PARAMETRIZATION TO PATCH-DISAGREEMENT IN ORDER TO IMPROVE K-SVD DENOISING

Nerya Hadad and Maxim Lipatrov

Technion - Department of Computer Science

Project in Sparse Representations in Signal and Image Processing – 236862 by Prof. Michael Elad

Outline

- ABSTRACT
- INTRODUCTION
- SPARSELAND AND K-SVD
- THE PROPOSED ALGORITHM
- ORIGINAL EXPERIMENTS WITH THE PROPOSED ALGORITHM
- ALGORITHM IMPLEMENTATION
- ADDITIONAL WORK - PARAMETRIZATION
- CONCLUSIONS AND FUTURE DIRECTIONS

ABSTRACT

This is a presentation for the summary of the paper "PATCH-DISAGREEMENT AS A WAY TO IMPROVE K-SVD DENOISING" by Elad and Romano, extended by our remarks and descriptions.

In Addition, we propose parametrization in order to extend the original work, adds simulations to reproduce the paper's results and possibly reach slightly better results by tuning the introduced parameters.

INTRODUCTION - image denoising with degradation model

This paper deal with image denoising.

We'll use a degradation model of the form $y = x + v \quad (1),$ where:

y is a given known (deteriorated) image.

x is an original image.

v is an additional zero-mean Gaussian noise independent to x, which is the popular noise in image processing.

The denoising process is a seeking for approximation of the original image x.

INTRODUCTION - Patch-based methods

We will deal with the above problem using Patch-based methods.

Patch-based methods became very popular in last 20 years in all fields of image and signal processing, and might be helpful because:

- The input signal can be effectively large and can be processed if divided to patches.
- Neighbor pixels share the same information.

This paper suggest method to deal with main disadvantage of patch-based methods: the gap between the local processing of patches and the global image recovery. If we processes the patches independently, we loosing the relations between "neighbors" patches.

INTRODUCTION - K-SVD in short

Many denoising algorithms in field of signal and image processing are patch-based, but we will give a special attention to K-SVD.

In nutshell:

- It takes overlapping patches from the image and processes each patch assuming it can be represented as a sparse linear combination of elements of redundant dictionary.
- Restoring each patch by using this sparse model.
- Reconstructing the full image by averaging the overlapping patches.

After “sparseland” introduction, we will see K-SVD in more detail.

INTRODUCTION - disagreement patch

The “**disagreement patch**” defined as the difference between local denoised result from some patch and its corresponding patch from the global denoised outcome.

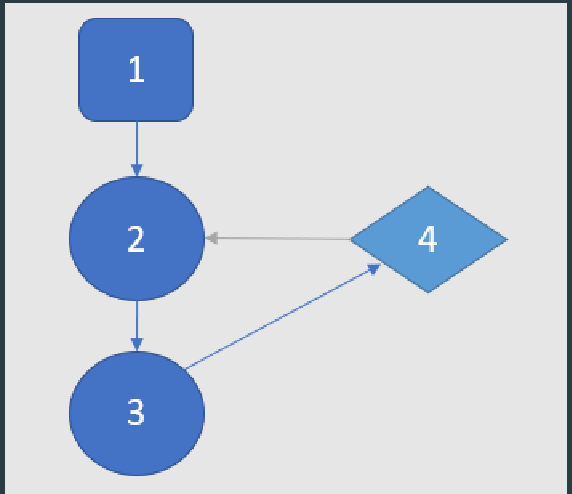
The disagreement patch represent the gap between the local processing of patches and the global image recovery. Hence, it’s can be used to overcome the disadvantage of patch-based methods and improve the results.

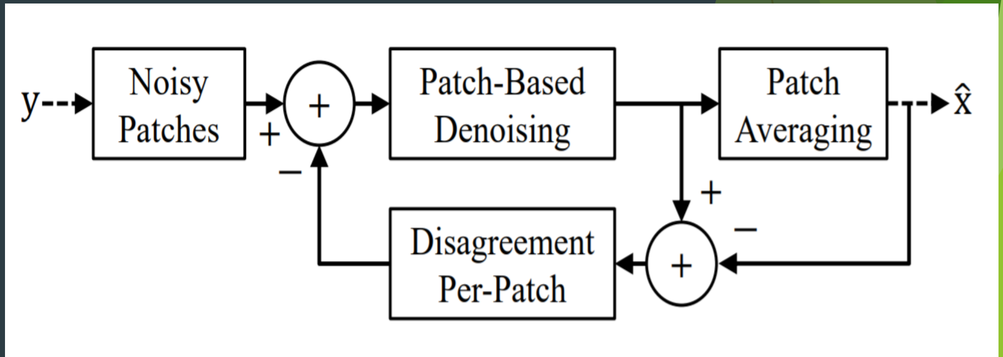
After “sparseland” introduction, we will see the “disagreement patch” in more detail.

INTRODUCTION – local-global solution

By “share the disagreement” between these overlapping patches. This can be done by the following iterative procedure, which is basically an addition to K-SVD:

1. Extracting the noisy patches, overlapping, of predefined size, as before.
2. Per patch: if it’s the first iteration, define the disagreement patch as 0, otherwise define it as we mentioned before (for last iteration’s recovery) and subtract it from the noisy input patches.
3. Apply K-SVD on result of the subtraction – for every patch.
4. Reconstruct by averaging the denoised outcome of current iteration.





SPARSELAND AND K-SVD - sparse assumption

“Sparseland” model of algorithms assumes that a signal can be represented\estimated by a few atoms of a redundant dictionary: $x = D\alpha$, such that:

$x \in \mathbb{R}^n$ is an input signal.

$D \in \mathbb{R}^{n \times m}$ is dictionary, where $m \gg n$ leading to redundancy of D.

$\alpha \in \mathbb{R}^m$ is a sparse vector.

SPARSELAND AND K-SVD - sparse estimation

Given deteriorated input y, we will use this model in order to estimate the original input signal x: $\hat{x} = D\hat{\alpha}$, where:

$\hat{x} \in \mathbb{R}^n$ is our estimation of x.

D $\in \mathbb{R}^{n \times m}$ is the dictionary.

$\hat{\alpha} \in \mathbb{R}^m$ - sparse projection of y onto the set of low-dimensional subspaces that D spans, our estimation of α.

SPARSELAND AND K-SVD - sparse solution

Given ϵ as the maximal error bound that we can treat, we will seek for $\hat{\alpha}$ - solution of:

$$(P_0\epsilon) \quad \hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \|\alpha\|_0 \quad s.t. \|D\alpha - y\|_2^2 \leq \epsilon^2, (2)$$

While this is a NP-hard problem, it can be properly approximated, as done in OMP, which used by K-SVD.

SPARSELAND AND K-SVD - dictionary

Universal dictionaries not always give the optimal restoration result for specific task.

Better results may be achieved by dictionary learning: adaptation of predefined dictionary to the concrete image or signal.

SPARSELAND AND K-SVD - dictionary learning

One of the steps of K-SVD is a dictionary update by solving the problem:

$\left[\hat{D},\{\hat{\alpha}_i\}_{i=1}^N\right] = \underset{\hat{D},\{\hat{\alpha}_i\}_{i=1}^N}{\operatorname{argmin}}\left[\sum_{i=1}^N \|D\alpha_i - y_i\|_2^2\right]\ s.t.\ Supp\{\alpha_i\} = Supp\{\hat{\alpha}_i\},\quad (3)$ where:

$\hat{D}$ is a learned updated dictionary.

$Supp\{\cdot\}$ is the indices of non-zero elements of given vector.

Iterating this process of solving equation (3) – “Dictionary update step” with solving equation (2) – “Sparse coding step” – is a well-known Dictionary Learning.

SPARSELAND AND K-SVD - K-SVD

As seen, K-SVD is patch-based algorithm.

It iterate as just described. Then, it reconstructing the denoised image by weighted-averaging the resulting patches.

It approximates the solution of the following problem:

$$\left[\hat{x}, \widehat{D}, \{\hat{\alpha}_i\}_{i=1}^N\right] = \underset{\hat{x}, \widehat{D}, \{\hat{\alpha}_i\}_{i=1}^N}{argmin}\ \mu\|x-y\|_2^2 + \sum_{i=1}^{N} \|\alpha_i\|_0 \quad s.t.\ \ \forall i\ [\|D\alpha_i - R_i x\|_2^2 \le \epsilon^2]$$

where:

N denotes the number of image patches.

$R_i \in \mathbb{R}^{n \times rc}$ extracts the i'th patch from the global image.

THE PROPOSED ALGORITHM - disagreement patch

As seen, The "**disagreement patch**" defined as the difference between local denoised result and its corresponding patch from the global denoised outcome.

Thus, the "disagreement patch" at the k^{th} iteration of patch i defined as:

$q_i^k = D^k \alpha_i^k - R_i \hat{x}^k, \qquad (4)$ where:

$D^k \alpha_i^k$ is the locally denoised result (of patch i).

$R_i \hat{x}^k$ is the corresponding part from the global denoised outcome.

THE PROPOSED ALGORITHM - narrow the local-global gap

By subtracting every "disagreement patch" from the corresponding input patch $R_i y$, the overlapping patches encouraged to narrow the gap and get an agreement, in addition to their local denoising.

Thus, the input patch i of next iteration is:

$$p_i^k = R_i y - q_i^k = R_i y - D^k \alpha_i^k + R_i \hat{x}^k = R_i \hat{x}^k + \left(R_i y - D^k \alpha_i^k\right) = R_i \hat{x}^k + r_i^k, \qquad (5)$$

r_i^k is local method-noise patch obtained at k^{th} iteration.

Namely, the algorithm tries to recover a patch from the global estimation $\hat{x}^k$, corrupted by the method-noise patch r_i^k.

THE PROPOSED ALGORITHM - pseudo

Initialization:

(1.) $k \leftarrow 0,\ \forall i:\ q_i^0 \leftarrow 0.$ $(2.)$ D^0 - an initial dictionary.

Repeat: Until maximal restoration quality is obtained (else go to 1).

1. Sparse coding step: using the OMP, compute as in equation (2):

$$\left[\{\hat{\alpha}_i^{k+1}\}_{i=1}^N\right] = \min_{\{\hat{\alpha}_i\}_{i=1}^N} \sum_{i=1}^N \|\alpha_i\|_0 \ \ s.t.\ \forall i \left[\left\|D^k\alpha_i - p_i^k\right\|_2^2 \le \epsilon^2\right]$$

2. Dictionary update step, solve equation (3):

$$\left[D^{k+1}, \{\alpha_i^{k+1}\}_{i=1}^N\right] = \min_{D,\{\alpha_i\}_{i=1}^N} \sum_{i=1}^N \left\|D\alpha_i - p_i^k\right\|_2^2 \ \ s.t.\ Supp\{\alpha_i\} = Supp\{\hat{\alpha}_i^{k+1}\}$$

   In practice, D^{k+1} is obtained from D^k by K-SVD.

3. Image reconstruction step, solve:

$$\hat{x}^{k+1} = \min_x \sum_i \left\|D^{k+1}\alpha_i^{k+1} - R_i x\right\|_2^2 + \mu\|x - y\|_2^2$$

   which is basically the averaging of the denoised patches on the overlaps, followed by a weighted average with the noisy image.

4. Disagreement-update step, compute as in equation (4),

$$\forall i, q_i^{k+1} = D^{k+1}\alpha_i^{k+1} - R_i\hat{x}^{k+1}$$

   set $k \leftarrow k + 1$.

Output: $\hat{x}^k$ - the last result.

ORIGINAL EXPERIMENTS WITH THE PROPOSED ALGORITHM - K-SVD comparison

After obtaining initial dictionary by applying 20 iterations of the K-SVD algorithm:

This algorithm with 30 iterations has been compared to the K-SVD.

Parameters of compression:

- Pictures (6 different).
- Corruption by additive zero-mean Gaussian noise with standard deviation σ (6 different std).

ORIGINAL EXPERIMENTS WITH THE PROPOSED ALGORITHM - results

The noise energy of $p_i^k = R_i\hat{x}^k + r_i^k$ (5) must be larger than σ.

Thus, the experiments show that using bigger noise energy ($\hat{\sigma}$ > σ) leads to better performance.

For every σ the chosen $\hat{\sigma}$ in the above table is giving the best results.

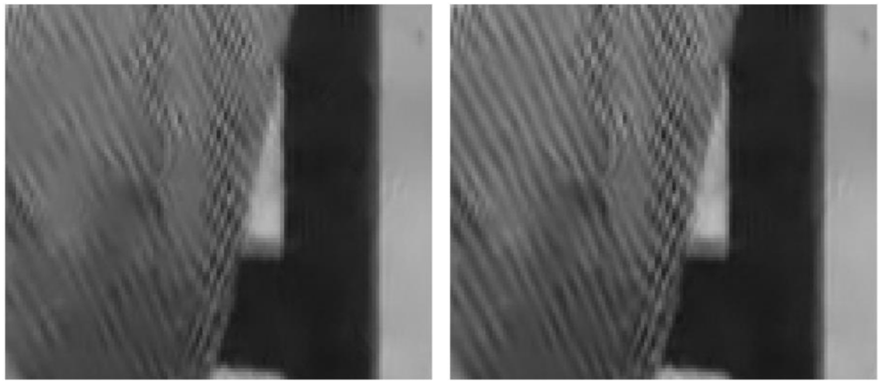
The best results per each image and noise level are highlighted.

σ	$\hat{\sigma}$	Barbara		Boat		House		Fingerprint		Peppers		Couple		Average
		Orig	New	Orig	New	Orig	New	Orig	New	Orig	New	Orig	New	New
10	1.12σ	34.55	34.55	33.63	**33.70**	36.06	**36.09**	32.37	**32.40**	34.81	**34.83**	33.57	**33.69**	0.05
20	1.06σ	30.86	**31.03**	30.40	**30.62**	33.16	**33.37**	28.48	**28.65**	32.31	**32.42**	30.05	**30.31**	0.19
25	1.06σ	29.61	**29.88**	29.31	**29.58**	32.30	**32.62**	27.29	**27.51**	31.50	**31.61**	28.96	**29.29**	0.25
50	1.04σ	25.38	**26.12**	25.92	**26.36**	27.93	**28.69**	23.31	**23.98**	28.16	**28.58**	25.28	**25.80**	0.59
75	1.02σ	22.89	**23.53**	23.94	**24.45**	25.22	**25.96**	20.00	**21.49**	25.80	**26.36**	23.65	**24.10**	0.73
100	1.02σ	21.83	**21.99**	22.86	**23.16**	23.63	**24.29**	18.28	**19.55**	24.26	**24.76**	22.63	**22.88**	0.52

ORIGINAL EXPERIMENTS WITH THE PROPOSED ALGORITHM - results and conclusions

- In terms of PSNR the algorithm get the best results compared to K-SVD for large σ.
- The algorithm improves the recovery of edges (a+b) and texture areas (c+d).

(a) K-SVD, PSNR = 30.88dB (b) Algorithm 1, PSNR = 31.43dB



(c) K-SVD, PSNR = 28.56dB (d) Algorithm 1, PSNR = 29.59dB

ALGORITHM IMPLEMENTATION - differences

There might be some different between the original results\code (which we don't have) and our implementation, for examples:

- The input pictures might be slightly different or in another resolution.
- Denoising steps and exact implementation of K-SVD and disagreement:
  - Patch size, Parameter μ, initial dictionary.
  - Error-constrained OMP instead of sparsity-constrained.

Thus the best results might come with different noise energy and be different.

ALGORITHM IMPLEMENTATION - results comparison

As in the original paper:

- Bigger noise level => smaller gain.
- Average result: in general, the bigger the initial noise level, the better the improvement.

Differences:

- In our case the average improvement is bigger than in original paper.
- Every sigma is bigger(1.12 - 1.16, compare to 1.02 - 1.12 in the original paper).
- One case of no improvement: Both in

$\sigma = 10$, but different picture.

σ	$\hat{\sigma}$	Barbara		Boat		House		Fingerprint		Peppers		Couple		Average
		Orig	New	Orig	New	Orig	New	Orig	New	Orig	New	Orig	New	
10	1.16	34.15	34.53	33.51	33.70	35.79	36.24	32.46	32.46	34.03	34.45	33.39	33.65	0.28
20	1.16	30.54	31.01	30.20	30.58	32.22	33.42	28.50	28.61	30.31	30.98	29.92	30.29	0.53
25	1.16	29.27	29.80	29.00	29.50	30.95	32.61	27.30	27.44	29.22	29.83	28.74	29.19	0.65
50	1.16	25.17	25.52	25.16	25.73	26.64	28.93	23.51	23.80	25.30	25.73	24.96	25.45	0.73
75	1.14	22.17	22.93	22.73	23.46	23.63	24.67	21.01	21.20	22.28	23.28	22.62	23.25	0.73
100	1.12	20.24	21.13	20.56	21.60	21.69	22.31	19.10	20.25	20.37	21.43	20.57	21.46	0.94

ADDITIONAL WORK - PARAMETRIZATION - multiplier of disagreement-patches

We turned to question how the disagreement-patches values influence the denoising result.

We defined the parameter α, as an additional multiplier of disagreement-patches.

Thus, the Disagreement-Update step in Algorithm 1 is now:

$$\forall i, q_i^{k+1} = \alpha \cdot \left(D^{k+1}\alpha_i^{k+1} - R_i\hat{x}^{k+1}\right)$$

Notice that $\alpha = 0$ mean that the resulting algorithm is K-SVD in every iteration, while $\alpha = 1$ is the current algorithm.

ADDITIONAL WORK - PARAMETRIZATION - first reaserch

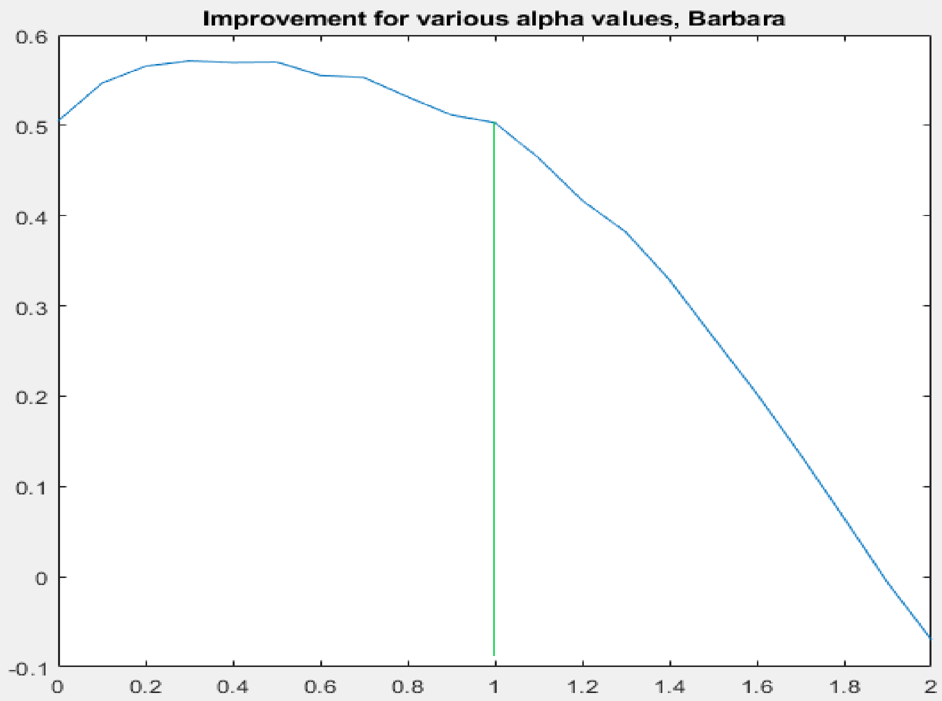
We tried various values between 0 and 2, with step of 0.1.

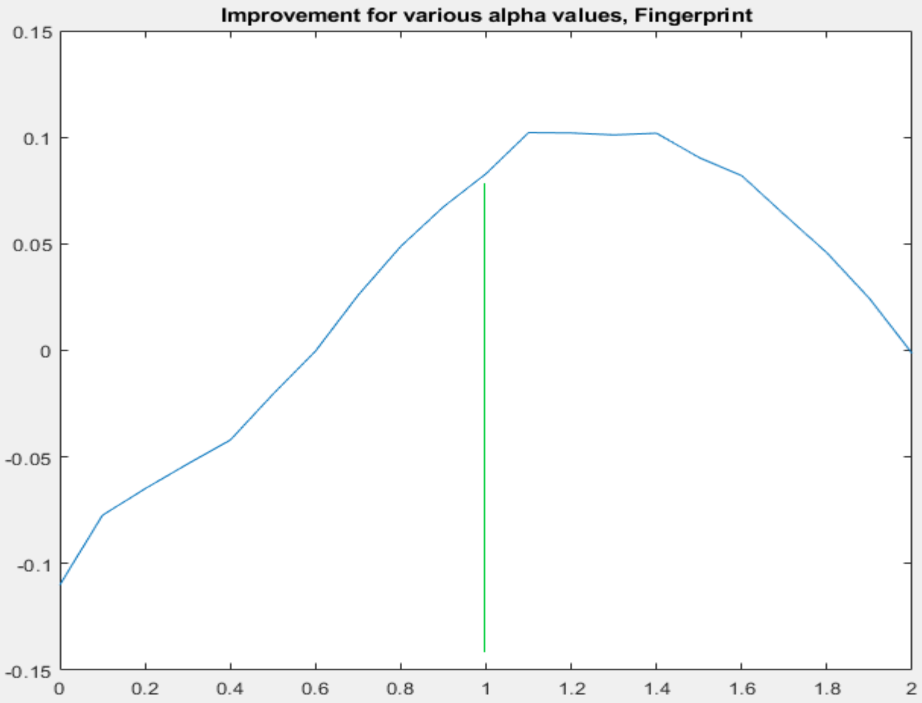
First, for 2 of the most different pictures: Barbara and Fingerprint.

We used fixed concrete σ=20 and gain 1.16.

As can be seen α=1 is not the best option. There is still a space for improvement up to 0.1 more in terms of PSNR.

The α providing the peak improvement was not the same for both pictures – about 0.4 for Barbara and about 1.3 for Fingerprint.





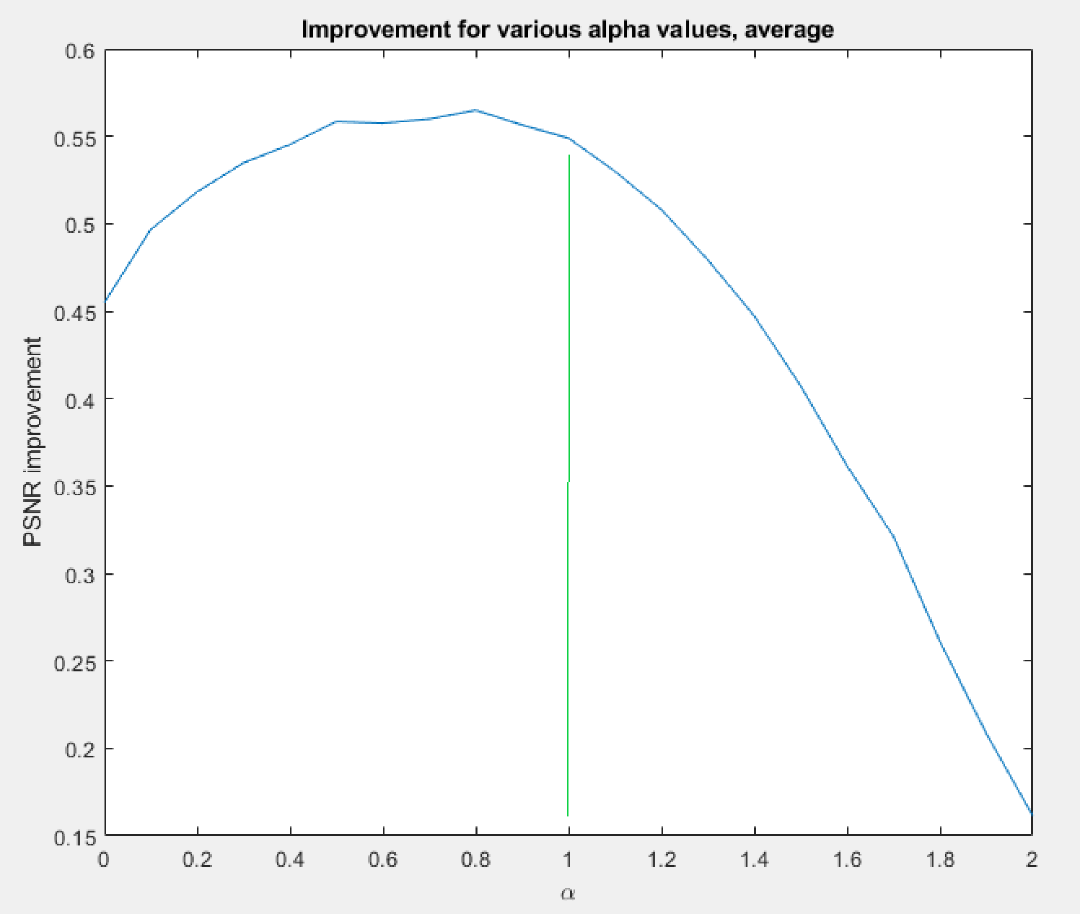
ADDITIONAL WORK - PARAMETRIZATION - determine α value

After seeing that different pictures might get different α for the best PSNR: we decided that α will be tuned with the best average PSNR result of all the pictures, for fixed concrete σ= *20 and gain 1.16*.

We tried various values between 0 and 2, with step of 0.1.

As can be seen α=0.8 is giving the best results: extra improvement of 0.05 dB.

While this improvement is not big, the results imply that the original algorithm can be improved.



CONCLUSIONS AND FUTURE DIRECTIONS - summary

Our work in short:

- Paper summary with comments regarding what we learned in the course.
- Implementation of the algorithms K-SVD and “Sharing the disagreement”.
- Same experiments as in original paper in purpose to get the same results.
- Proposed a small parametrization to the “Sharing the disagreement”.

We learned a lot as a result and succeeded in both implementing the above algorithms and getting the almost same results as authors of original paper.

We showed that default value of α=1 is not the best in most cases.

CONCLUSIONS AND FUTURE DIRECTIONS - future directions

For $\alpha = 0.8$ we get the best average improvement of PSNR, but it is not the best for every picture. Thus, different α for different pictures can get better results.

Looking on the results images and dictionaries implies that best value of α vary depending on image textures.

Possible future direction is to build some kind of texture analyzer, possibly AI-based, which can predict the near-best value of α.

References

[1] Y. Romano and M. Elad, Patch-Disagreement and a Way to Improve K-SVD Denoising, ICASSP, Brisbane, Australia, April 19-24, 2015.

[2] Le Hou, Dimitris Samaras, Tahsin M. Kurc, Yi Gao, James E. Davis, Joel H. Saltz; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2424-2433

[3] PATCH-BASED IMAGE INTERPOLATION: ALGORITHMS AND APPLICATIONS, Xin Li, Lane Dept. of Computer Science and Electrical Engineering, West Virginia University, 2008

[4] Patch-Based Image Inpainting with Generative Adversarial Networks, Ugur Demir, Gözde B. Ünal , ArXiv 2018

[5] M. Aharon, M. Elad, and A.M. Bruckstein, The K-SVD Algorithm, Proceedings of SPARSE05, Rennes, France, November 2005.

[6] Y. Romano and M. Elad, "Improving K-SVD denoising by post-processing its method-noise," in IEEE Int. Conf. on Image Proc., Sept 2013, pp. 435–439.

[7] J. Sulam and M. Elad, "Expected patch log likelihood with a sparse prior," in submitted to EnergyMinimization-Methods workshop, Hong-Kong, January 13-16 2015.

[8] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in IEEE Int. Conf. on Computer Vision, 2011, pp. 479–486.

[9] SOS Boosting for Image Deblurring Algorithms, M.Elad, Y.Romano, S.Peled, 2018.