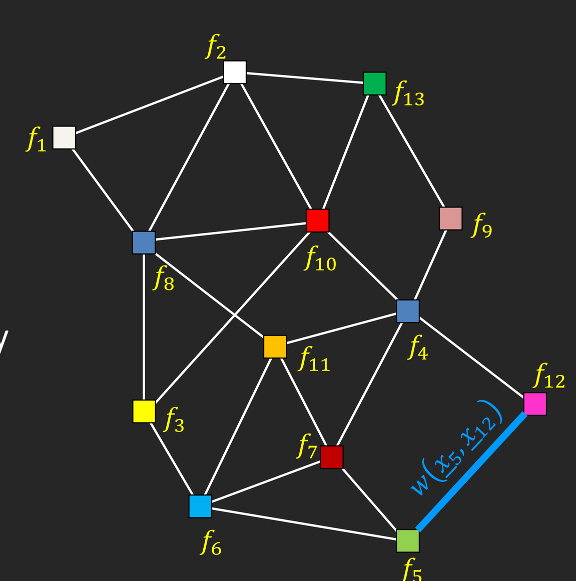


Many signal-processing tools (filters, alg., transforms, ...) are tailored for uniformly sampled signals → Now we encounter different types of signals: e.g., point-clouds and graphs. Can we extend classical tools to these signals? → Our goal: Generalize the wavelet transform to handle this broad family of signals → The true objective: Find how to bring sparse representation to processing of such signals

We use these developed tools in order to better process images ... and specifically compress facial images

2. Wavelet for Graphs: Formulation

- We are given a graph:
  - The $i-th$ node is characterized by a d-dimen. feature vector $\underline{x}_i$
  - The $i-th$ node has a value f_i
  - The edge between the $i-th$ and $j-th$ nodes carries the distance $w(\underline{x}_i, \underline{x}_j)$ for an arbitrary distance measure $w(\cdot,\cdot)$.

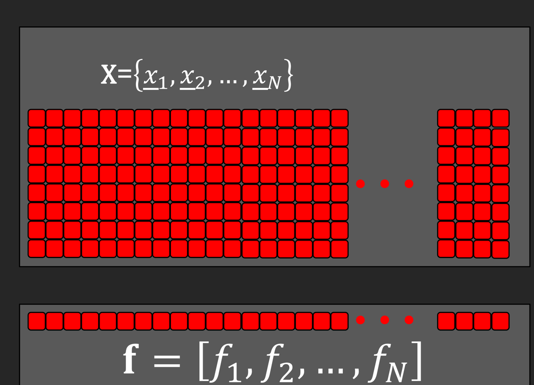


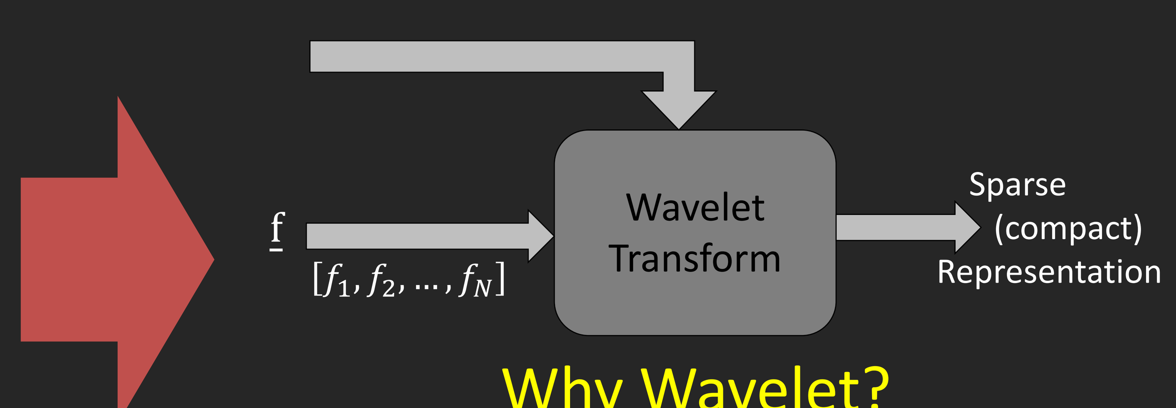
- Assumption: a "short edge" implies close-by values, i.e.

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

for almost every pair (i, j).

- A Different Way to look at this data structure:
  - We start with a set of d-dimensional vectors $\mathbf{X} = \{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N\} \in \mathbb{R}^d$. These could be:
    - Feature points for a graph's nodes,
    - Set of coordinates for a point-cloud.
  - A scalar function is defined on these coordinates, $f: \mathbf{X} \rightarrow \mathbb{R}$, giving $\mathbf{f} = [f_1, f_2, \ldots, f_N]$.
  - We may regard this dataset as a set of N samples taken from a high dimensional function $f: \mathbb{R}^d \rightarrow \mathbb{R}$.
  - The assumption that small $w(\underline{x}_i, \underline{x}_j)$ implies small $|f_i - f_j|$ for almost every pair (i, j) implies that the function behind the scene, f, is "regular".





Why Wavelet?

Wavelet for regular piece-wise smooth signals is a highly effective "sparsifying transform". We would like to imitate this for our data structure.

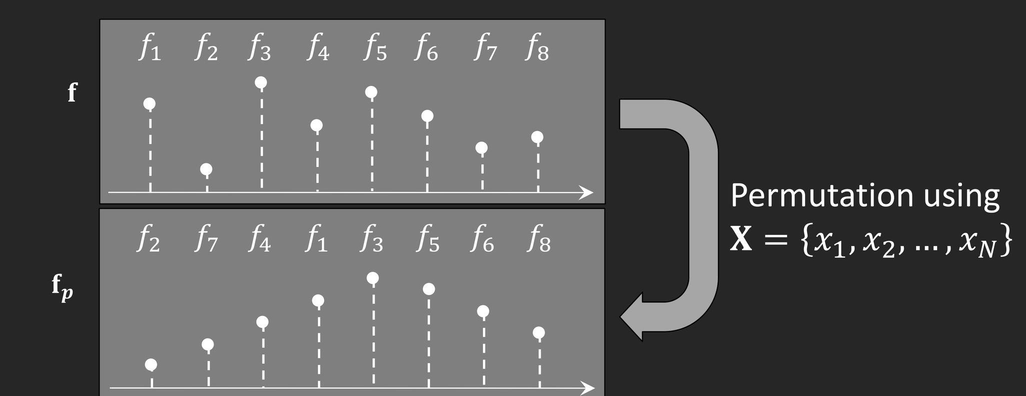
References

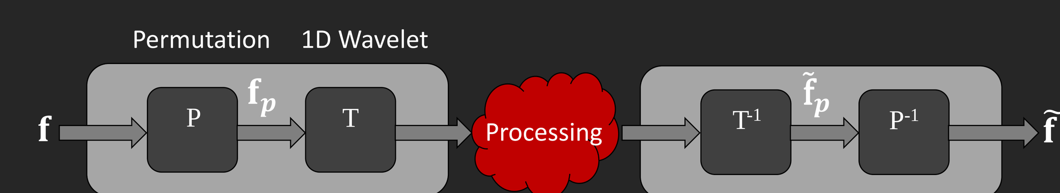
1. I. Ram, M. Elad, and I. Cohen, "Generalized Tree-Based Wavelet Transform", IEEE Trans. Signal Processing, vol. 59, no. 9, pp. 4199–4209, 2011.
2. I. Ram, M. Elad, and I. Cohen, "Redundant Wavelets on Graphs and High Dimensional Data Clouds", IEEE Signal Processing Letters, Vol. 19, No. 5, pp. 291–294 , May 2012.
3. I. Ram, M. Elad, and I. Cohen, "The RTBWT Frame – Theory and Use for Images", to appear in IEEE Trans. on Image Proc.
4. 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.
5. I. Ram, I. Cohen, and M. Elad, "Facial Image Compression using Patch-Ordering-Based Adaptive Wavelet Transform", Submitted to IEEE Signal Processing Letters.
6. M. Gavish, B. Nadler, and R. Coifman, "Multiscale wavelets on Trees, Graphs and High-Dimensional Data: Theory and Applications to Semi-Supervised Learning", ICML 2010.
7. M. Elad, R. Goldenberg, and R. Kimmel, "Low bit-rate compression of facial images," IEEE Trans. Image Process., vol. 16, no. 9, pp. 2383–2379, 2007.
8. O. Bryt and M. Elad, "Compression of facial images using the k-svd algorithm," J. Vis. Commun. Image Represent., vol. 19, no. 4, pp. 282–270, 2008.
9. O. Bryt and M. Elad, "Improving the k-svd facial image compression using a linear deblocking method," in Proc. 25th IEEE Conv. Electrical and Electronics Engineers in Israel, pp. 533–537, 2008.

3. The Proposed Methodology

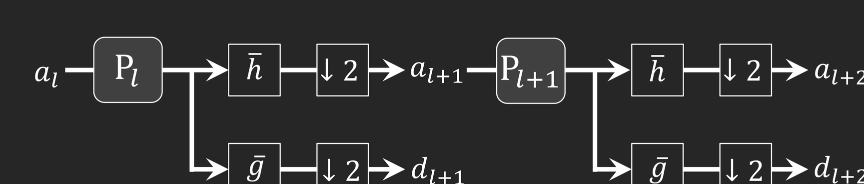
- Wavelet for Graphs: I wish we would have thought of it first ...
  - "Diffusion Wavelets" R. R. Coifman, and M. Maggioni, 2006.
  - "Multiscale Methods for Data on Graphs and Irregular ... Situations" M. Jansen, G. P. Nason, and B. W. Silverman, 2008.
  - "Wavelets on Graph via Spectal Graph Theory" D. K. Hammond, and P. Vandergheynst, and R. Gribonval, 2010.
  - "Multiscale Wavelets on Trees, Graphs and High ... Supervised Learning" M . Gavish, and B. Nadler, and R. R. Coifman, 2010.
  - "Wavelet Shrinkage on Paths for Denoising of Scattered Data" D. Heinen and G. Plonka, 2012.
  - ...

- The Main Idea (1) - Permutation





- The Main Idea (2) - Permutation
  - 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.
  - Thus, the difference between this and the plain 1D wavelet transform applied on $\mathbf{f}$ are the additional permutations, thus preserving the transform's linearity and unitarity, while also adapting to the input signal.

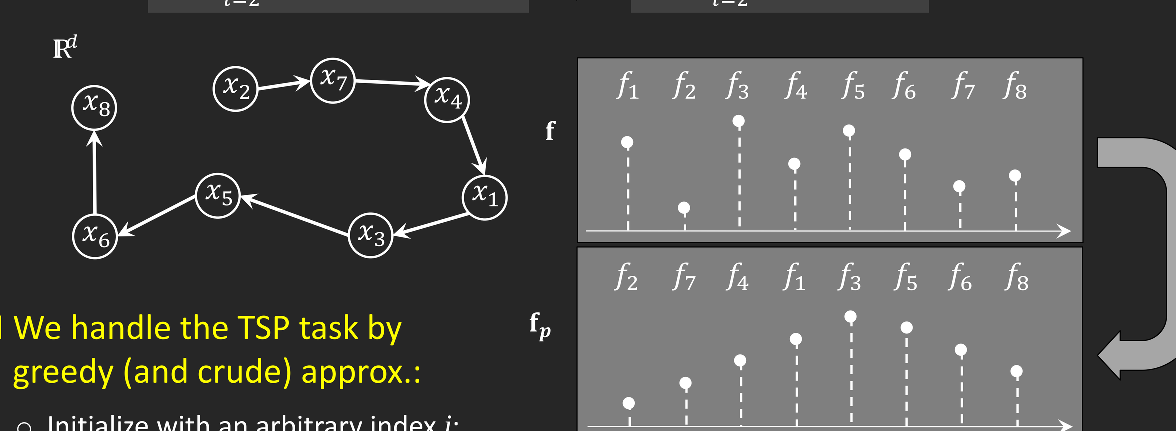
4. Building the Permutations

- Core Concept – TSP:
  - Lets start with P_0 – the permutation applied on the incoming signal.
  - Recall: the wavelet transform is most effective for piecewise regular signals. → thus, P_0 should be chosen such that $P_0\mathbf{f}$ is most "regular".
  - Lets use the feature vectors in $\mathbf{X}$, reflecting the order of the values, f_k. Recall:

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

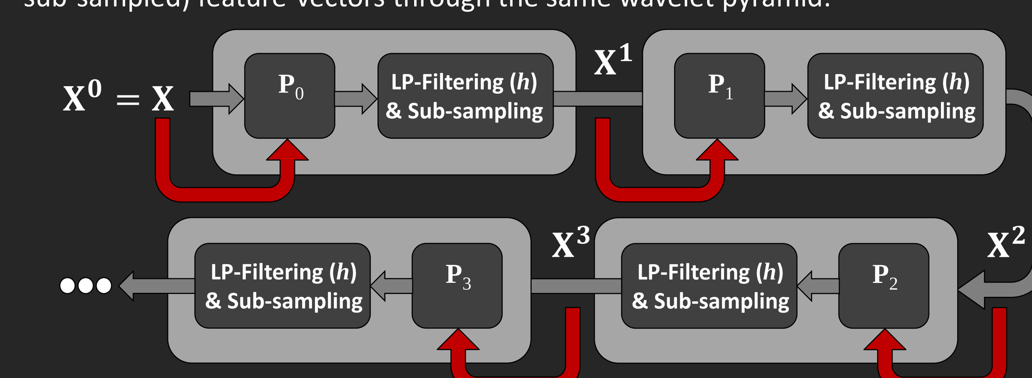
  - Thus, instead of solving for the optimal permutation that "simplifies" $\mathbf{f}$, we order the features in $\mathbf{X}$ to the shortest path that visits in each point once, in what will be an instance of the Traveling-Salesman-Problem (TSP):

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



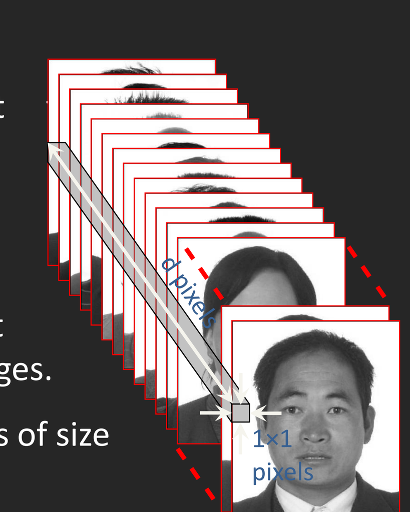
- We handle the TSP task by greedy (and crude) approx.:
  - Initialize with an arbitrary index j;
  - Initialize the set of chosen indices to $\Omega(1)=\{j\}$;
  - Repeat $k=1:1:N$-1 times:
    - Find x_i – the nearest neighbor to $x_{\Omega(k)}$ such that $i \notin \Omega$;
    - Set $\Omega(k+1)=\{i\}$;
  - Result: the set Ω holds the proposed ordering.

- What About the Inner Permutations?
  - So far we concentrated on P_0 at the finest level of the multi-scale pyramid.
  - 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:



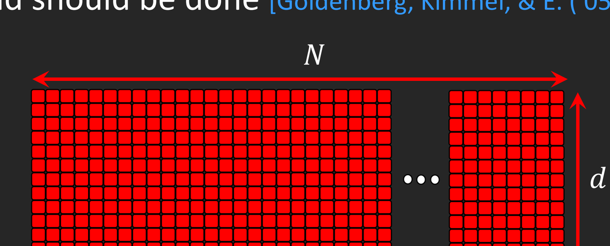
5. What About Image Compression?

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



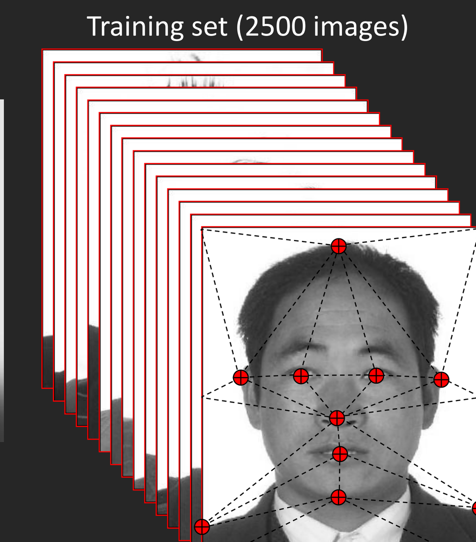
On the training set:
1. Detect main features and warp the images (20 bytes)
2. Compute the mean image and subtract it
3. Find the common ordering that creates the smoothest path
4. 2D→1D, apply wavelet and code leading coefficients

On the test image:
5. Warp, remove the mean, permute, apply wavelet on the 1D signal and code

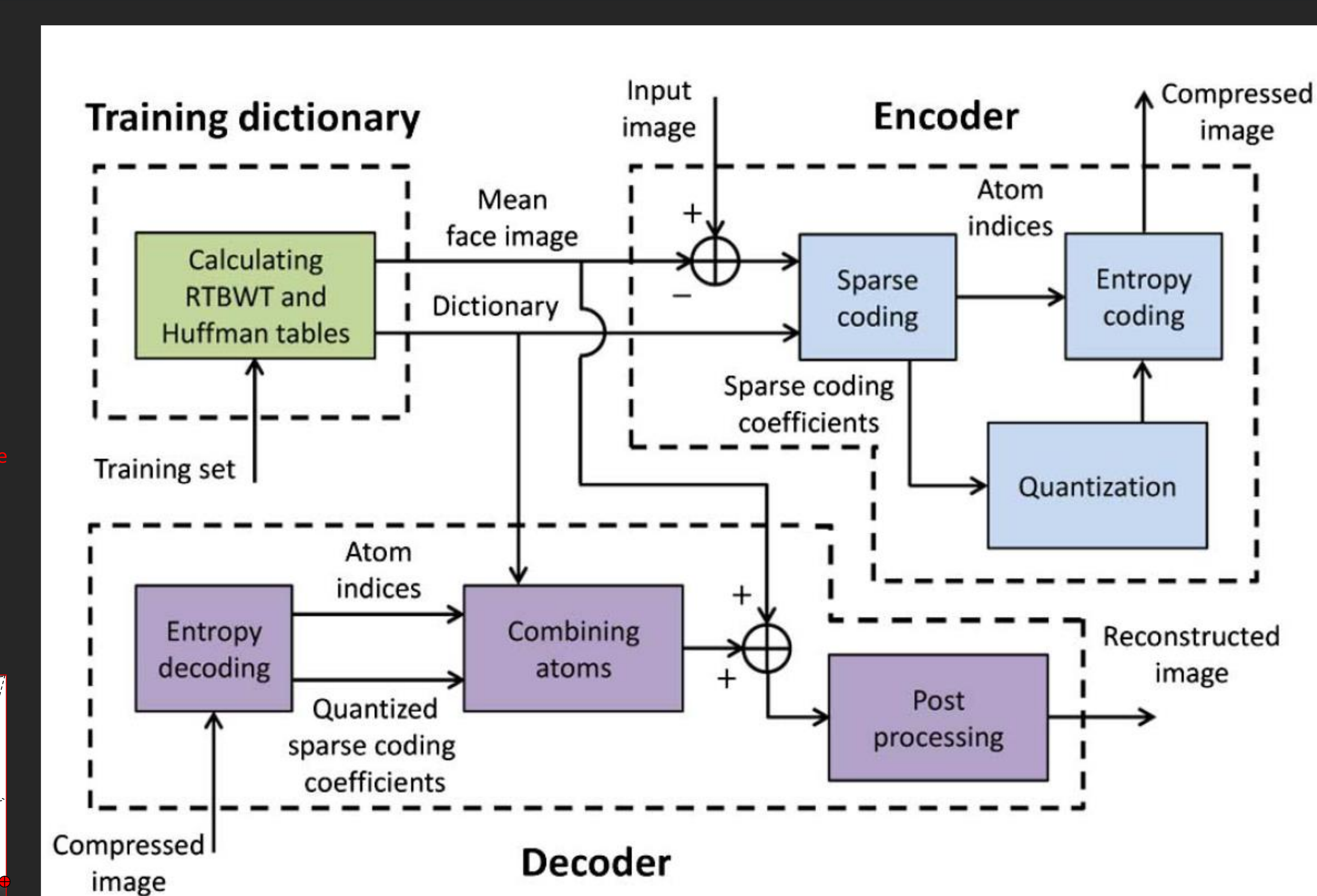


Mean Image

Training set (2500 images)

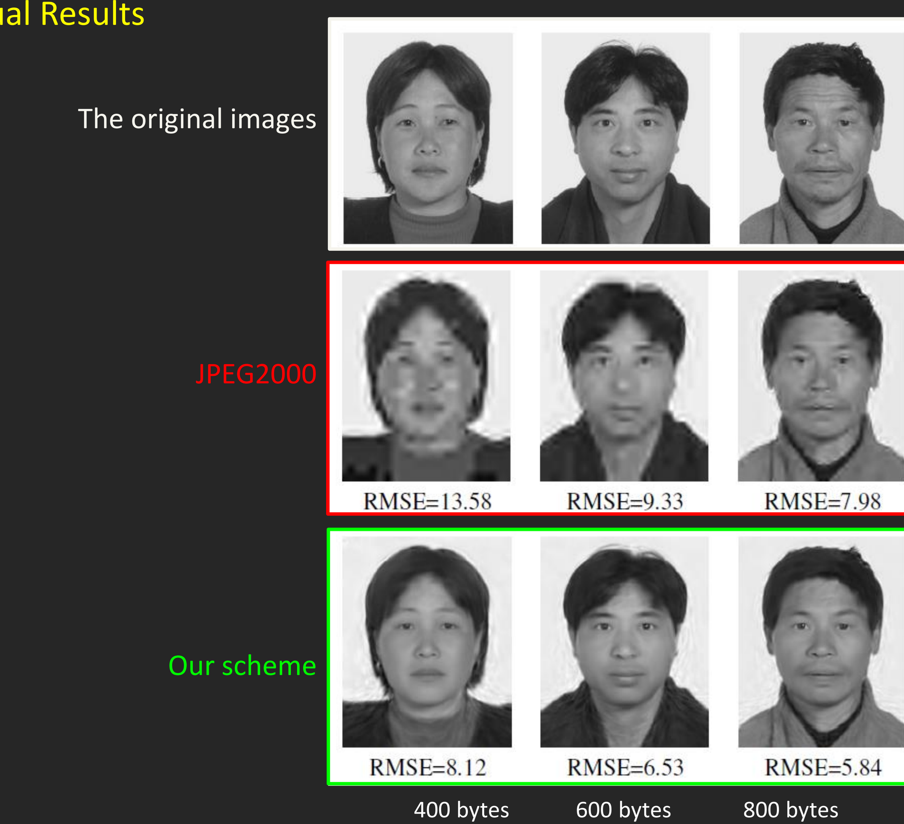


- Every row in the above array corresponds to a face image.
- Once we have formed the above array of data, our permutation-based wavelet can be designed.
- When applying this transform to a given new face image, it is expected to sparsify it very well → compression.

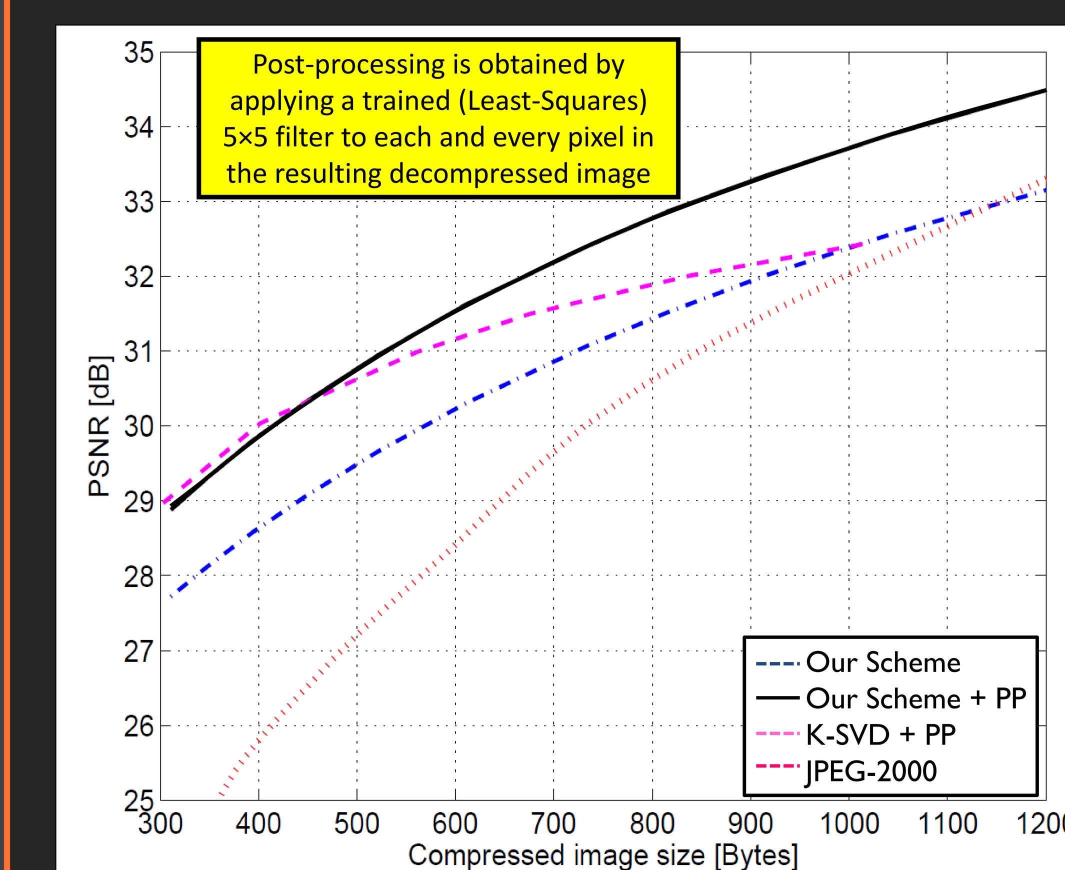


6. Results

- Visual Results



- Rate Distortion Curves



elad@cs.technion.ac.il