236862 Sparse and Redundant Representations

Course Project

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

Maxim Lipatrov Nerya Hadad

[maxim-li@campus.techion.ac.il](mailto:maxim-li@campus.techion.ac.il) [neryahadad@gmail.com](mailto: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

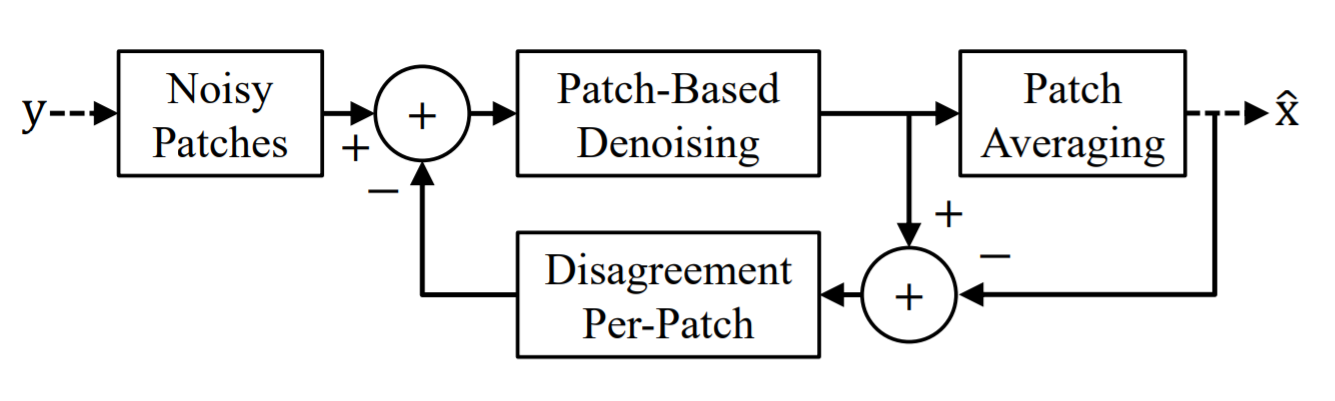
where x is an original image, y is a given known (deteriorated) image and v is an additional zero-mean Gaussian noise independent to x. Why Gaussian? Gaussian noise is the popular one in image processing because of fact that digital images gets it from the camera sensor due to, for example, low light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

/\* Different works addressing the local-global gap… ?\*/

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. We 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./\*TODO\*//\*now back to 2? That’s what I did in the powerPoint\*/

In section /\*TODO\*/ we propose to introduce hyper-parameters 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.

**SPARSELAND AND K-SVD**

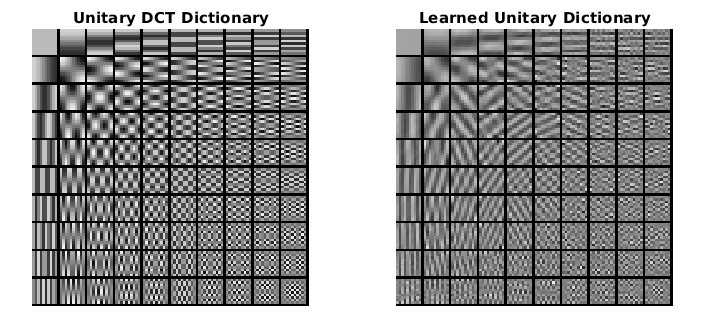
“Sparseland” model of algorithms assumes that a signal can be represented by a very few atoms of a redundant dictionary. Denote the dictionary as D and the original input signal as x, then:

where is a sparse vector, and where leading to redundancy of . Now, using this model in order to estimate the original input signal and given deteriorated one , we can find an estimation of – let’s denote it as – the projection of onto the set of low-dimensional subspaces that spans (recall that is sparse) and then obtain the denoised signal by:

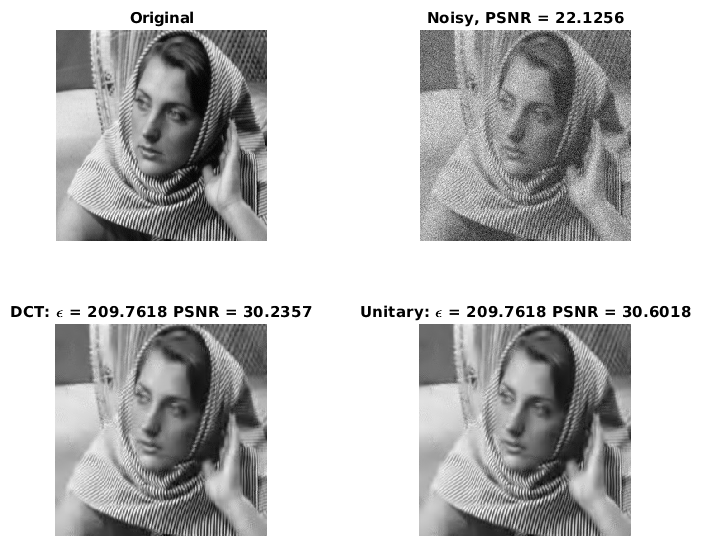
Now, define as the maximal error bound that we can treat, we will seek for - solution of:

where is an operator counting the number of non-zeros, i.e. the -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.*



*Figure 3. Original part of Barbara image, its noisy version, denoised by DCT version and denoised by learned DCT dictionary, 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:

where the resulting is a learned updated dictionary and 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 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:

where denotes the number of image patches, and the matrix extracts the i'th patch from the global image.

**THE PROPOSED ALGORITHM**

According to K-SVD algorithm, each patch is treatment 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:

where is the locally denoised result (on place of patch ) and is the corresponding part from global estimate, both at iteration. We described before that sparse coding does not consider relations between patches, hence 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 from – 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:

where is local method-noise patch obtained at iteration.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Algorithm 1: Sharing the disagreement**

Initialization:

1. – an initial dictionary.

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

1. Sparse coding step: using the OMP, solve:
2. Dictionary update step, solve:

In practice, is obtained from by K-SVD.

1. Image reconstruction step, solve:

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

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

and then set .

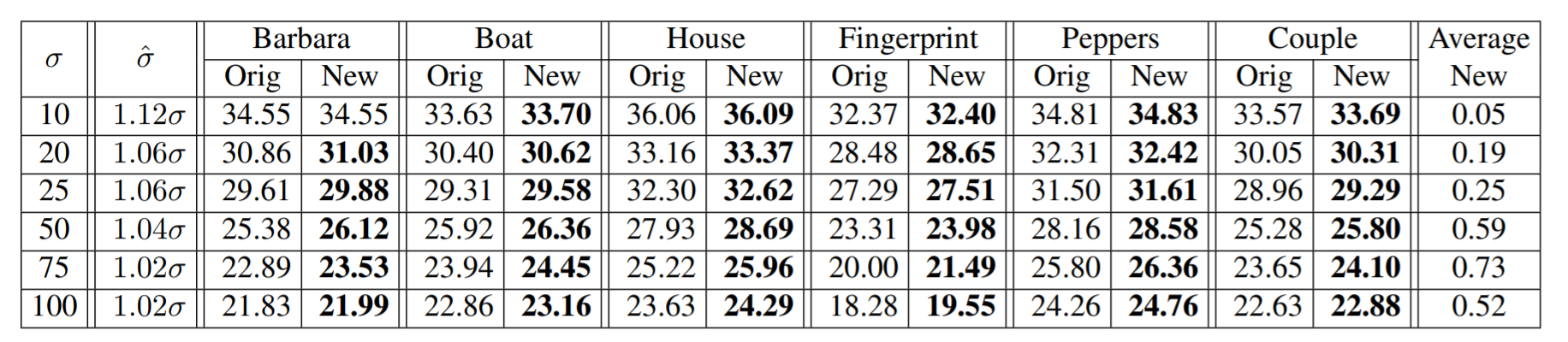
Output: – the last result.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The algorithm tries to recover a patch from the global estimation , corrupted by the method-noise patch . 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.

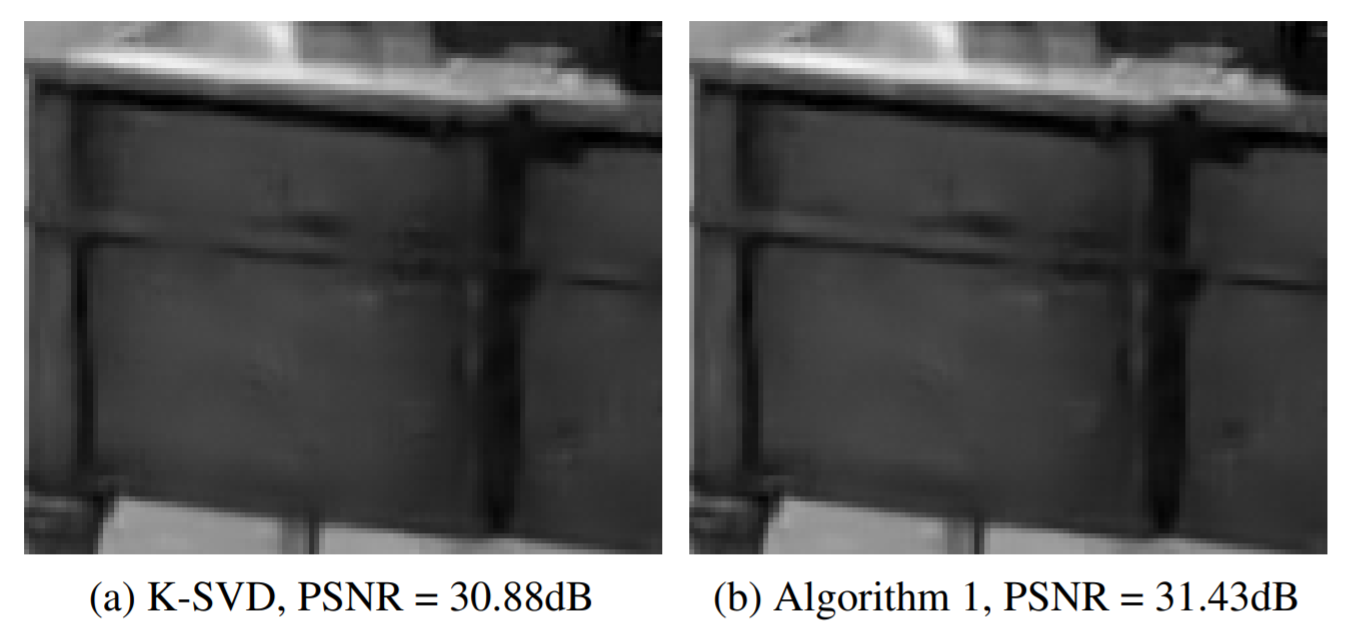
**EXPERIMENTS WITH THE PROPOSED ALGORITHM**

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 as described in (1). Results were compared via Peak Signal to Noise Ratio (PSNR) defined as , where 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 (5), which must be larger than , then this bigger noise energy 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).

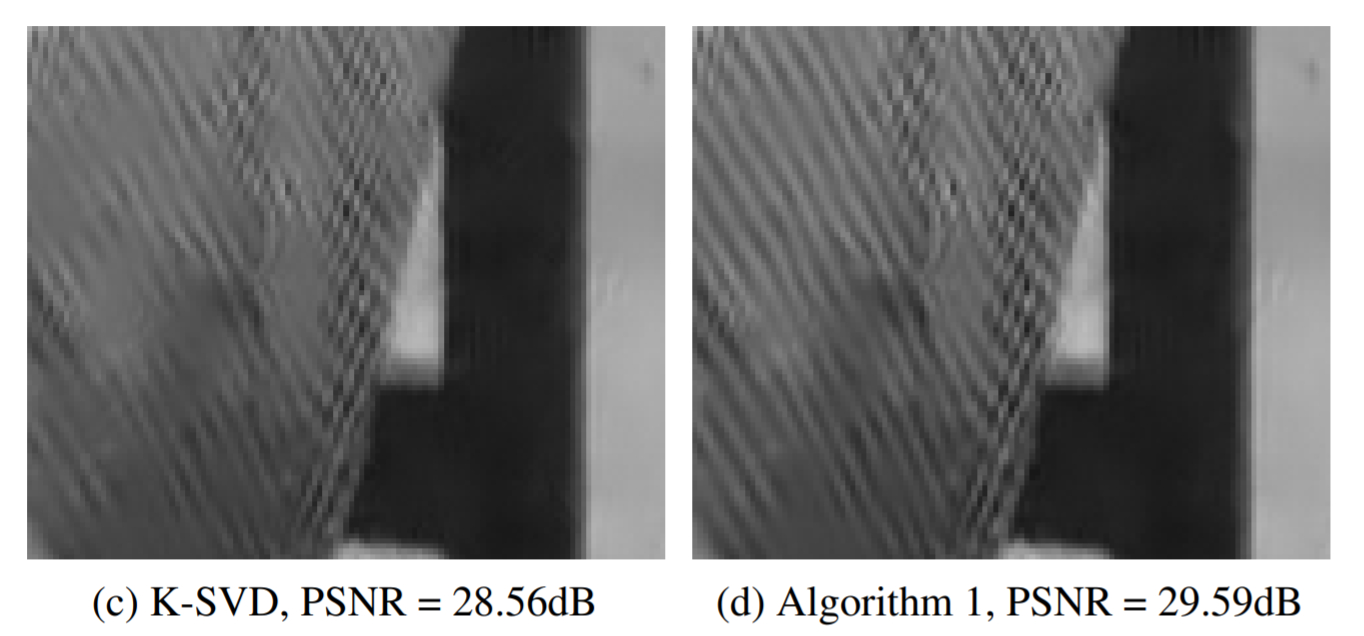


*Table 1. Comparison between the denoising results [PSNR] of the original K-SVD algorithm [5] and its ”sharing the disagreement” outcome (Algorithm 1). The best results per each image and noise level are highlighted.*

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. We’ll use the same approach of comparison for evaluation of proposed addition of parametrization to “sharing the disagreement” algorithm described above.

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.