A Comparison of K-SVD with Traditional Denoising Algorithms

Paola Ardón, Jose Bernal, Rodrigo Daudt, Sepehr Mohaimanian, Èric Pairet*, Armine Vardazaryan
Department of Electrical Engineering, University of Burgundy
Le Creusot, France
Email: *eric.pairet@u-bourgogne.com

***Abstract*—The advance of new technologies of this century has allowed to develop tools for medical process. In particular, analysis of retinopathy images has become one of the main ways to early detect some Non-Communicating Diseases, such as diabetes. The main problem of these kind of images is that noise is added during the acquisition step. This noise may imply wrong medical resolutions and, hence, human life may be affected. Thus, improving the quality of the inputs before processing is an important task. In this paper, a comparison of state-of-the-art denoising methods as well as the K-SVD algorithm is presented. The evaluation was performed on synthetic images and SD-OCT images. Results show that K-SVD is a suitable tool for denoising SD-OCT images since it is able to decrease the noise while preserving edges.**

***Index Terms*—Denoising algorithms, K-SVD, retinopathy images, OCT segmentation**

I. Introduction

Nowadays, one of the main challenges of medicine and engineering is to develop tools for supporting the medical diagnosis. According to the World Health Organization (WHO), about 15% of the population's deaths between 2010 and 2020 will be caused by Non-Communicating Diseases (NCD), such as cardiovascular diseases, diabetes, cancer and chronic respiratory diseases [1]. Thus, the target of this medical tools is to detect timely these diseases such that they can be cured or controlled.

In particular, diabetes detection has been carried out through Retinopathy Image Analysis (RIA). Fundus cameras, scanning lasers, angiography, and, more recently, Optical Coherence Tomography (OCT) are some techniques to acquire retinopathy images. It is important to be aware that these images may be corrupted by noise during the acquisition process. This is due to limitations of sensor capabilities and characteristics, transmission issues, among other facts. As a result, low-quality images, which may not be suitable for medical analysis, are obtained.

There are different types of noise arising during the acquisition process and also varied techniques trying to overcome them. The criteria for deciding which algorithm to use usually depends on the specific features and constraints of the scenario. Ideally, the type and level of noise are known beforehand. However, most of the time, the case is the opposite and estimation and characterization of the noise are required. Under this perspective, there are two questions that should be answered before deciding how to address the situation: (1) Is there any a priori knowledge about the situation? The selection of denoising techniques is fundamental and it can be guided by knowledge of the data to process. Selecting wrongly an algorithm may distort the image instead of improving its quality. (2) What is the goal? There are different approaches for image denoising, but some of them are not suitable for some applications. For instance, Weiner filter is used for removing noise in the sense of Mean Square Error (MSE) but it is not desired for applications in which the visual perception is the target. Having this in mind, there is no algorithm capable of denoising every image and, thus, expectations should be clear before addressing the problem.

In this paper, an evaluation of different state-of-the-art algorithms for noise removal is presented. The paper is organized as follows. Considered algorithms are described and examined in Section II. Synthetic and retinopathy datasets for performing the experimental validation are introduced in Section III. The considered algorithms are tested following the experimental validation and the final results are analysed in Section IV. Finally, some final remarks and future work are presented in Section V.

II. Denoising Algorithms

In the following sections, we discussed the idea behind some traditional and state-of-the-art denoising algorithms as well as their advantages and disadvantages.

A. Mean filter

Given a noisy image g, the restored value on the denoised image $\hat{f}$ at the position (x, y) corresponds to the average of the neighbourhood N (See Eq. 1).

$$\hat{f}(x, y) = \frac{1}{|N(x, y)|} \sum_{(u,v)\in N(x,y)} g(x, y). \tag{1}$$

This process acts on the supposition that the noise is concentrated on the upper part of the frequency spectrum. This approach is able to remove pixels which are not representative in the considered neighbourhood and also reduce the noise by blurring the image. Thus, high frequencies are lost during the process.

This is the simplest denoising method and is not very effective. Denoising through mean filter simply applies a linear transformation to the image (a convolution with a smoothing

kernel), and it makes no attempt to interpret the information in the image and use it in the denoising process.

The main drawbacks of this method are that is sensible to outliers and that it blurs the processed image.

B. Median filter

The median filter is a spatial filter based on order statistics. It replaces the value at each position with the 50th percentile of the neighbourhood (See Eq. 2).

$$\hat{f}(x, y) = \underset{(u,v)\in N(x,y)}{\text{median}}\, g(x, y), \tag{2}$$

Unlike the mean filter, the median filter is able to detect outliers of a neighbourhood and remove them with a smaller impact on the higher frequencies. For the same reason, this filter is well-known for denoising images affected by salt and pepper noise.

C. Filtering by Use of Local Statistics (LS filter)

Digital Image enhancement by using local statistics is a computational technique that involves contrast and noise filtering on two-dimensional arrays based on their local mean and variance. One of the greatest advantages of this type of algorithms is that they are non-recursive and each pixel is processed independently. As a consequence, this approach has a great advantage when is used in real time image processing.

This algorithm was developed to overcome the greatest problem of early techniques in image processing, the computation of the image transformation. Usually, the Fourier or Walsh transform does not represent a big setback for the one-dimensional data array. However, it was very time consuming for a two-dimensional array. As a result, these early techniques were proved not to be suitable for real-time image processing applications.

The assumption of the algorithm based on local statistics is that the sample mean and variance of a pixel is equal to the local mean and variance of all the pixels within a fixed range. For example, in the additive noise filtering case, the variance is calculated as the difference variance of the noise in the corrupted image and the noise itself, the same method is used for multiplicative noise. This simple approach has been pointed as to lack mathematical elegance and sophistication, compared to other techniques. However, the results indicate it is a very effective tool for contrast stretching and noise filtering of images.

Let x_{ij} be the brightness of the pixel (i, j) in a two - dimensional $N \times N$ image. The local mean and variance are then calculated over a $(2n + 1) \times (2m + 1)$ window. The local mean is defined as:

$$\mu_{ij} = \frac{1}{(2n + 1)(2m + 1)} \sum_{k=i-n}^{n+i} \sum_{l=j-m}^{m+j} x_{kl}, \tag{3}$$

and the local variance is:

$$v_j = \frac{1}{(2n + 1)(2m + 1)} \sum_{k=i-n}^{i+m} \sum_{l=j-m}^{j+m} (x_{kl} - \mu_{ij})^2. \tag{4}$$

From these equations, it is not hard to extend the algorithm to deal with images corrupted by additive or multiplicative noise or even both. A noise corrupted image is described as:

$$z_{ij} = x_{ij} * u_{ij} + w_{ij}. \tag{5}$$

Where the mean and variance are calculated as:

$$E[(u_{ij} - \vec{u_{ij}})(u_{kl} - \vec{u_{kl}})] = \sigma^2 * \delta_{ik} * \delta_{jl}. \tag{6}$$

From the structure of the algorithm, it is easy to see that the principal computational load relays on the calculation of the local mean and variance of the image. To make the calculations faster, an improvement to the algorithm is proposed where the image is partitioned in square sub-regions over which the local variance and mean are calculated. Further, the local mean and variance of a pixel are approximated by the use of two-dimensional interpolation formulas. This improvement seems to be promising and perfectly suitable for real time -parallel image processing.

D. Hard and soft thresholding in wavelet domain (wavelet filter)

The Wavelet transform is a signal processing technique for cases when frequency varies over time. For certain classes of signals and images, wavelet analysis provides more precise information about signal data than other signal analysis techniques.

The wavelet transform is used extensively in signal de-noising. The usual way to de-noise signals in wavelet domain is to first transform the signal into wavelet domain, apply hard or soft thresholding and then transform back.

Hard thresholding is a noise suppression method, that applies the following transformation to the empirical wavelet coefficients:

$$F(x) = x \cdot I(|x| > t), \tag{7}$$

where t is a threshold value. For de-noising to perform adequately, t must be chosen carefully.

The theoretically optimal value for t is $t = \sqrt{2\sigma^2 \log(n)/n}$, where $\sigma^2$is the variance of the noise and n is the length of input data. In practice, usually, a smaller value is usually used [2].

Soft thresholding, just like hard thresholding, incorporates a transformation of the empirical wavelet coefficients. The only difference is the chosen nonlinear transformation:

$$S(x) = \text{sign}(x)(|x| - t) \cdot I(|x| > t), \tag{8}$$

where, again, t is the threshold value.

However, when the signal contains discontinuities, the de-noising will also result in artifacts: pseudo-Gibbs phenomena, when the signal is alternatingly overshooting or undershooting

its level. These artifacts depend on the precise alignment between the signal and the basis elements, therefore, depend both on wavelets and the input data.

A solution was proposed in "Translation-Invariant De-Noising" [3], where Coifman and Donoho present an algorithm to minimize the effects of this phenomenon. They propose to shift the signal prior to thresholding, then shift the signal back. This can be done for both time and frequency shifting. For the case of a time-shift, let S_h denote the circulant shift by h: $(S_h x)_t = x_{(t+h)}$,where $x_t \in 0 \leq t \leq n$ is the given signal. A frequency shift by ξ can be represented as follows: $(M_\xi x)_t = e^{i\xi t} x_t$, where M_ξ is the modulation. To incorporate the shifting into the denoising scheme, the signal is first shifted, de-noised then unshifted. A better way is to average the result of this process using a range of shifts.

Because there is no single 'right' choice for the shift parameters h and ξ that will yield the best result for all signals, the authors use a technique called cycle-spinning, where the wavelet de-noising is averaged over all n circulant shifts, without restraining the value to a range. This technique is translation invariant and requires only n log(n) time.

E. K-SVD

This method of de-noising is based on sparse and redundant representations over trained dictionaries. In [4] the authors propose two possible implementations: with a dictionary pre-trained with high quality images, or with training a new dictionary using the corrupted. The algorithm minimizes the number of components $||\alpha||_0$, while the error $||D\alpha - X||_2$ is bounded, where D is the used dictionary, X is the input image image and α is a vector that represents the columns of dictionary D needed to reconstruct the image X. X is processed in small patches of $\sqrt{n}$ x $\sqrt{n}$. The used dictionary must be based on the Sparseland model: The Sparseland model can be represented with the triplet (ε, L, D), where ε is the upper limit for the sum of the square differences between the input image and its denoised version, D is the used overcomplete dictionary, and L is the maximum number of non-zero elements in α. It is required that the corrupted image in this algorithm belong to the Sparseland model. That means that all $\sqrt{n}$ x $\sqrt{n}$ patches in the image can be represented through columns of the matrix D.

The Sparseland dictionary D can be trained using image patches (Z) of good quality, each of size $\sqrt{n}$ x $\sqrt{n}$. D can be found through minimizing the following expression:

$$\varepsilon(D, \{a_j\}_{j=1}^M) = \sum_{j=1}^{M} [\mu_j ||\alpha||_0 + ||D\alpha_j - z_j||_2^2], \tag{9}$$

where μ is chosen implicitly.

The next step, then, is to use D to compute a set of near-optimal projections α. The proposed algorithm to calculate α is the Orthonormal Matching Pursuit (OMP) algorithm, which is an approximation algorithm [4]. The expression for α is:

$$\hat{\alpha}_{ij} = \arg\min_{\alpha} \mu_{ij} ||\alpha||_0 + ||D\alpha - x_{ij}||_2^2. \tag{10}$$

Alternately, patches from the corrupted image can be used for iteratively training the dictionary D, as K-SVD dictionary learning process has a noise rejection capability.

In this case, at first, D is assumed to be known. We can begin the process using a pre-trained dictionary or using any overcomplete dictionary so that a sparse representation α is possible, such as an overcomplete DCT dictionary. The problem is then defined as

$$\begin{aligned}\{\hat{D}, \hat{\alpha}_{ij}, \hat{X}\} = &\arg \min_{\hat{D}, \hat{\alpha}_{ij}, \hat{X}} \lambda ||X - Y||_2^2 \\ &+ \sum_{ij} \mu_j ||\alpha||_0 + \sum_{ij} [||D\alpha_{ij} - R_{ij}X||_2^2],\end{aligned} \tag{11}$$

where R is a matrix that extracts all the (i,j) patches from the image in the Sparseland format and Y is the corrupted image.

First, D and X are assumed fixed, which permits to compute $\hat{\alpha}_{ij}$. Then, given $\hat{\alpha}_{ij}$, the dictionary D can be recomputed using K-SVD. After that, using both D and α, an output image X can be computed:

$$\hat{X} = \arg\min_{x} \lambda ||X - Y||_2^2 + \sum_{ij} [||D\hat{\alpha}_{ij} - R_{ij}X||_2^2]. \tag{12}$$

But because the new X will have a different level of noise, and that value is used in preceding steps, a few more iterations of this process are performed.

Summarizing, this denoising method is based on local operations and involves a sparse decomposition of the image blocks, using a dictionary. The dictionary is trained on patches of a high-quality image or on patches of the noisy image itself.

III. Experimental Validation

In order to do a qualitative and quantitative analysis of the algorithms presented in Section II, a set of trials based on synthetic images was set up. The carried evaluation is twofold: (1) the algorithms were evaluated using synthetic images which were degenerated with a known type of noise and specific parameters for them; (2) we tested the different denoising techniques on retinopathy images. The complexity of denoising these images is that the kind of noise and its parameter are not known. Therefore, some noisy parameters have to be estimated.

A. Synthetic images

When talking about synthetic images, one makes reference to those in which the level of noise is low. The interest of using this kind of images for testing is that they have some characteristics related, for instance, to the content of high frequencies or low frequencies. Degenerated versions of these images can be used to evaluate the performance of denoising algorithms since the originals are known. To make the synthetic images, the three images shown in Fig. 1 were noised in purpose. Cameraman image is characterised by having low frequencies, Lena image has intermediate frequencies and baboon image is

composed by high frequencies. Therefore, using this images as a base of the synthetic images will give a proper analysis expanding the vast majority of the frequency range.

Fig. 1. Original images. From left to right: Lena, cameraman and baboon.

Each one of these images was corrupted with 5 different types of noise: Gaussian, Rician, uniform, salt and pepper and speckle noise. While the first four types of noise are additive, the last one is multiplicative. Specifically, the characteristic parameters of each kind of noise are represented in Table I.

TABLE I
CHARACTERISTICS OF THE DIFFERENT TYPES OF NOISE.

Type of noise	**Characteristics**
Gaussian	μ=0 , σ=0.1
Rician	μ=0.05 , σ=0.1
Uniform	μ=0 , σ=0.1
Salt and Pepper	5% salt , 5% pepper
Speckle	μ=0 , σ=0.04

After applying the described noises in the original images seen in Fig. 1, the obtained noisy images are the ones shown in Fig. 2.

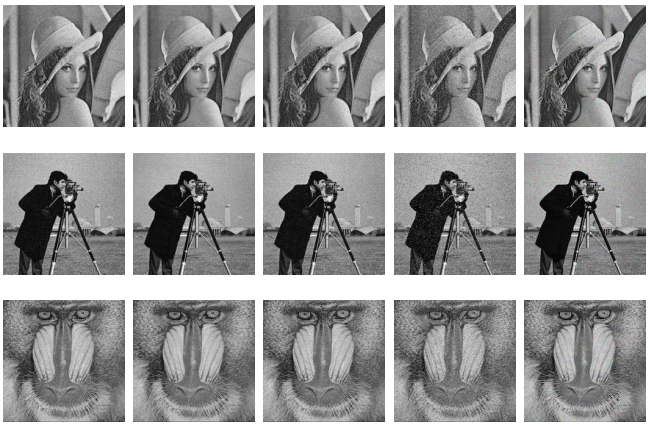

Fig. 2. Noised images. From left to right: additive Gaussian, additive Rician, additive uniform, additive salt and pepper and multiplicative speckle noise, respectively.

For visualization purposes, all the images shown in Section IV will be shown in a colormap format. As can be seen in Fig. 3, the noise added to the images in Fig. 2 can be better with this color visualization.

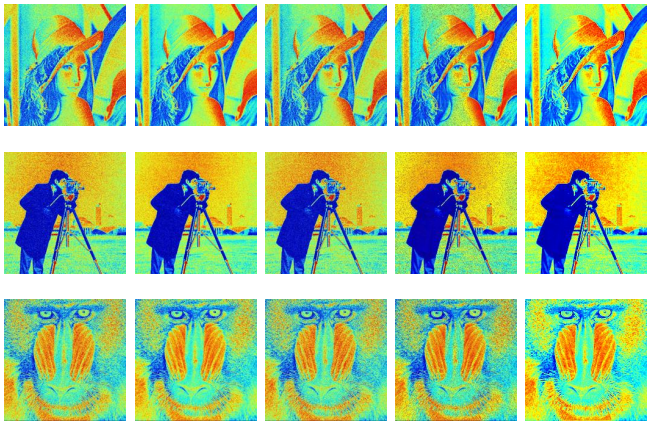

Fig. 3. Noised images represented in a colormap format. From left to right: additive Gaussian, additive Rician, additive uniform, additive salt and pepper and multiplicative speckle noise, respectively.

B. Retinopathy images

In the previous section, we presented an evaluation based on noising synthetic images in which the noise characteristics were known in advance. However, these parameters are usually not given in real-life scenarios. To evaluate the algorithm under this kind of conditions, we consider Spatial-Domain Optical coherence tomography (SD-OCT) images. The evaluation using these images consists of the following steps. First, a volume of OCT is taken and processed using the different considered algorithms. For this task, we consider the dataset provided by the University of Girona [5]. Second, the noise is estimated and the determined value is used as input for the K-SVD algorithm.

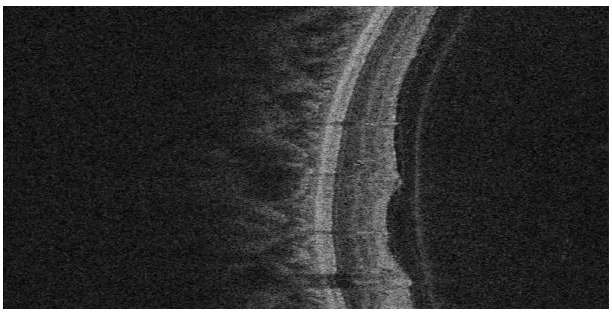

Fig. 4. Retinopathy image.

The goal of denoising retinopathy images is to obtain a better segmentation of the different retinopathy layers. For this, the resulting denoised images are segmented using the tool called OCT Explorer developed by the University of Iowa [6]. Finally, the outcome of the segmentation is compared against the segmentation of the noisy retinopathy volume, which can be seen in Fig. 5.

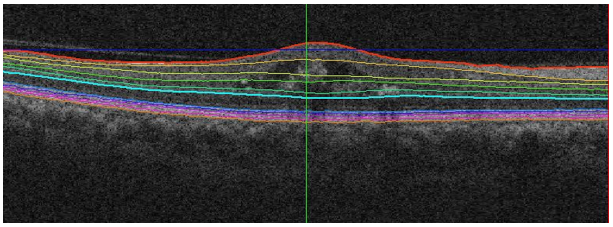

Fig. 5. Segmentation of a noisy retinopathy volume.

The output segmentation of the noisy volume shows up all the 11 existing layers of the retina. Since there is not any ground-truth to check the accuracy of the resulting segmentations, only the number of obtained layers and a visual analysis can be given in the following analysis.

IV. Results

The algorithms were evaluated under the conditions presented in Section III. The results of the evaluation for each algorithm are presented in the following sections.

A. Parameter influence

K-SVD algorithm, as presented in Section II-E, requires to tune up some parameters. Each variable was varied during the evaluation process to observe its impact on the results. In general, the observed impact was the following:

1) Block size: The block size represents the dimension of the patch to consider. On one hand, when evaluating synthetic images, the higher the value of the parameter, the more blurred the result. On the other hand, the best results for SD-OCT was obtained when using a block size of 32 – the biggest value tested for this parameter.

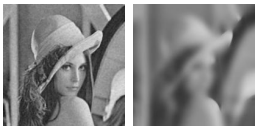

Fig. 6. Denoised images with K-SVD algorithm using different block sizes. On the left the block size is 4 while on the right is 32.

2) Dictionary size: This parameter determines the number of patches to consider for the dictionary. The size by itself does not modify the result dramatically.

Fig. 7. Denoised images with K-SVD algorithm using different dictionary sizes. On the left the dictionary size is 256 while on the right is 512.

3) Training iterations (Trainnum): This value corresponds to the number of iterations performed during the training step. Usually, when it increases, the result is better. However, we observed that there is a limit for which no better result is obtained. It is required that its value must be greater or equal to the dictionary size.

4) Gain: The influence of this parameter is seen on the edges of the resulting images. When its value is decreased, the image is less denoised but the edges were better preserved.

5) Maximum number of atoms (maxatoms): This parameter is related to the number of elements on the dictionary to consider when reconstructing a patch of the noisy image. The higher the value, the better the reconstruction but also the slower the algorithm gets. Thus, the idea is to find the minimum value such that the trade-off between reconstruction and computation time is balanced.

6) Noise estimation: As stated previously, the level of distortion on the SD-OCT images is not known beforehand and this value is required by the K-SVD algorithm. Then, the estimation of this parameter is carried out. The method consists in taking a homogeneous patch of the image and, by looking at the histogram, determine the sigma value. The observed value for sigma among the test set was around 10. Moreover, we calculated this parameter for different retinopathy volumes getting similar results.

B. Synthetic images

In this section, the obtained results of denoising the synthetic images with the methods explained in Section II are shown.

First, the mean, median, local statistics and wavelet filters are evaluated with the synthetic images of Section III-A. After that, the K-SVD filter is applied in such images. With these last results, proper comparisons are done between the K-SVD and the traditional denoising algorithms.

1) Mean filter: Fig. 8 shows the results of the proposed images after the mean filter is applied. As observed from the colormap set, the filter has a better performance on Rician noise compared to the rest of images.

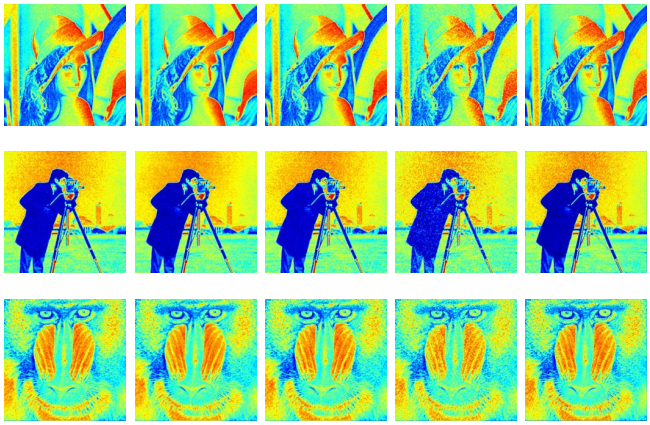

Fig. 8. Denoised images with the mean filter represented in a colour map format. From left to right: additive Gaussian, additive Rician, additive uniform, additive salt and pepper and multiplicative speckle noise, respectively.

2) Median filter: Fig. 9 shows the results of the proposed images after the median filter is applied. As observed from the colour map set, the filter has a better performance on salt and pepper compared to the rest of images.

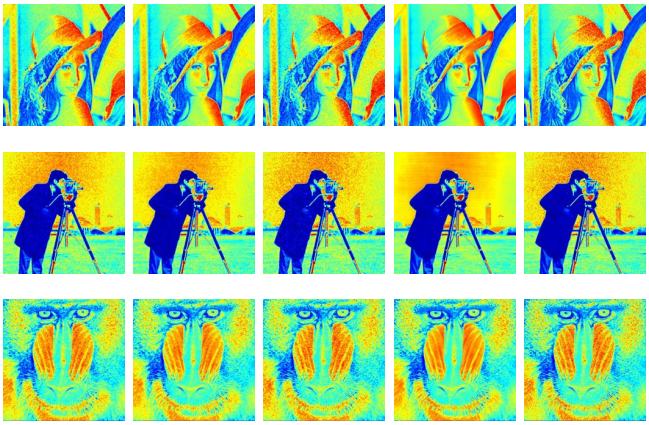

Fig. 9. Denoised images with the median filter represented in a colormap format. From left to right: additive Gaussian, additive Rician, additive uniform, additive salt and pepper and multiplicative speckle noise, respectively.

3) LS filter: Fig. 10 shows the results of the proposed images after local statistics filter is applied. As observed from the colormap set, the filter has a better performance on denoising high-frequency images with Gaussian, Rician or uniform noise.

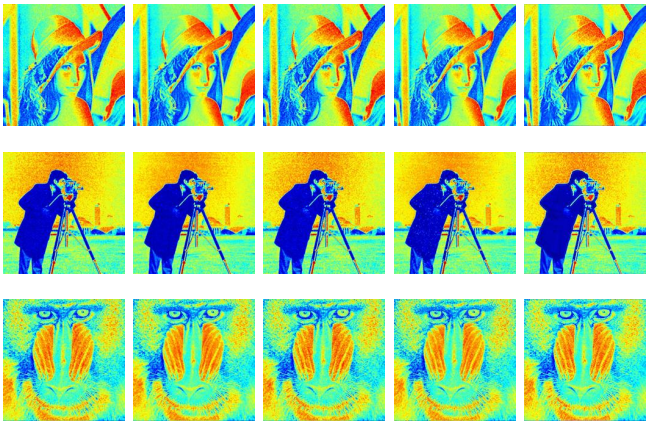

Fig. 10. Denoised images with the LS filter represented in a colormap format. From left to right: additive Gaussian, additive Rician, additive uniform, additive salt and pepper and multiplicative speckle noise, respectively.

4) Wavelet filter: Fig. 11 shows the results of the proposed images after the median filter is applied. As observed from the results represented in Section V, the Wavelet filter does not perform better than other methods.

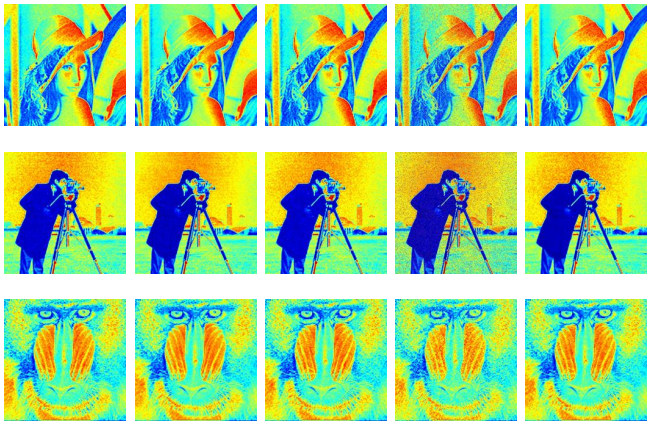

Fig. 11. Denoised images with the wavelet filter represented in a colormap format. From left to right: additive Gaussian, additive Rician, additive uniform, additive salt and pepper and multiplicative speckle noise, respectively.

5) K-SVD filter: Fig. 12 shows the results of the proposed images after K-SVD filter is applied. As observed from the colormap set and the results represented in Section V, the filter has a better performance on Gaussian, uniform and speckle noise. However, its performance is not remarkable when denoising high-frequency images.

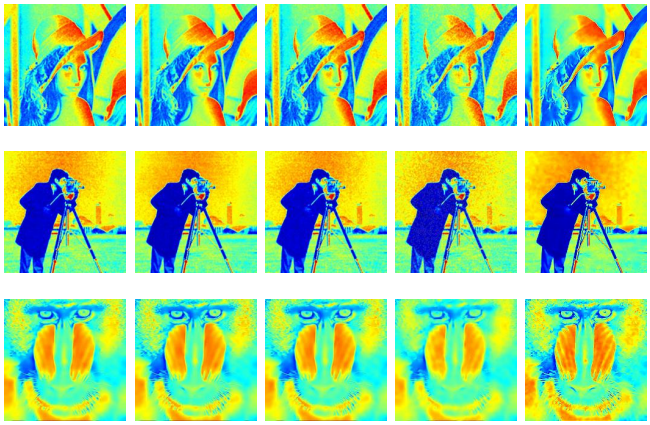

Fig. 12. Denoised images with the K-SVD filter. From left to right: removing Gaussian, Rician, uniform, salt and pepper and speckle noise, respectively.

The comparative results are shown in Fig. 13. It can be seen that the PSNR of the initial noise is improved in most of the cases except for Baboon test set in which the results are the same or worse. In general, Baboon is the test set leading to the worst restoration for all the evaluated algorithms. This fact could be a consequence of the characteristics of the spiky hair of the baboon. These spikes form a set of edges which can be "easily" affected by adding noise and also by restoring with the considered algorithms. Thus, when the denoising process is carried out on this section, most of it, which is formed by high frequencies, is lost.

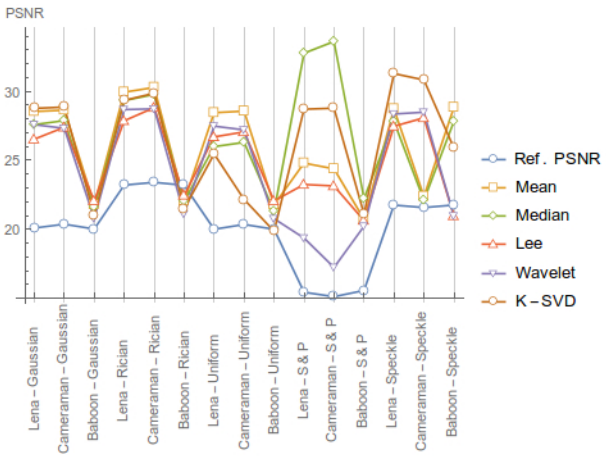

Fig. 13. Comparison of the K-SVD filter with traditional filtering techniques.

The K-SVD algorithm is able to remove noise from the different test images. However, its performance was not always the best. It can be observed that in some cases, the values of PSNR for mean filter and median filter are higher than the ones for K-SVD. Contrarily, we see that the wavelet filter does not succeed in filtering any of the types of noise, compared to other methods.

As expected, median filter outperforms when evaluated under salt and pepper noise in comparison with the other algorithms.

C. Retinopathy images

To provide a quantitative comparison of the denoised retinopathy images with the different methods, we computed the PSNR. In this case, it was performed between the input image and the output of each algorithm. The obtained results was of 26.68 dB. It is clear that although this computation does not give the real PSNR, the obtained value is useful to get an idea of the performance of the different algorithms. In general terms, K-SVD is able to reduce noise on the image without affecting the key components (the layers on the images) more than the other algorithms.

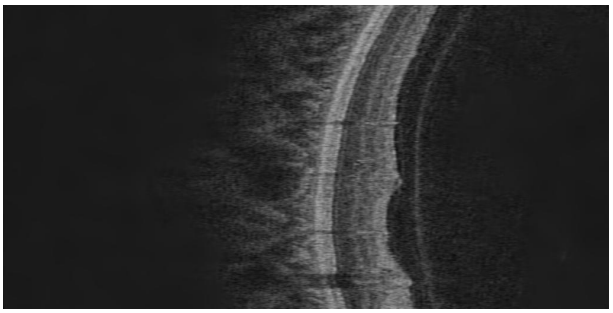

Fig. 14. Denoised retinopathy image with the K-SVD filter.

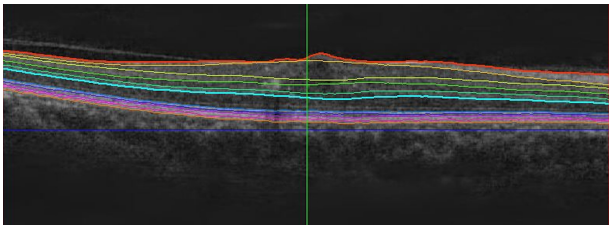

Fig. 15. Segmentation of a retinopathy volume after denoising using the K-SVD filter.

V. Final remarks

The K-SVD algorithm was compared against different well-known techniques for denoising images such as mean, median, lee and wavelet-based filter. The algorithm was analysed in terms of the influence of its parameters on the results and, also, on its capability to remove noise.

The evaluation showed that the K-SVD algorithm was able to denoise synthetic and SD-OCT images achieving better results in some cases compared to the other algorithms. Although noise was reduced, the segmentation carried out on an OCT volume evidenced that no significant improvement was observed between noisy and denoised images.

The K-SVD algorithm is recommended for removing noise on SD-OCT images since, in terms of PSNR, the results were the best. However, further evaluation on the segmentation section should be done in order to determine the best parameters for this task.

VI. Appendices

This appendix includes all the original grayscale denoised images and its PSNR values.

Fig. 16. Denoised images with the mean filter. From left to right: removing Gaussian, Rician, uniform, salt and pepper and speckle noise, respectively.

Fig. 17. Denoised images with the median filter. From left to right: removing Gaussian, Rician, uniform, salt and pepper and speckle noise, respectively.

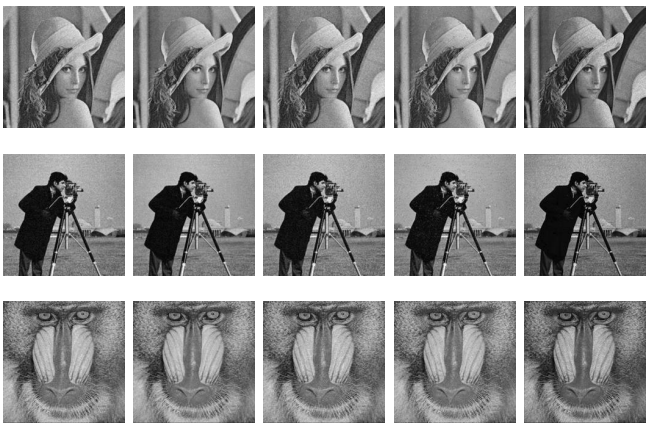

Fig. 18. Denoised images with the LS filter. From left to right: removing Gaussian, Rician, uniform, salt and pepper and speckle noise, respectively.

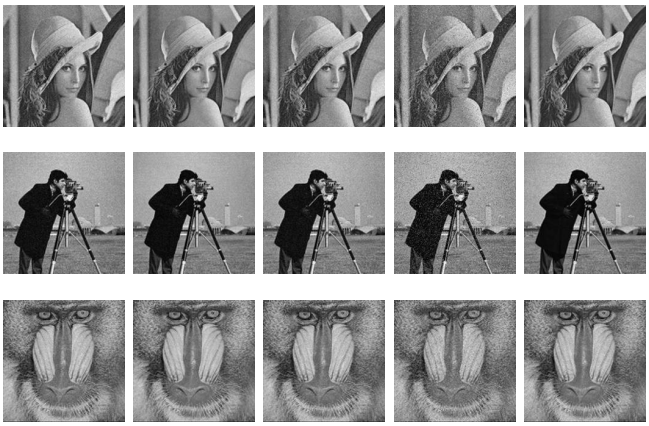

Fig. 19. Denoised images with the wavelet filter. From left to right: removing Gaussian, Rician, uniform, salt and pepper and speckle noise, respectively.

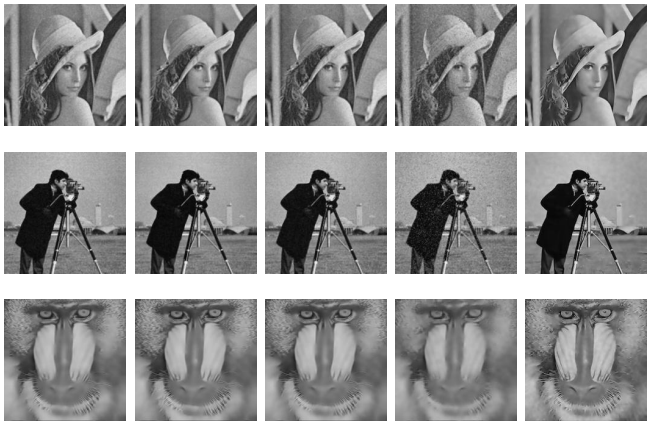

Fig. 20. Denoised images with the K-SVD filter. From left to right: removing Gaussian, Rician, uniform, salt and pepper and speckle noise, respectively.

In order to easily compare the performance of the different methods, the PSNR results have been grouped in different tables according to the different kind of noise that have been analyzed.

TABLE II
PSNR FOR DENOISING ALGORITHMS CONSIDERING GAUSSIAN NOISE

Technique	**Lena**	**Cameraman**	**Baboon**
Mean	28.54	28.64	21.77
Median	27.59	27.88	21.60
LS	26.53	27.40	**22.06**
Wavelet	19.91	20.18	17.76
K-SVD	**28.77**	**28.86**	21.02
Input noise	**20.79**	**20.37**	**20.02**

TABLE III
PSNR FOR DENOISING ALGORITHMS CONSIDERING RICIAN NOISE

Technique	**Lena**	**Cameraman**	**Baboon**
Mean	**29.94**	**30.29**	22.05
Median	29.33	29.78	22.04
LS	27.87	28.82	**22.46**
Wavelet	22.61	22.82	19.27
K-SVD	29.35	29.87	21.50
Input noise	**23.20**	**23.40**	**23.19**

TABLE IV
PSNR FOR DENOISING ALGORITHMS CONSIDERING UNIFORM NOISE

Technique	**Lena**	**Cameraman**	**Baboon**
Mean	28.45	28.57	21.77
Median	25.99	26.30	21.31
LS	26.70	27.06	**22.06**
Wavelet	19.84	20.16	17.72
K-SVD	**28.72**	**28.78**	21.10
Input noise	**20.00**	**20.35**	**20.00**

TABLE V
PSNR FOR DENOISING ALGORITHMS CONSIDERING SALT AND PEPPER NOISE

Technique	Lena	Cameraman	Baboon
Mean	24.80	24.38	20.72
Median	**32.77**	**33.62**	**22.22**
LS	23.25	23.14	20.64
Wavelet	21.18	25.22	15.31
K-SVD	25.47	22.16	19.86
Input noise	**15.43**	**15.11**	**15.56**

TABLE VI
PSNR FOR DENOISING ALGORITHMS CONSIDERING SPECKLE NOISE

Technique	Lena	Cameraman	Baboon
Mean	28.73	22.38	**28.84**
Median	27.82	22.11	27.82
LS	27.47	28.08	20.97
Wavelet	28.36	28.49	20.97
K-SVD	**31.29**	**30.83**	25.90
Input noise	**21.75**	**21.57**	**21.76**

REFERENCES

[1] World Health Organization. Chapter 1 burden: mortality, morbidity and risk factors. http://www.who.int/nmh/publications/ncd_report_chapter1.pdf?ua=1, 2010. [Online; accessed 2015-29-12].

[2] Rafal Weron Pavel Czek, Wolfgang Hrdle. Statistical tools for finance and insurance. http://sfb649.wiwi.hu-berlin.de/fedc_homepage/xplore/tutorials/xlghtmlnode93.html, 2005.

[3] R.R. Coifman and D.L. Donoho. Translation-invariant de-noising. *In Wavelets and Statistics, Lecture Notes in Statistics. New York*, pages 125–150, 1995.

[4] M. Elad and M. Aharon. Image denoising via sparse and redundant representation over learned dictionaries. *IEEE Transactions on Image Processing, Vol. 15*, 2006.

[5] Joan Massich Guillaume Lemaitre. Initiative for collaborative computer vision benchmarking. http://visor.udg.edu/dataset/.

[6] Milan Sonka Kimberly A Glynn. Iowa reference algorithms: Human and murine oct retinal layer analysis and display. http://www.iibi.uiowa.edu/, 2007.