236862 Sparse and Redundant Representations Course Project

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

Maxim Lipatrov

maxim-li@campus.techion.ac.il

Nerya Hadad

neryahadad@gmail.com

ABSTRACT

This paper is a summary of paper "PATCH-DISAGREEMENT AS A WAY TO IMPROVE K-SVD DENOISING"[1] by Romano and Elad, extended by our remarks and descriptions, and as addition 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. The source code, this paper and the original one are available on GitHub: https://github.com/neryah/PATCH-DISAGREEMENT-AS-A-WAY-TO-IMPROVE-K-SVD-DENOISING-implementation-and-improvement.

INTRODUCTION

Patch-based methods became very popular in last 20 years in all fields of image and signal processing, such as image classification [2], interpolation [3], inpainting [4] and more. One can claim that patch-based approach can be helpful because the input signal can be effectively large and hence can't be processed as is in some way but can be processed if divided to patches. Other can claim that "neighbor pixels share the same information" and hence patch-based approach works great. In practice, both claims are correct.

In this paper we want to deal with image denoising, which is also a long studied and significant problem in image processing. We'll use a degradation model of the form

$$y = x + v \quad (1),$$

where x is an original image, y is a given known (deteriorated) image and v is an additional zero-mean Gaussian noise independent to x. Why Gaussian? Gaussian noise is the popular one in image processing because of fact that digital images gets it from the camera sensor due to, for example, low light conditions. The denoising process here is a seeking for approximation of original image x given deteriorated one y. Assume w.l.o.g that x,y and v are held as column vectors.

Many denoising algorithms in field of signal and image processing are patch-based, but we want to give a special attention to K-SVD [1][5], about which we learned in this course and were impressed by its approach and achieved results, where the K-SVD is very effective in estimating the underlying signal. 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, and then reconstructing the full image by averaging the overlapping patches.

Here comes the disadvantage in the patch-based model – the gap between the local processing of patch and the global need of the whole image. This model processes the patches independently, disregarding of its “neighbors” and relations between them (as we mentioned before, neighbor pixels share the same information). From this observation we reach that the K-SVD model can be improved if we take the inter-relations between neighbor patches into account.

Many recent works in image processing addressed this local-global gap but focusing on K-SVD algorithm we propose to treat this gap in a different way. In original paper, authors define the difference between local denoised result from some patch and its corresponding patch from the global denoised outcome – as a “disagreement patch”. This “disagreement patch” is not empty because of individual processing of every patch. Now, due to fact that denoised image patch is an average of overlapping patches, this special patch considers the difference between neighbor patches of the local denoised result, and hence we can use it to improve the result. We suggest to “share the disagreement” between these overlapping patches. This can be done by the following iterative procedure, which is basically a addition to K-SVD (Figure 1).

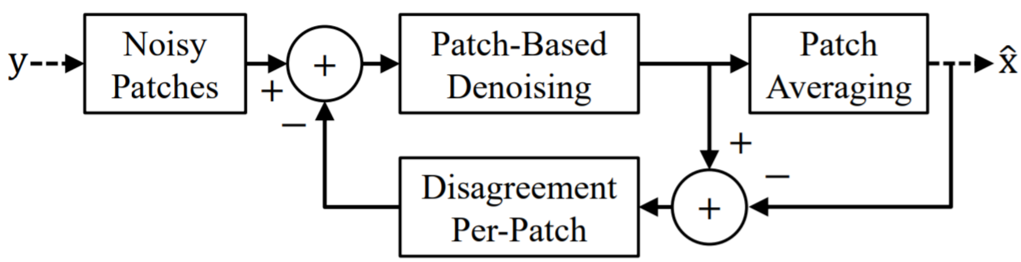


Figure 1. The schema of proposed algorithm

This procedure includes the steps:

1) Extracting the noisy patches, overlapping, of predefined size, as before.
2) Now, per-patch: if it’s first time we reach step 2, define disagreement as empty (equals to 0), otherwise define it as we mentioned before (the difference between local denoised result of last iteration and the corresponding patch from the averaged outcome of last iteration) and subtract it from the noisy input patches.
3) Apply K-SVD on result of subtraction – for every patch.
4) Reconstruct by averaging the denoised outcome of current iteration (and back to 2).

In section 5 we propose to introduce parameter in order to tune the influence on “disagreement patch” on the outcome. Consider that this idea of “sharing the disagreement” can be applied on every patch-based method to improve the local-global gap, not specifically denoising, and not specifically K-SVD.

1 SPARSELAND AND K-SVD

"Sparseland" model of algorithms assumes that a signal can be represented by a very few atoms of a redundant dictionary. Denote the dictionary as D and the original input signal as x, then:

$$x = D\alpha,$$

where $\alpha \in \mathbb{R}^m$ is a sparse vector, $x \in \mathbb{R}^n$ and $D \in \mathbb{R}^{n\times m}$ where $m \gg n$ leading to redundancy of D. Now, using this model in order to estimate the original input signal x and given deteriorated one y, we can find an estimation of α – let's denote it as $\hat{\alpha}$ – the projection of y onto the set of low-dimensional subspaces that D spans (recall that α is sparse) and then obtain the denoised signal by:

$$\hat{x} = D\hat{\alpha}$$

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

$$(P_0\epsilon) \quad \hat{\alpha} = \min_{\alpha} \|\alpha\|_0 \quad s.t. \|D\alpha - y\|_2^2 \le \epsilon^2 \text{ , } (2)$$

where $\|\cdot\|_0$ is an operator counting the number of non-zeros, i.e. the L_0-norm. As we saw in course, this is a NP-hard problem, but can be properly approximated via greedy or relaxation methods such as Thresholding, Orthogonal Matching Pursuit (as done in K-SVD) and others.

Moreover, we saw in the course that achieving a good dictionary for specific task can be hard, and universal dictionaries like combinations of Fourier, Hadamard, Wavelets, DCT and more not always give the optimal restoration result, although easy in usage. The way may give better result is a dictionary learning, i.e. adaptation of predefined dictionary to the concrete image. In final "wet" course project we adapted DCT dictionary to the part of well-known image "Barbara" and mentioned that the resulting dictionary varied a lot from initial and its atoms started to look very similar to image textures, especially on scarf, chair and tablecloth – see figures 2 and 3.

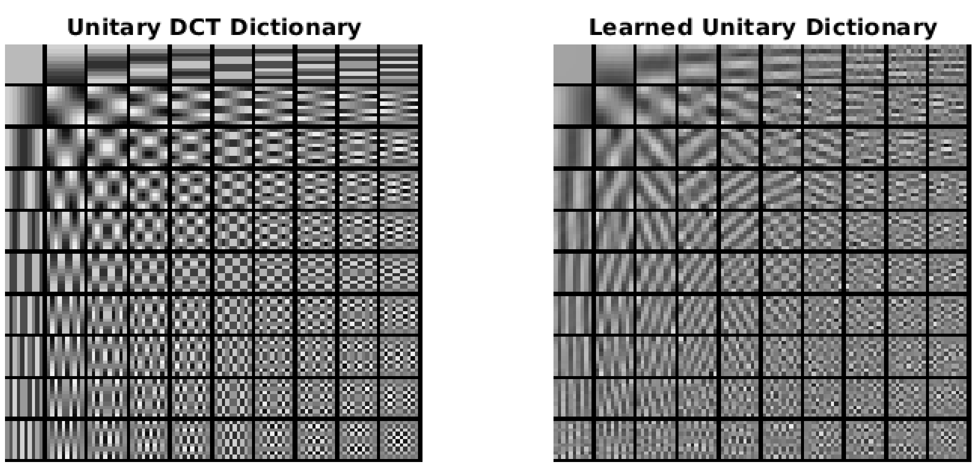


Figure 2. The comparison of DCT dictionary before and after learning on Barbara image, using Procrustes Analysis (taken from one of course homeworks).

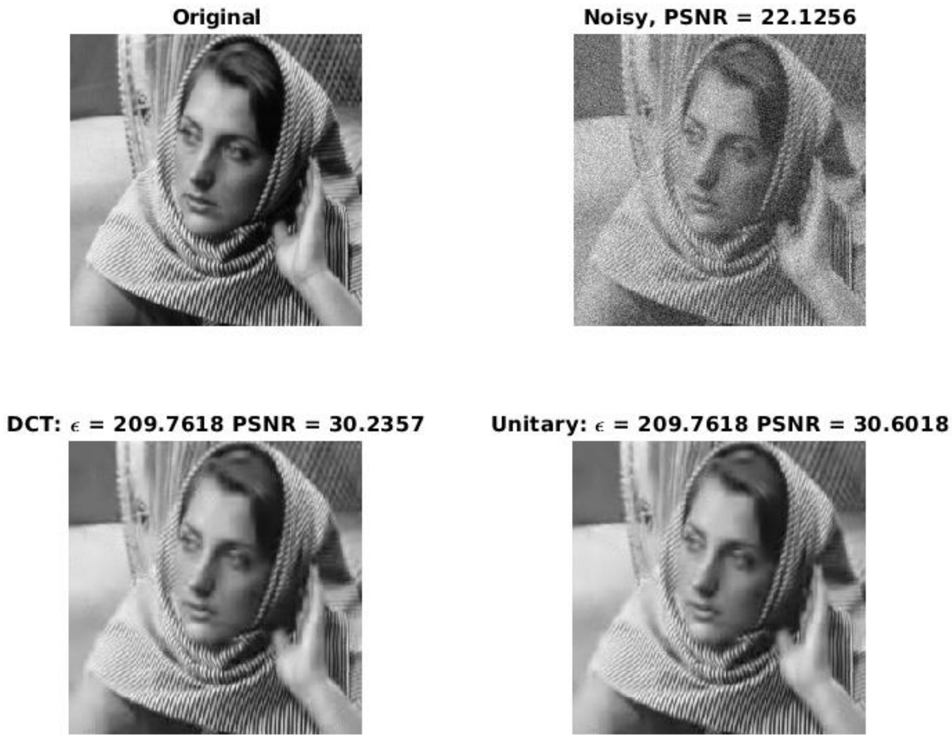


Figure 3. Original part of Barbara image, its noisy version, denoised by DCT version and denoised by learned DCT dictionary (via Procrustes Analysis), and their resulting PSNR.

Thus, we mention that dictionary learning may give better restoration results. One of the steps of K-SVD is a dictionary update by solving the problem:

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

where the resulting $\widehat{D}$ is a learned updated dictionary and $Supp\{\cdot\}$ returns 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.

Unfortunately, we can't use in practice sparsity-inspired algorithms on a whole image as they are limited to handle relatively small signals. That's why the patch-based method is used in K-SVD: it breaks the image to $\sqrt{n} \times \sqrt{n}$ overlapping patches, cleans them by iterating between sparse coding and dictionary update as described above and then reconstructing the denoised image by weighted-averaging the resulting patches and placing them to their original locations. This is the K-SVD, and it approximates the solution of the following problem:

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

where N denotes the number of image patches, and the matrix $R_i \in \mathbb{R}^{n \times rc}$ extracts the i'th patch from the global image.

2 THE PROPOSED ALGORITHM

According to K-SVD algorithm, each patch is treaten independently, and that's why we have a place for improvement – we lost the estimations on the overlaps, disregarding the patch positions. The proposed algorithm aims to narrow this local-global gap by encouraging the overlapping patches to influence each other. More specifically, the "consensus" problem involves the minimization of a single global variable (the denoised image), where the objective and constraint terms split into N parts (the recovery of the overlapping patches). In addition, the closely related "sharing" problem involves the adjustment of local variables to minimize their own (local) cost function, as well as the shared (global) objective. Following these ideas, the proposed iterative method drives the overlapping patches towards an agreement by sharing the neighbors disagreements, thus called "sharing the disagreement".

Let's define now the "disagreement patch" (in same context of K-SVD denoising described above) 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 (on place of patch i) and $R_i \hat{x}^k$ is the corresponding part from global estimate, both at k iteration. We described before that sparse coding does not consider relations between patches, hence q_i^k can't be considered as zero vector and its energy is not negligible. Next, we are making overlapping patches collaborate (what was lacking in standard K-SVD denoising) by modifying the input patches for the next iteration of denoising – by subtracting q_i from $R_i y$ – and result of subtraction becomes the input for next iteration. This procedure is repeated several times, according to schema at Fig. 1. According to K-SVD and definition (4), we can express the new input patch as:

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

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

<u>**Algorithm 1: Sharing the disagreement**</u>

<u>Initialization</u>:

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

<u>Repeat</u>: Until maximal restoration quality is obtained (else go to 1).

1) Sparse coding step: using the OMP, solve:

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

2) Dictionary update step, solve:

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

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 (4),

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

and then set $k \leftarrow k + 1$.

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

Algorithm 1 tries to recover a patch from the global estimation $\hat{x}^k$, corrupted by the method-noise patch r_i^k. This approach is different from earlier work of M.Elad and Y.Romano [6], trying to improve the denoising result by post-processing its method-noise. The EPLL [7, 8] approach also reduces the local-global gap but in a different way that “sharing the disagreement” approach does. The uniqueness in this approach is the fact that it harnesses intermediate patch-denoising results, which are inner to K-SVD.

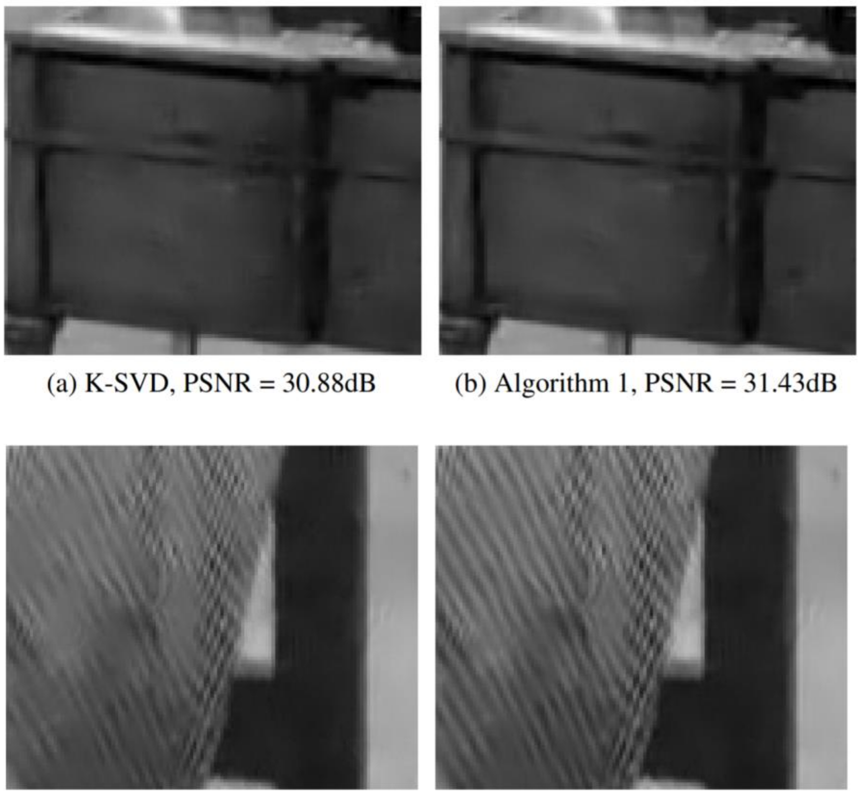
3 ORIGINAL EXPERIMENTS WITH THE PROPOSED ALGORITHM

In the original paper, authors presented detailed results for several noise methods and test images: Barbara, Boat, Fingerprint, House, Peppers and Couple. These images were corrupted by additive zero-mean Gaussian noise with standard deviation σ as described in (1). Results were compared via Peak Signal to Noise Ratio (PSNR) defined as $20\log_{10}\left(\frac{255}{\sqrt{MSE}}\right)$, where MSE is a Mean Squared Error between the original and the denoised images, i.e. the smaller the difference between the original and the denoised one, the bigger the PSNR. They also found that using the bigger noise energy leads to better performance, which originates from the noise energy of $p_i^k = R_i\hat{x}^k + r_i^k$ (5), which must be larger than σ, then this bigger noise energy $\hat{\sigma}$ was tuned and various good results were achieved. The denoising results of Table 1 are obtained by applying Algorithm 1 for 30 iterations, where each iteration includes sparse coding and dictionary update steps. In addition, the initial dictionary is obtained by applying 20 iterations of the K-SVD algorithm (leading to what is referred to in the table as ’Orig’ results).

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

Table 1. Comparison between the denoising results [PSNR] of the original K-SVD algorithm [5] and the ”Sharing the disagreement” outcome (Algorithm 1). The best results per each image and noise level are highlighted.

From Table 1, in terms of PSNR, the proposed algorithm improves the original K-SVD denoising for all images and noise levels (especially for large σ). Visually, the proposed method improves the recovery of edges and texture areas, and are consistent with PSNR increase, as showed in Figure 4.



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

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

Figure 4. Results of comparison of original K-SVD denoising and denoising using Algorithm 1 from original paper. First pair better shows the better denoising of edges while the second one – better denoising of textures.

We use the same approach of comparison for evaluation of proposed addition of parametrization to “sharing the disagreement” algorithm described above. We first try to implement both algorithms – K-SVD and Algorithm 1, and then perform the experiments similar to which authors made.

4 OUR TRY TO REACH THE SAME RESULT

Trying to reach the same results as authors of original paper, we implemented K-SVD and “Sharing the disagreement” algorithms on our own, on Matlab, as part of our course project. For comparison, we used:

- DCT Dictionary of 256 atoms of length 64, non-unitary. It is given to K-SVD as initial dictionary.
- Various noise levels σ as done in original paper: 10, 20, 25, 50, 75, 100.
- Test images as in original paper: **Fingerprint, Barbara, Boat, Couple, House, Peppers**.
- Parameter $\mu = \frac{1}{200}$, which is used for averaging of noisy picture with reconstructed one.
- Patch size of 8x8 pixels.
- Error-constrained OMP instead of sparsity-constrained, as done in original paper and not as we done during the course. Hence, we introduced a gain as a parameter for both algorithms, and error bound for OMP was computed as gain times σ multiplied by size of patch (its side length).
- Various gain values – described below.
- 20 Iterations of K-SVD and then giving the resulting dictionary to “Sharing the disagreement” as initial dictionary.
- Performing 30 iterations of “Sharing the disagreement”, where every iteration has two Sparse Coding steps and two Dictionary Update steps.

After several tries, we reached the best results which we summarized in Table 2.

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

Table 2. Our results of implementing and running K-SVD and “Sharing the disagreement” for denoising images. We denote here original noise level as σ, the noise level we gave as parameter to OMP as $\hat{\sigma}$, result of K-SVD as “Orig” and result of “Sharing the Disagreement” as “New” – in PSNR. The “Average” is the average improvement in PSNR between tested images.

For every noise level, we iterated through various values of gain between 1.0 to 1.2 to find the one which makes the “Sharing the disagreement” deliver the best denoising result, as authors of original paper done. In original paper, as noise level becomes bigger, the best gain becomes smaller, and you can reach from the table that we have the same property for gain. However, the best gain varies from original one, for every sigma – in our case it is slightly bigger (between 1.12 to 1.16, while in original paper it was between 1.02 to 1.12). Regarding lack of improvement, the only one case where in original paper there were no improvement is the case for Barbara and $\sigma = 10$, while in our case the no improvement case was with Fingerprint, but for same noise level. Regarding average result, in general, the bigger the initial noise level, the better the improvement, and in our case the average improvement is bigger than in original paper. The example of original DCT dictionary, learned by K-SVD and reached after using Algorithm 1 and denoising results are shown on Figure 5.

a)

b)

Figure 5. Containing two parts:

a) Original image Barbara, its noisy version with σ=20, and denoised by K-SVD and Algorithm 1.

b) The original DCT dictionary, learned by K-SVD and reached after using Algorithm 1, for image Barbara, σ=20, gain 1.16. The difference between three dictionaries is not negligible at all

We think that all the inconsistencies with original paper's results can be caused by fact that denoising steps, exact input images (and their resolutions) and implementation of both algorithms may vary between ours and original paper authors' ones. At least, authors of "Sharing the disagreement" used the very optimized and precisely configured base version of K-SVD (which can be taken via link from Michael Elad's site), while we implemented both algorithms on our own, without a big amount of options and not as fast and tuned as it can be.

As a whole picture, results we got in our experiment are very similar to original ones. You can get them evaluated and printed by running the course_project_short_demo.m and course_project_full_run.m from our GitHub repository.

5 OUR PROPOSAL OF PARAMETRIZATION

Impressed by PSNR increase made by "Sharing the Disagreement", we turned to question how the disagreement-patches values influence the denoising result. We going to introduce the parameter we named just α (alpha), which is going to be an additional multiplier of disagreement-patches, i.e. in original algorithm its value always equals 1.

That is, we change Disagreement-Update step in Algorithm 1 to be:

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

i.e. for every computed disagreement-patch, we multiply it by α.

From this definition we can reach that if $\alpha = 0$, then the resulting algorithm is like the K-SVD, because of fact that patches are not considered at all and equal to zero. Otherwise, if $\alpha = 1$, the resulting algorithm is exactly Algorithm 1, i.e. "Sharing the disagreement". We decided to try various values between 0 and 2, with step of 0.1. Then, we took two very different images – Barbara and Fingerprint, and tried these various values of α in modified Algorithm 1 in comparison to K-SVD in same conditions as for experiments in previous chapter, for fixed concrete σ=20 and gain 1.16.

The obtained results are on Figure 6.

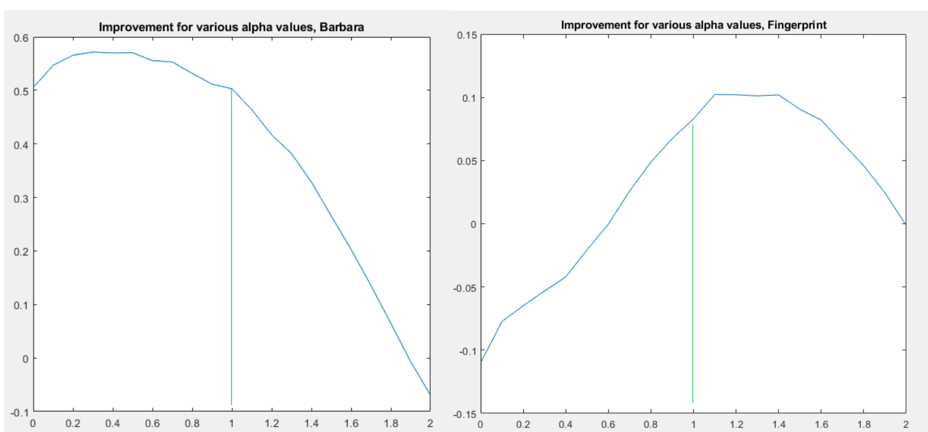


Figure 6. Result of iterating on various values of α for images Barbara and Fingerprint, where X axis is value of α used, and Y axis is the improvement in PSNR (in dB) compared to K-SVD as before, for σ=20 and gain 1.16. Green line points at improvement for $\alpha = 1$, i.e. the case from original paper.

The results we obtained were very interesting. First, we see that using $\alpha = 1$ i.e. using the "Sharing the disagreement" as-is – is not the best choice we can do, and in both cases, there is still a space for improvement up to 0.1 more in terms of PSNR. It may sound little, but may be reasonable, and can be used together with other techniques like SOS boosting [9].

Second, the α providing the peak improvement value was not the same for both pictures – about 0.4 for Barbara and about 1.3 for Fingerprint. It can be possibly caused by fact that these images are very different in their structure and textures.

Wondering about the average case, we ran the very same piece of code evaluating improvement for various values of α, on every one of 6 test images we had in previous experiments (Fingerprint, Barbara, Boat, Couple, House, Peppers) and made an average for obtained results (still with fixed concrete σ=20 and gain 1.16). You can get them evaluated and showed by running the seeking_alpha_on_average.m from our GitHub repository. Results which we got are on Figure 7.

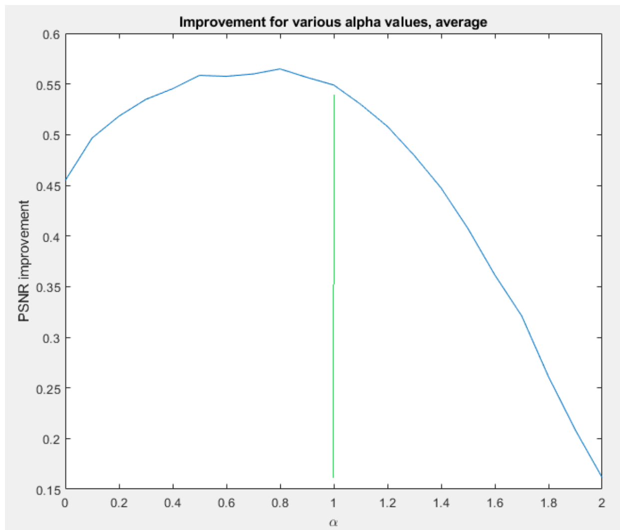


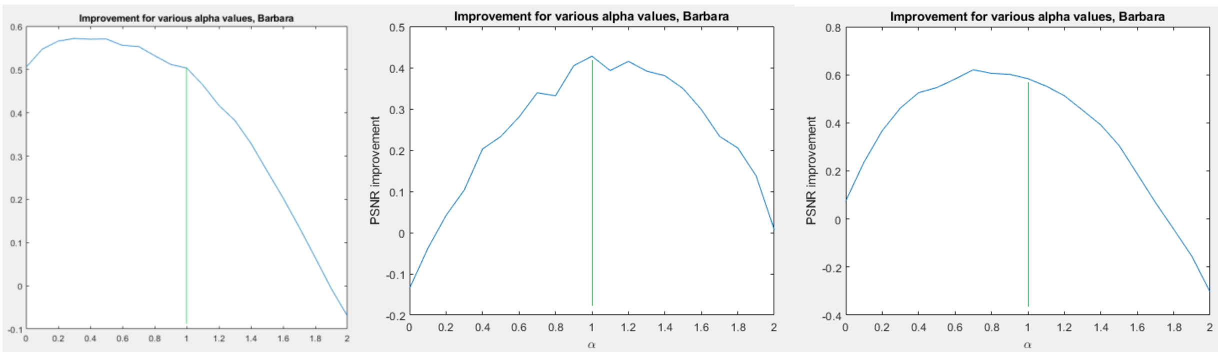
Figure 7. Result of iterating on various values of α all test images, where X axis is value of α used, and Y axis is the improvement in PSNR (in dB) compared to K-SVD as before, for σ=20 and gain 1.16.

In such an average case as we ran for Figure 6, the best value of α is around 0.8, and not 1, giving an average improvement of 0.05 dB, which is small, but still pointing that the result of original Algorithm 1 can be improved using $\alpha \neq 1$.

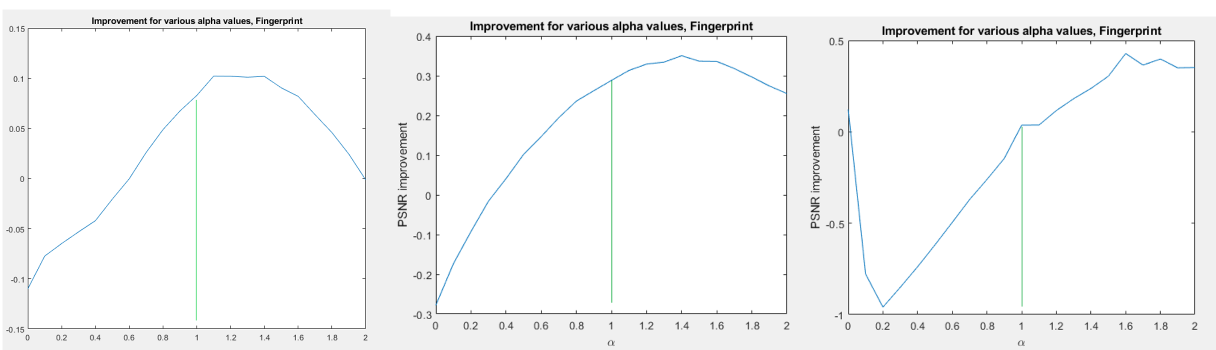
Analyzing graphs from figures 6 and 7, we can reach that that if we slightly tune the α per specific picture, we can get additional benefit of about 0.05 and more, depending on the specific image. Moreover, every such a graph has a form of "bell", possibly with several local maximums – for alpha between 0 and 2.

Wondering if alpha is correlated to noise level σ in some way, we ran the same piece of code as before on images of Barbara and Fingerprint, and then on all images taking average, for various values of α and σ, to find the best ones and check the correlation between them. Results of this experiment are displayed on Figure 8: for image Barbara best values of α were between 0.4 to 1, without correlation to σ. In case of Fingerprint, best values of α were between 1 to 1.4, growing with σ – as we thought, we reached result that vary from this of Barbara, possibly due to other kind of textures on test image. Running the same experiment on all images and averaging the improvement, the best values of α were between 0.6 to 1, without correlation to σ.

a) Barbara: $\sigma = 20$ $\sigma = 50$ $\sigma = 75$



b) Fingerprint: $\sigma = 20$ $\sigma = 50$ $\sigma = 75$



c) Average: $\sigma = 20$ $\sigma = 50$ $\sigma = 75$

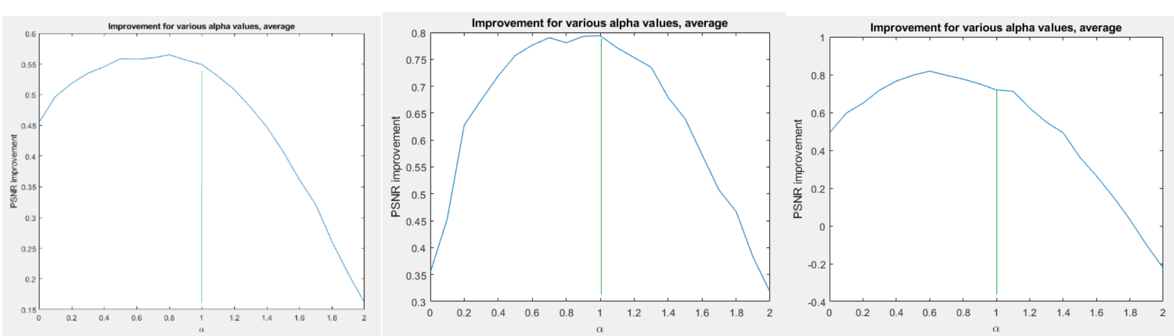


Figure 8. Result of iterating on various values of α between 0 and 2, and various values of σ where X axis is value of α used, and Y axis is the improvement in PSNR (in dB) compared to K-SVD as before, for gain 1.16, for: a)Barbara, b)Fingerprint, c) Average for 6 test images.

As a partial conclusion from these experiments, we propose to either select α to be around 0.8 (average between 0.6 and 1 as in part c of Figure 8) to get the average benefit, or iterate for several values between wider borders and visually select the best result (if it is possible, because we found that the average benefit can be reached is about 0.05 dB, and is hard then to distinguish the best one. But, for example, for $\sigma = 75$ in average case the improvement is about 0.1 if we choose $\alpha = 0.6$ instead of 1).

6 CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we summarized the original paper "Patch-Disagreement and a Way to Improve K-SVD Denoising" by Elad and Romano, added comments to the summary regarding what we learned during the course, implemented the both original K-SVD from [5] and "Sharing the disagreement" from original one, tried to do the same experiment as in original paper and get the same result, and 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 in most cases is not the best one and can be tuned to reach additional benefit is PSNR, while the average best value for α between tested images is around 0.8. Average possible benefit in PSNR is about 0.05-0.07 dB which can be significant.

Possible future direction can be is to build some kind of texture analyzer, possibly AI-based, which can predict the near-best value of α – we think that Figure 5 shows that best value of α vary depending on image textures, pay attention on the difference between best values. If α selected incorrectly, we can get PSNR decrease instead.

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.