Ukrainian Catholic University

Faculty of Applied Sciences

Data Science Master Programme

K-SVD algorithm for image denoising

Linear Algebra final project report

Authors:
Bohdan Matviiv
Dmytro Babenko

24 January 2019

Image denoising with use of K-SVD

Abstract: Image denoising is an actual and practical issue for different domains / industries and its algorithms are constantly improving in different aspects. For given paper we unveiled given issue by describing its background and internals, overviewed K-SVD as our denoising algorithm, explained its key points, solution, advantages and disadvantages, and compared it with basic denoising algorithms.

Key words: denoising, image denoising, image denoising algorithm, SVD application, K-SVD

I. Introduction

Currently digital image capture is a widespread technology used in different aspect of people lives. From medicine to media, from military to science digital photography become a pillar for our communication and understanding of our world. While the technology is rapidly improving in hardware, especially with development of smartphones and other technologies like drones or autonomous car systems, it still has some drawbacks and limitations. First of all consumer electronics like smartphones uses image sensor of very small size due miniaturization. In physical aspect, it means that such sensor captures less light, which leads to appearance of skewed images and unwanted signals – or as more common named – image noise. While in a given case the issue is more related to commercial interest, there are other domains where noise appears as an important factor like radio astronomical images, tissues microscope photo imagery, computer vision for autonomous car or UAV video capture from battlefield. All these examples are heavily relied on image clarity and precise interpretability while having complicated conditions for making the image due different factors.

As it was stated before, image noise is an unwanted signal of light, which was wrongly captured by a camera sensor or skewed during its transmission. It can be described as a random set of brightness or colors for random pixels or block of pixels.

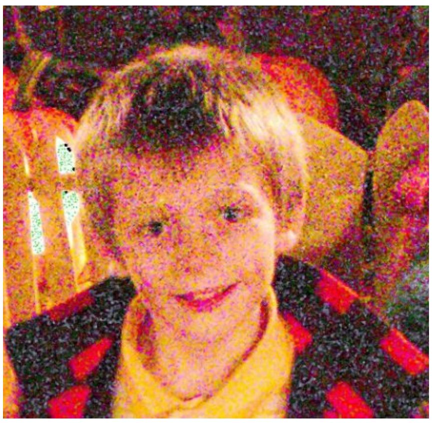


Figure 1 - Real life example of noise on image taken in low light condition

On the Figure 1 we can observe example of noisy image taken by a conventional camera in a low light condition. It has a lot of wrongly defined pixels which results in a low quality. From consumer viewpoint it can lead to unsatisfactory results and experience which leads to drop in product sales and profits.

Another example of denoising is drone footage, as example on Figure 2.

Figure 2 - UK RAF drone footage

Unmanned aerial vehicles (or drones) are operated by visual guidances from their camera . It is also used for detecting and aiming at targets. Footage is transmitted from the drone camera to the operator by satellite or radio and the process has few bottlenecks. First of all low light condition can worsen image quality. Also opposing force can use jammers to block drone operability or tamper signals. Also the communication channel itself can be worsened by natural causes like weather, geomagnetic storm, etc. All these issues can lead to noise in transmitted image which can lead to worsening drone operability.

Third example can be referenced from medicine domain. Figure 3 contains ultrasound image of human pancreas

Figure 3 - Ultrasound image of pancreas

This is also a good example of dealing with aspects of image denoising - due technical and physics limits output image could be not as clear and informative as end users want so in a lot of cases it should be processed by denoising algorithm for obtaining beneficial information in healthcare domain.

II. Problem setting

As it was stated and shown earlier, image noise is an important issue in different domains. So image denoising as an instrument can bring a lot of beneficial value to end users. To understand approaches and solution for image denoising we need also to acknowledge with noise nature and its types.

Noise is also composed of two elements: fluctuations in color and luminance (Figure 4). Color or "chroma" noise is usually more unnatural in appearance and can render images unusable if not kept under control. The example below shows noise on what was originally a neutral grey patch, along with the separate effects of chroma and luminance noise. [noises]

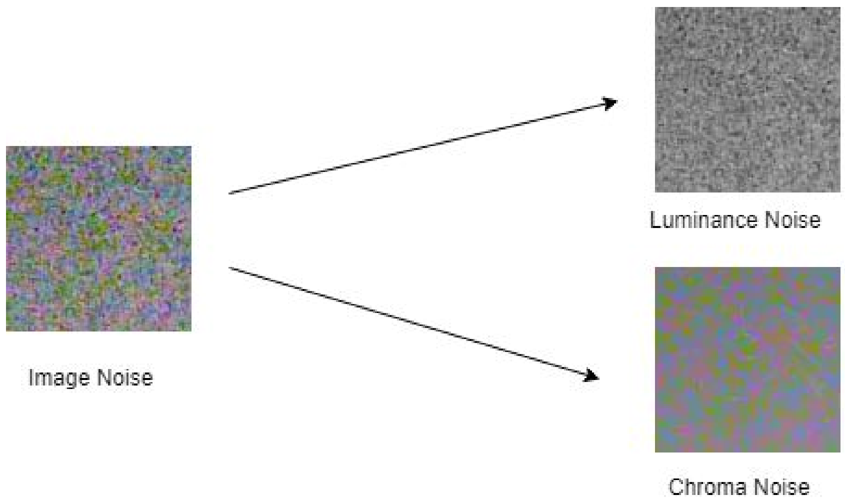


Figure 4 - Basic components of image noise

Noise reduction can be used to selectively reduce both chroma and luminance noise, however complete elimination of luminance noise can result in unnatural or "plasticy" looking images.

There are a set of different noise models :

– Gaussian noise

– Rayleigh noise

– Erlang (Gamma) noise

– Exponential noise

– Impulse (salt-and-pepper) noise

– Uniform noise

An image gets corrupted with different types of noise. Noise may be classified as substitutive noise (impulsive noise:salt and pepper noise,random valued impulse noise,etc.) and additive noise (e.g. additive white Gaussian noise). [noise_types]

These models differ in noise distribution, their visual representation and image resulting histogram is displayed on figure 5.1 and 5.2

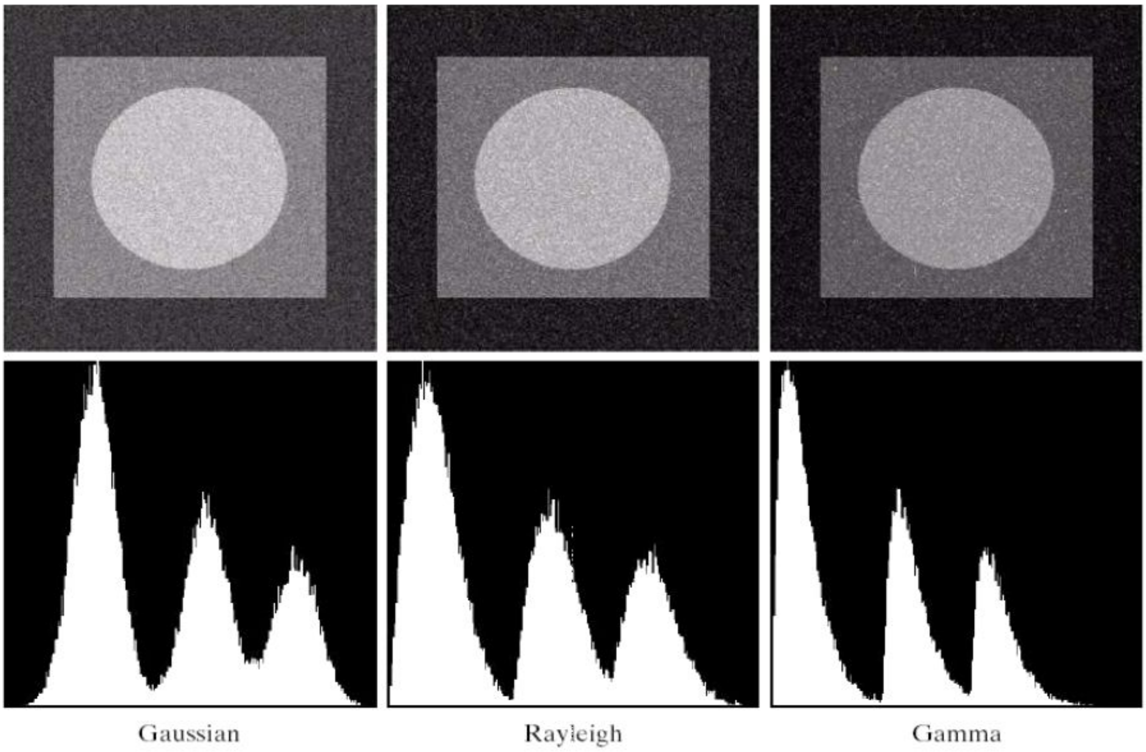


Figure 5.1 - Visual representations of different noise types and their image histograms

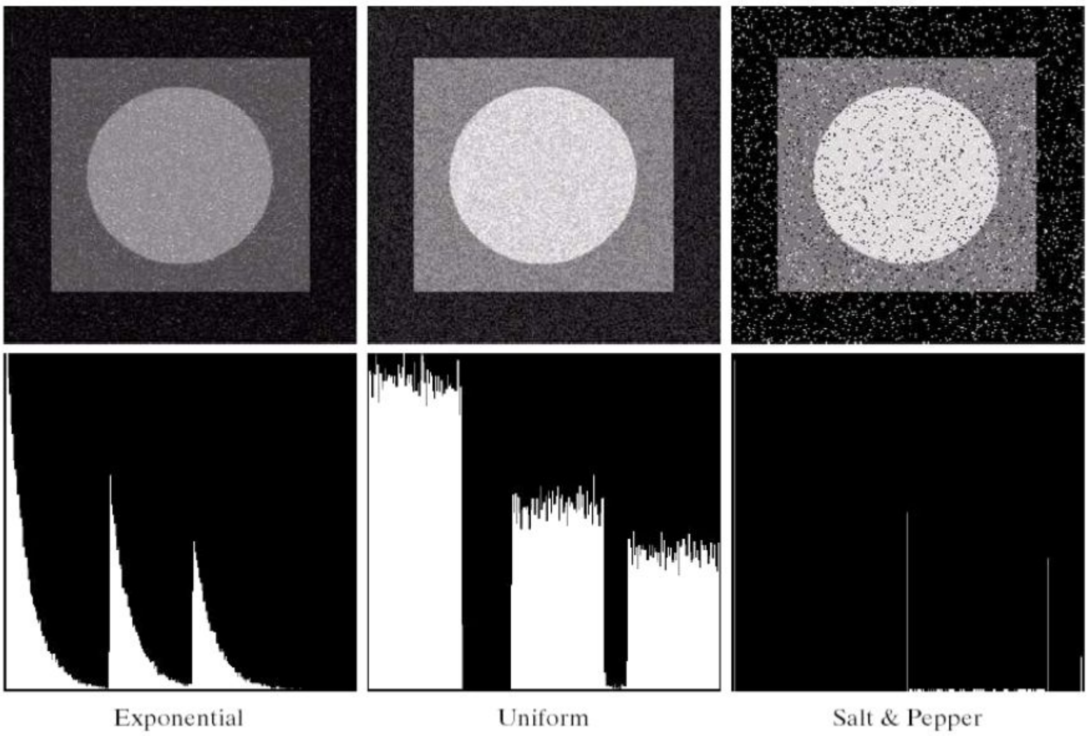


Figure 5.2 - Visual representations of different noise types and their image histograms

III. Related work

There are a set of alternative methods previously used as image denoising algorithm :

- Spatial Filtering is denoising method based on changing intensities of the neighboring pixels. Using spatial filtering, the image is transformed (convoluted) based on a kernel H which has certain height and width (x,y), defining both the area and the weight of the pixels within the initial image that will replace the value of the image. [spatial_filtering]
- Linear Filters process time-varying input signals to produce output signals, subject to constraint of linearity. A mean filter is the optimal linear filter for Gaussian noise in the sense of mean square error. Linear filters too tend to blur sharp edges, destroy lines and other fine image details, and perform poorly in the presence of signal-dependent noise. The wiener filtering method requires the information about the spectra of the noise and the original signal and it works well only if the underlying signal is smooth. Wiener method implements spatial smoothing and its model complexity control correspond to choosing the window size. [linear]
- Mean Filter is a simple, intuitive and easy to implement method of smoothing images, i.e. reducing the amount of intensity variation between one pixel and the next. It is often used to reduce noise in images. The idea of mean filtering is simply to replace each pixel value in an image with the mean (`average') value of its neighbors, including itself. This has the effect of eliminating pixel values which are unrepresentative of their surroundings. Mean filtering is usually thought of as a convolution filter. Like other convolutions it is based around a kernel, which represents the shape and size of the neighborhood to be sampled when calculating the mean. Often a 3×3 square kernel is used, as shown in Figure 1, although larger kernels (e.g. 5×5 squares) can be used for more severe smoothing. (Note that a small kernel can be applied more than once in order to produce a similar - but not identical - effect as a single pass with a large kernel.) [mean]
- Non-linear filters have many applications, especially in removal of certain types of noise that are not additive. Generally spatial filters remove noise to a reasonable extent but at the cost of blurring images which in turn makes the edges in pictures invisible. In recent years, a variety of nonlinear median type filters such as weighted median, rank conditioned rank selection, and relaxed median have been developed to overcome this drawback. [linear]
- Non Local Means filtering is based on estimating each pixel intensity from the information provided from the entire image and hence it exploits the redundancy caused due to the presence of similar patterns and features in the

image. In this method, the restored gray value of each pixel is obtained by the weighted average of the gray values of all pixels in the image. The weight assigned is proportional to the similarity between the local neighborhood of the pixel under consideration and the neighborhood corresponding to other pixels in the image [non_local].

IV. Approach to solution

Our picked topic is the K-SVD algorithm application in improving image quality. In the basics, K-SVD represents a dictionary-learning algorithm, which creates a dictionary with use of singular value decomposition. K-SVD is the generalization of the k-means clustering method, and it works by iteratively alternating between sparse coding the input data based on the current dictionary, and updating the atoms in the dictionary to better fit the data.

The basic model suggests that natural signal can be efficiently explained as linear combinations of prespecified atom signals, where the linear coefficients are sparse (most of them are zero) [ksvd_explain].

One of the main goal of our topic to use K-SVD for reducing noise from the image.

K-SVD is used to solve the problem of how to select a set of basis vectors for the efficient representation of signals in a given dataset.

Given an original signal y in an n-dimensional space and a set of basis vectors, we need to find a compact representation of y using the subspace spanned by the basis vectors.

K-SVD advantages and disadvantages

K-SVD used for dictionary-based learning has a problem dealing with high-dimensional signals. Because of that, it requires large storage because the computed non-zero coefficients reside in a different location [ksvd_explain]. Also, K-SVD operates by an iterative update which does not guarantee to find the global optimum [ksvd_cons].

K-SVD algorithm

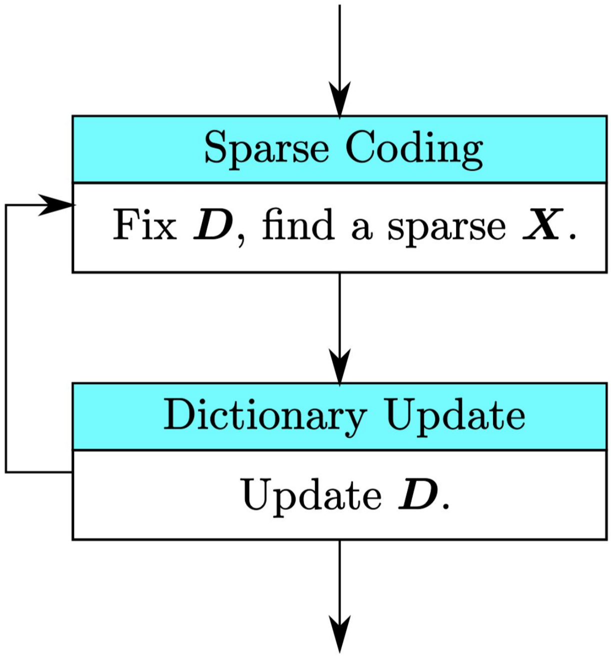
The algorithm accepts initial overcomplete dictionary D, a number of iterations, a set of training signals arranged as the columns of the matrix X. K-SVD is a kind of generalization K-means, as follows. The K-means clustering can be also regarded as a method of sparse representation. That is, finding the best possible codebook to represent the data samples $\{y_i\}^M_{i=1}$ by nearest neighbor, by solving

$$min\{\|Y - DX\|^2_F\} \text{ subject to all } i \quad x_i = e_k \text{ for some } k.$$

which is equivalent to

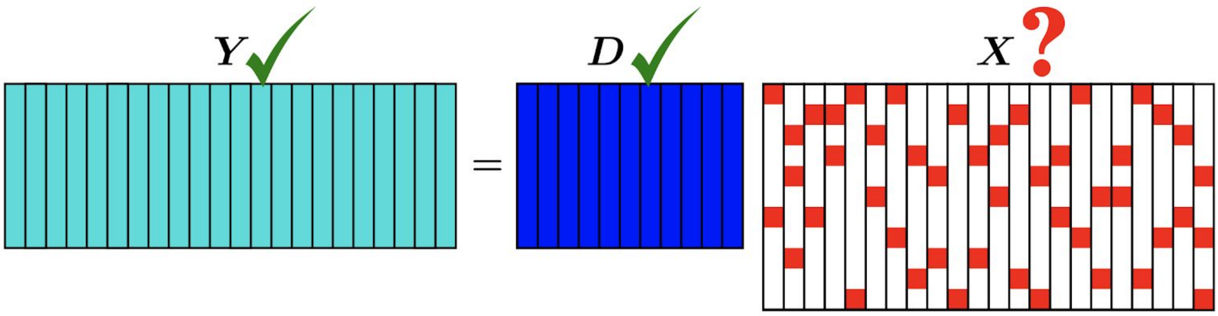
$$min\{\|Y - DX\|^2_F\} \text{ subject to all } i \ \|x_i\|_0 = 1 .$$

In general, the algorithm consists of two stages - sparse coding and dictionary update



Sparse coding.

Fix matrix D (usually at first time it is selected randomly with normalized columns), solve for X in $min\{\|Y - DX\|^2_F\}$ subject to for all i $\|x_i\|_0 \leq T$. Here we find the sparse matrix X is a matrix which most of elements are zero.



There are few methods how to find the matrix X such as

- Orthogonal Matching Pursuit (OMP)
- Subspace Pursuit (SP)
- Iterative Hard Thresholding (ITH)

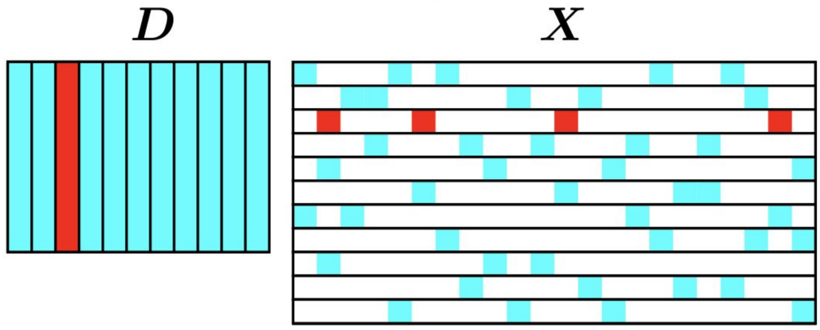
In most cases, OMP method is selected.

Dictionary update

For each k-atoms (columns of matrix D), update this column in D and the corresponding row in X (see Figure below). Also, fix other columns in D and the corresponding rows in X.

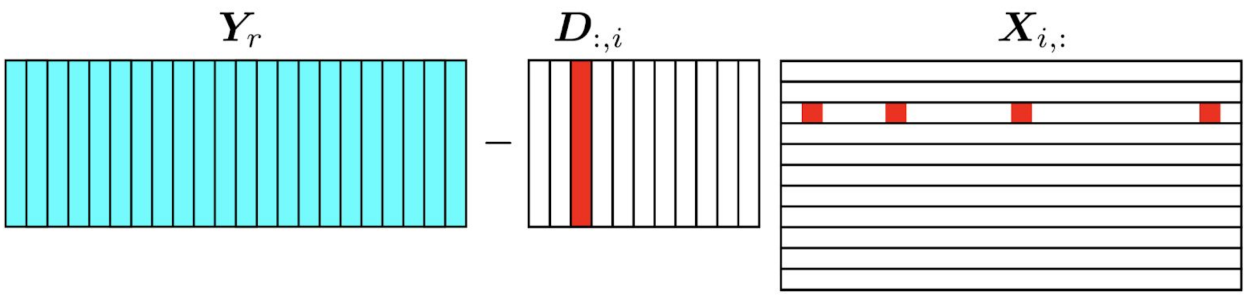
$$\Omega = \{(i,j) : X_{i,j} \neq 0\}$$

$$X_\Omega = \{X : X_{i,j} = 0, \ \forall \ (i,j) \ \in \ \Omega^C\}$$



In this case we can write distance which we want to minimize in the following way

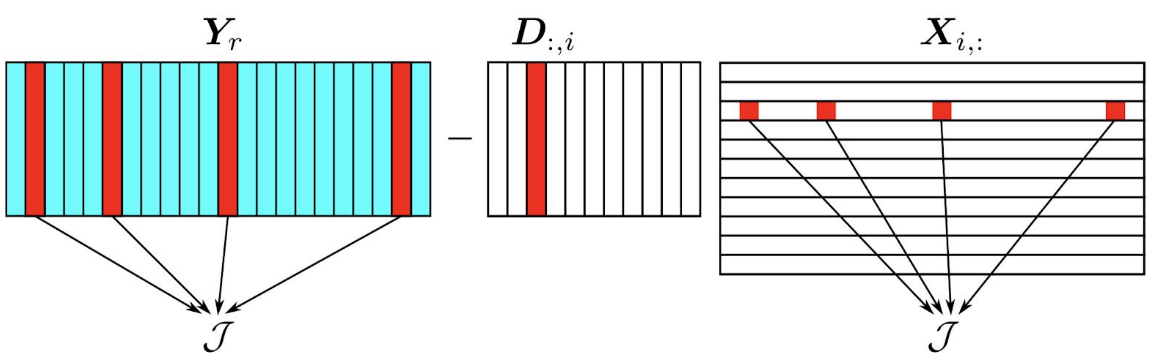
$$\|Y - DX\|^2 = \|Y - D_{:,j\neq i}X_{j\neq i,:} - D_{:,i}X_{i,:}\|^2 = \|Y_r - D_{:,i}X_{i,:}\|^2$$



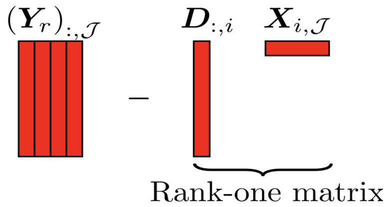
In another way we can write

$$\|Y - DX\|^2 = \|Y - D_{:,j\neq i}X_{j\neq i,:} - D_{:,i}X_{i,:}\|^2 = \|Y_r - D_{:,i}X_{i,:}\|^2 =$$

$$= \|(Y_r)_{:,J} - D_{:,i}X_{i,J}\|^2 + c$$



It is clear to notice that matrix $D_{:,i}X_{i,J}$ is rank-one matrix.



Here, SVD helps us to use optimal rank-one approximation

$$A = \Sigma\lambda_i u_i v_i^T \approx \lambda_1 u_1 v_1^T$$

As we update the dictionary (matrix D), the column of *D,* which we consider at current iteration would be update by the first column of matrix *U* from SVD of matrix Y_r : $D_{:,i} = u_1$.

Sparse coding and update dictionary repeats until we reach necessary tolerance or passed the maximum number of iterations.

The pseudo-code of K-SVD algorithm described on the Figure 2.

function $[\mathbf{D}, \mathbf{X}] = \text{ksvd}(\mathbf{Y}, \mathbf{D}, K, T, \epsilon, N)$
$\mathbf{y} \Leftarrow$ observed signal to encode of size $n \times 1$
$\mathbf{D} \Leftarrow$ dictionary of atoms of size $n \times K$
$T \Leftarrow$ sparsity threshold
$\epsilon \Leftarrow$ error tolerance
$K \Leftarrow$ number of atoms in dictionary
$N \Leftarrow$ number of k-svd iterations to run

for $i < N$ **do**
  $\mathbf{X} = \text{OMP}(\mathbf{D}, \mathbf{Y}, \epsilon, T)$
  for $j <= K$ **do**
    idx $\Leftarrow$ all non-zero indices of $\mathbf{x}_j^r$
    if idx is empty **then**
      $\mathbf{E} \Leftarrow \mathbf{Y} - \mathbf{DX}$
      $p \Leftarrow$ position where error vector has max norm
      $\mathbf{x}_j^r \Leftarrow 0$ % Deselect j-th atom
      $\mathbf{d}_j \Leftarrow \frac{\mathbf{y}_p}{||\mathbf{y}_p||_2}$
    else
      lst $\Leftarrow \{\mathbf{x}_k | k \in \mathbf{idx}\}$
      $\mathbf{x}_j \Leftarrow 0$ % Deselect the j-th atom
      $\mathbf{E}_k^R \Leftarrow \mathbf{Y}_{\mathbf{lst}} - \mathbf{DX}$
      $[\mathbf{U}, \mathbf{S}, \mathbf{V}^T] \Leftarrow \text{SVD}(\mathbf{E}_k^R)$
      $\mathbf{x}_{j,p}^r \Leftarrow \mathbf{S}_{1,1}\mathbf{V}_1 \;\; \forall \;\; p \in \mathbf{idx}$
      $\mathbf{d}_j \Leftarrow \frac{\mathbf{U}_1}{||\mathbf{U}_1||_2}$
    end if
  end for
end for

Return $\mathbf{D}, \mathbf{X}$

Figure 2 - Pseudo-code of K-SVD algorithm [pseudo_code]

Simple Applications to use k-SVD for denoising

There is matrix $Y_{features\ x\ targets}$ which describes information about some image. In simple case, K-SVD algorithm can be run for smaller matrix $Y_{features\ x\ M}$ ($M < targets$) and find D, X. After that we can find new matrix X using sparse coding for $Y_{features\ x\ targets}$ and D as input parameters. Finally we can calculate the matrix which contains information about denoised image. $Y_{denoise} = DX$.

V. Evaluation

Human factor is interesting aspect of image signal denoising evaluation in comparison with sound signal denoising or any other. Because the results are visible, easy interpretable and can used as a final product, people tend to evaluate denoising rather by aesthetical.

From the Figures 12, 13, 14 below we can notice how noise was reduced using K-SVD algorithm and also there were measured PSNR which accepts human decision about denoising.

Figure 12 - Coffee example. From left to right: original image, noisy image ($psnr = 20.6\ dB$), reduced noise ($psnr = 25.2\ dB$) respectively

Figure 13 - Chelsea example. From left to right: original image, noisy image ($psnr = 20.1\ dB$), reduced noise ($psnr = 29.9\ dB$) respectively

Figure 14 - Lenna example. From left to right: original image, noisy image ($psnr = 20.2\ dB$), reduced noise ($psnr = 28.6\ dB$) respectively

There are comparison K-SVD denoising with Non-local means denoising algorithm on Figure 15, 16. In provided examples, peak signal-to-noise ratio prefers K-SVD algorithm, but human conclusion can be different.

Non-local means ($psnr = 25.5\ dB$) K-SVD ($psnr = 28.6\ dB$)

Figure 15 - Comparing Non-locals mean and K-SVD (Lenna example)

Non-local means ($psnr = 27\ dB$) K-SVD ($psnr = 29.9\ dB$)

Figure 16 - Comparing Non-locals mean and K-SVD (Chelsea example)

VI. Conclusion

There were analyzed and implemented application of basic K-SVD algorithm for image denoising. From the Figure 12-14 we can notice how noise were reduced from noisy image and can compare with original images. For metric estimation, we used peak signal-to-noise ratio. There were comparison between Non-local means and K-SVD algorithm and PSNR tell us that K-SVD gave better result (reduced more noise).

Although given method produces acceptable quality it can be further improved with K-SVD modifications.

References

[noises] - Sean McHugh “DIGITAL CAMERA IMAGE NOISE”, 2015. https://www.cambridgeincolour.com/tutorials/image-noise-2.html

[noise_types] - Sontakke, Mandar D., and Meghana S. Kulkarni. "Different Types of Noises in Images and Noise Removing Technique." IJATES 3.1 (2015).

[spatial_filtering] - Athanasiou, Lambros S., Dimitrios I. Fotiadis, and Lampros K. Michalis. Atherosclerotic Plaque Characterization Methods Based on Coronary Imaging. Academic Press, 2017.

[non_linear] - Buades,A.,Coll,B.,Morel,J.M., “A non local algorithm for image denoising”, Computer vision and Pattern Recognition, 2005,CVPR 2005, IEEE Computer Society Conference on Vol 2,pp,60-65,2005.

[linear] - Kaur, Sandeep, and Navdeep Singh. "Image Denoising Techniques: A Review." International Journal of Innovative Research in Computer and Communication Engineering 2.6 (2014).

[mean] - Kumar, R. N., and U. J. Kumar. "A spatial mean and median filter for noise removal in digital images." Int J Adv Res Electr Electron Instrum Eng 4.1 (2015).

[ksvd_algorithm] - Wei Dai, Boris Mailhé, Wenwu Wang. "Dictionary Learning for Sparse Representations Algorithms and Applications" Imperial College London Queen Mary University of London University of Surrey, 2013.

[ksvd_explain] - Yang, J., Bouzerdoum, A. & Phung, S. (2009). A new approach to sparse image representation using MMV and K-SVD. In J. BlancTalon, W. Philips, D. Popescu & P. Scheunders (Eds.), Advanced concepts for intelligent vision systems : ACIVS 2009 (pp. 200-209). Berlin, Germany: Springer-Verlag.

[ksvd_cons] - Rubinstein, R., Bruckstein, A.M., and Elad, M. (2010), "Dictionaries for Sparse Representation Modeling", Proceedings of the IEEE, 98 (6): 1045–1057

[pseudo_code] - Chatterjee, Priyam, and Instructor-Prof Peyman Milanfar. "Denoising using the K-SVD Method." *Course Web Pages, EE*264 (2007): 1-12.

[ksvd_implementation] - https://github.com/DmytroBabenko/DenoiseImage_KSVD