

An Extensive Comparison of Image Denoising Algorithms

Introduction to Image Processing Class

Department of Electrical Engineering, University of Burgundy
Le Creusot, France

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 this 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 recent algorithms 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, subspace filter, BMxD, K-SVD, NLM, PGPD, 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), deaths of about 15% of the population 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 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 in subsequent steps, 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 characterisation of the noise are required in advance. Under this perspective, there 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 “easier” when there is a priori knowledge of the data to process. Selecting wrongly an algorithm may distort the image instead of improving its quality. (2) Which 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 and compared 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).

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

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 neighborhood 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 is that is sensible to outliers and that it blurs the processed image.

B. Median filter

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

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

Unlike mean filter, median filter is able to detect outliers of a neighborhood 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 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:

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

and the local variance is:

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

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 noisy corrupted image is described as:

$$z_{ij} = x_{ij} * u_{ij} + w_{ij}. \quad (5)$$

Where the mean and variance are calculated as:

$$E[(u_{ij} - \bar{u}_{ij})(u_{kl} - \bar{u}_{kl})] = \sigma^2 * \delta_{ik} * \delta_{jl}. \quad (6)$$

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)

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:

$$F(x) = x \cdot I(|x| > t), \quad (7)$$

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 σ^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 non-linear transformation:

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

where, again, t is the threshold value.

However, when the signal contains discontinuities, the denoising 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, 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. Subspace

Synthetic aperture radar (SAR) imaging technique is a popular technique for remote sensing and monitoring applications because of its usability under various weather conditions and its ability to provide high-resolution imagery. A SAR image is generated by sending electromagnetic waves from a moving platform, space borne or airborne, toward the target surface and by coherently processing the returned backscattered signals from multiple distributed targets. However, the coherent processing causes speckle effect and gives SAR images its noisy appearance. Speckle presence appears as granular noise which reduces the image resolution and

may hamper the operation of image interpretation and analysis.

1) Adaptive spatial-domain filters: The assumptions made in implementing these filters are as follows:

- The SAR speckle is modeled as a multiplicative noise
- The noise and signal are statistically independent
- The sample mean and variance of a pixel are equal to its local mean and local variance computed within a window centered on the pixel of interest

The Lee and Kuan filters have similar formation but differ in signal model assumptions and derivation. The filters achieve a balance between averaging in homogeneous regions and a strict all-pass (identity) filter in edge contained regions.

The Frost filter attempts to strike a balance between averaging and identity filter by forming an exponential-shaped filter kernel that can adaptively vary from an average filter to an identity filter. At low coefficient variation, the filter is more average-like and, at high coefficient variation, the filter attempts to preserve sharp features by retaining its original pixel value.

The enhanced Lee and Frost filter uses three variations of coefficient values, namely, low, intermediate, and high, to divide an image into homogeneous regions, heterogeneous regions, and isolated point target regions, respectively. The filter outputs a local mean at homogeneous regions and retains the original pixel at points of high activity.

Disadvantages:

- Fail to maintain the mean value, particularly if the number of look of the original SAR data is small.
- The highly reflective point targets are blurred.
- The highly reflective point targets are blurred.

2) Wavelet transform: An outcome of wavelet theory, denoising in the discrete wavelet transform (DWT) domain may be stated as a thresholding of DWT coefficients of the noisy image. The log-transformed noisy image is either adaptively thresholded or empirically shrunk in an adaptive fashion has been utilized. The major drawbacks of such approach are the backscatter mean preservation in ho-

mogeneous areas, sharpness preservation, and ringing impairments. To overcome these deficiencies, Argenti proposed a minimum-mean-square error filtering performed in the decimated wavelet domain by means of an adaptive rescaling of the detail coefficients and the local space-varying signal, where the noise statistics are estimated in the wavelet domain. Using statistical modeling of wavelet coefficients, Ranjani et al. proposed a speckle suppression technique using dual-tree wavelet transform by putting into consideration the significant dependences of the wavelet coefficients across different scales. The interscale dependence of the wavelet coefficients in each sub-band is modeled using bivariate Cauchy probability density function (pdf).

3) Subspace-based technique: To decompose the vector space of the noisy image into a signal-plus-noise subspace and the noise subspace. The noise removal is achieved by nulling the noise subspace and controlling the noise distribution in the signal subspace. For white noise, the subspace decomposition can theoretically be performed by applying the Karhunen-Loeve transform (KLT) to the noisy image. Linear estimator of the clean image is performed by minimizing image distortion while maintaining the residual noise energy below some given threshold.

For colored noise:

- 1) A prewhitening approach prior to KLT transform.
- 2) A generalized subspace for simultaneous diagonalization of the clean and noise covariance matrices.

Problem: The fundamental signal and noise model for subspace methods is additive noise uncorrelated with the signal. In SAR images, the noise is multiplicative in nature.

Solution:

- A homomorphic framework takes advantage of logarithmic transformation in order to convert multiplicative noise into additive noise.
- But this nonlinear operation totally changes the statistics of SAR images and induces bias in their mean values. For the purpose of radiometric preservation, the biased mean needs to be corrected, along with antilog operation.

4) Statistical modeling of speckle noise in SAR images: With homogeneous targets and weakly

TABLE I
MEAN AND STANDARD DEVIATION OF SPECKLE NOISE

L	\bar{N}_l	\bar{v}_{N_l}	\bar{N}_l (dB)	\bar{v}_{N_l} (dB)
1	\bar{N}_l	\bar{N}_l	$10 \log \bar{N}_l - 2.507$	5, 570
2	\bar{N}_l	$\bar{N}_l \sqrt{2}$	$10 \log \bar{N}_l - 1.174$	3, 488
4	\bar{N}_l	$\bar{N}_l \sqrt{2}$	$10 \log \bar{N}_l - 0.556$	2, 314
10	\bar{N}_l	$\bar{N}_l \sqrt{2}$	$10 \log \bar{N}_l - 0.221$	1, 408
20	\bar{N}_l	$\bar{N}_l \sqrt{2}$	$10 \log \bar{N}_l - 0.109$	0, 983

textured areas in SAR images, the speckle noise is fully developed, and the multiplicative model is used to describe it. Since most available image denoising techniques were developed for additive white Gaussian noise (AWGN), it is necessary in case of a fully developed speckle noise to apply a logarithmic transform to the multiplicative model in order to convert it into additive. The process involves applying logarithmic transform prior to the denoising technique and then exponentially transforming the output to obtain the despeckled image. As a nonlinear operation, the logarithmic transform totally changes the statistics of SAR images, so the original speckle statistics cannot directly be used with the log-transformed images. In this section, the pdf, mean, and standard deviation values of the log-transformed speckle noise are briefly discussed. The purpose is to correct the biased mean for radiometric preservation. SAR images are usually available in two formats: intensity and amplitude:

Intensity Format:

$$G = W \times N \quad (9)$$

G denotes the SAR image intensity, W is the backscattering coefficients. Applying the logarithmic function to both sides of 9, we get

$$\log(G) = \log(W) + \log(N) \quad (10)$$

Finally, a table of mean and standard deviation of speckle noise in linear and logarithmic scales is constructed as follow

Amplitude Format: If Eq. 9 is in amplitude format and L = 1, then the pdf of N obeys the Rayleigh pdf. For L > 1 amplitude image, different techniques are used to obtain a closed analytical form for the pdf. Among these techniques, there are histogram estimation technique and approximation method using Edgeworth expansion.

5) *Subspace-based spatial domain constraint approach (SDC):* The noise here is assumed to be AWGN, uncorrected with the signal, after using a weighting scheme. Eventually, the implementation of Signal Subspace Approach for Uncorrelated Speckle Noise is as follow:

- 1) Apply the homomorphic transformation to the noisy image

$$Y_1 = \log(G) \quad (11)$$

- 2) Estimate the noise variance v^2
- 3) Compute the dimension of signal subspace r.
- 4) Using the estimated r in step 3, apply eigen decomposition on RY₁; then, extract the basis vectors of signal subspace U₁ and their related eigenvalues

$$\Delta_X^i = \Delta_{X_1}^i - v_n^2 \quad (12)$$

- 5) After estimating μ , the optimum linear estimator is computed.

$$H_{SDC} = U_1 \Delta_{X_1} (\Delta_{X_1} + \mu v_n^2 I)^{-1} U_1^T \quad (13)$$

- 6) Compute the clean image

$$\hat{X}_1 = H_{SDC} Y_{1,1} \quad (14)$$

- 7) Reverse the homomorphic effect by taking the exponential of X₁ as follows:

$$\hat{X}_1 = 10^{\hat{X}_1,1} \quad (15)$$

- 8) Apply bias adjustment according to Table I.

F. BMxD

BM3D is an image denoising strategy which uses block matching and collaborative filtering in 3D domain.

The aim of bloc matching is to stack similar 2D image fragments ("blocks"), in 3D arrays which are called "groups". Similarities are computed between candidate fragments at different spatial locations and each reference fragment. The groups are disjoints, so some blocks can be stacked in multiple groups. These groups have a "diameter" which is the maximum number of blocks inside.

After getting the groups by block matching, a collaborative filtering is applied, which includes : 3D transformation, shrinkage of the transform spectrum, and inverse 3D transform. Different 3D

transformations can be applied according to the type of noise, or it can be decomposed in 2D transform followed by 1D transform.

The last process used is the aggregation. Because the groups are disjoint, multiple estimates are given for some blocks so they will overlap. In order to aggregate them, the blocks are awarded with weights. For each final pixel, the block pixels corresponding are averaged with their given weights.

The algorithm is divided in two steps. The first one consists in applying block matching and collaborative filtering on the noisy image. The shrinkage of the coefficients of the 3D group is realised for this first step with an Hard-Thresholding filter. The basic estimate got from the first step helps to improves an other block matching in the second step. Collaborative filtering is applied once again on these new obtained blocks and the shrinkage is now realised by a Wiener filter, which attenuates the frequencies using the signal to noise ratio. The final estimate is therefore obtained, what corresponds to the denoised image.

Multiple parameters can be tuned in order to choose between a faster or more accurate algorithm, like : block size, group diameter, step between reference blocks and search area.

This method improves the non-local mean filter method in using 3D transform instead of 1D. Thanks to the filtering in transform domain applied on the already process groups, the method shows good preservations of uniform areas, smooth intensity transitions, textures, repeating patterns and sharp edges. The main advantages of this approach is the non-locality and the collaborative filtering

G. 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 $\sqrt{n} \times \sqrt{n}$. 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 $\sqrt{n} \times \sqrt{n}$ patches in the image can be represented through columns of the matrix D.

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

$$\varepsilon(D, \{\alpha_j\}_{j=1}^M) = \sum_{j=1}^M [\mu_j \|\alpha_j\|_0 + \|D\alpha_j - z_j\|_2^2], \quad (16)$$

where μ is chosen implicitly.

The next step, then, is to use D to compute a set of near-optimal vectors α ,

$$\hat{\alpha}_{ij} = \arg \min_{\alpha} \mu_{ij} \|\alpha\|_0 + \|D\alpha - x_{ij}\|_2^2. \quad (17)$$

This process guarantees a decreasing error after each iteration.

Alternately, patches from the corrupted image can be used for 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 such as an overcomplete DCT dictionary. The problem is then defined as

$$\begin{aligned} \{\hat{D}, \hat{\alpha}_{ij}, \hat{X}\} = & \arg \min_{\hat{D}, \hat{\alpha}_{ij}, \hat{X}} \lambda \|X - Y\|_2^2 \\ & + \sum_{ij} \mu_{ij} \|\alpha_{ij}\|_0 + \sum_{ij} [\|D\alpha_{ij} - R_{ij}X\|_2^2], \end{aligned} \quad (18)$$

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:

$$\hat{X} = \arg \min_X \lambda \|X - Y\|_2^2 + \sum_{ij} [\|D\hat{\alpha}_{ij} - R_{ij}X\|_2^2]. \quad (19)$$

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 noisy or a high-quality image.

H. NLM

Non local means (NL-means) algorithm for image denoising in [5] is based on a non-local averaging of all pixels in the image. The main difference of the NL-means algorithm compared to local filters or frequency domain filters is the systematic use of all possible self-predictions the image can provide, a principle used also in [6]. There are many different methods for denoising image with common technique of averaging. This averaging may be performed locally: the Gaussian smoothing model [7], the anisotropic filtering [8] and the neighborhood filtering [9], by the calculus of variations: the Total Variation minimization [10], or in the frequency domain: the empirical Wiener filters [9] and wavelet thresholding methods [11].

Non local means algorithm is based on the following equation:

$$NL[u](x) = \frac{1}{C(x)} \int_{\Omega} e^{-\frac{(G_a * |u(x+) - u(y+)|)^2(o)}{h^2}} u(y) dy, \quad (20)$$

where $x \in \Omega$,

$$C(x) = \int_{\Omega} e^{-\frac{(G_a * |u(x+) - u(y+)|)^2(o)}{h^2}} u(y) dy \quad (21)$$

is a normalizing constant, h is the filtering parameter and G_a is a Gaussian kernel. The new value of a pixel x is defined as the mean of all the pixels in the image whose neighborhood is similar to the neighborhood of x.

$$NL[v](i) = \sum_{j \in I} w(i, j) \cdot v(i, j), \quad (22)$$

where $v = \{v(i) \mid i \in I\}$ represents all the pixels in an Image and the weights for each pixel depend on the similarity between the two pixels I and j. The weights $w(i, j)$, depend on the similarity between two pixels I and j under the conditions: $0 < w(i, j) < 1$, $\sum_j w(i, j) = 1$.

The similarity between two pixels i,j is determined by the similarity of the intensity gray level vectors $v(N_i)$ & $v(N_j)$ of square neighborhoods of fixed size. This similarity index is defined by the weighted Euclidean distance $\|v(N_i) - v(N_j)\|_{(2,a)}^2$ where $a > 0$ is the Euclidean distance between the two noisy neighborhoods, making the system robust:

$$E \|v(N_i) - v(N_j)\|_{(2,a)}^2 = \|u(N_i) - u(N_j)\|_{2,a}^2 + 2\sigma^2. \quad (23)$$

The weights are defined as

$$w(i, j) = \frac{1}{Z(i)} e^{-\frac{-\|v(N_i) - v(N_j)\|_{2,a}^2}{h^2}}, \quad (24)$$

where $Z(i)$ is the normalizing constant

$$Z(i) = \sum_j e^{-\frac{-\|v(N_i) - v(N_j)\|_{2,a}^2}{h^2}}. \quad (25)$$

The advantage of NLM denoising is that it not only compares the gray level with a single pixel but with the geometrical configuration of the neighborhood.

1) Optimized Bayesian Non Local Means filtering: This method dedicated to ultrasound Images proposed in [12] is a modified version of widely used Non Local Means technique. The NLM algorithm analyses the patterns around the pixels rather than comparing intensity values which may be highly corrupted by noise. For each pixel, the patches from the whole image are compared to find restoration parameters. The modified algorithm proposes a Bayesian formulation of the NLM filter inspired by [13] which optimises the computational cost of the original algorithm. This algorithm can be implemented in two ways, pixelwise and blockwise.

Let us consider an image $u(x_i)$, $x_i \in \Omega^{\text{dim}}$, where Ω^{dim} is the size of the image and N_i is a patch around a pixel x_i defined by a square neighbourhood of size $(2d+1)^{\text{dim}}$, $d \in N$. Pixelwise NLM compares the patch around a pixel i with patch around the all pixels j belongs to a predefined search volume, Δ_i . For each i and j , the L-2 norm (distance) is computed and weighted with a Gaussian kernel as follows:

$$w(x_i, x_j) = \frac{1}{Z_i} e^{-\frac{\|u(N_i) - u(N_j)\|_2^2}{h^2}}, \quad (26)$$

where $u(N_i)$ and $u(N_j)$ are vectors containing all pixels in a patch N_i and N_j respectively, Z_i is the normalization constant and h is filtering parameter to control the decay of exponential function. After weighting each pixel the value for the restoration of a pixel i is calculated using equation below.

$$\text{NL}[u](x_i) = \sum_{x_j \in \Omega^{\text{dim}}} w(x_i, x_j) \cdot u(x_j). \quad (27)$$

Pixel wise NLM involves a great computational burden, hence blockwise NLM is proposed where the image is partitioned into overlapping blocks B_{ik} centred around pixels x_{ik} chosen equally distributed in whole image. The restoration is accomplished using equation

$$\text{NL}[u](B_{ik}) = \sum_{B_j \in \Delta_{ik}} w(B_{ik}, B_j) \cdot u(B_j), \quad (28)$$

where $w(B_{ik}, B_j) = \frac{1}{Z_{ik}} e^{-\frac{\|u(B_{ik}) - u(B_j)\|_2^2}{h^2}}$ and Z_{ik} normalization constant, Δ_{ik} is the neighborhood around pixel x_{ik} and h is filtering parameter to control the decay of exponential function. The Bayesian formulation of blockwise NLM efficiently analyses a particular block B_i by using an empirical estimator as below:

$$V(B_{ik}) = \frac{\frac{1}{\Delta_{ik}} \sum_{j=1}^{\Delta_{ik}} u(B_j) p(u(B_{ik}) | u(B_j))}{\frac{1}{\Delta_{ik}} \sum_{j=1}^{\Delta_{ik}} p(u(B_{ik}) | u(B_j))}, \quad (29)$$

where $p(u(B_{ik}) | u(B_j))$ is the probability density function of probability of each $u(B_{ik})$ conditioned to $u(B_j)$. Using this estimator modelling of noise present in a grayscale image becomes easier hence

and restoration can be made more efficient. Also if the prior knowledge of the noise present in an application is available, the conditional pdf can be compared with pdf of noise present and the process of denoising through NLM can be modified specific to that application.

I. PGPD

As a classical problem in computer vision area, image denoising plays a really important role in many applications in the real life, yet it is still an active topic because it provides an ideal test bed for image modeling techniques. We choose to try patch group based nonlocal self-similarity prior learning, which can be called as PGPD, to deal with denoising problem.

All the images and equations in this part are from [14]. We will try to use refined words to depict the whole method. For the sake of simplicity, we can decompose PGPD into 2 parts: first being the learning phase and second denoising algorithm itself.

First of all, let's explain the notion of patch group (PG). A patch group is formed by grouping a number of the most similar patches near to the target patch. For the sake of efficiency and time complexity, not all the patches in an image are used for grouping. We only search in a large neighborhood around the target patch. In IV-A3, We will discuss what would happen if we turn the size of this search neighborhood (or window).

Intuitively, there exists a large number of nonlocal similar patches in every image. If we are able to find the common information for each PG and learn it in certain models, then it would be easier for us to denoise this PG since we already know the prior knowledge. The common information we mention here is exactly so-called nonlocal self-similarity (NSS) prior. Many recent work obtained impressive results by applying NSS[15][16][17]. However, like BM3D[17], they acquire NSS from noisy input image without considering the NSS of clean images. This is not able to fully utilize NSS. So clean images are used for learning NSS prior in PGPD method. In the learning stage of PGPD, they apply Gaussian Mixture Model (GMM) for learning prior.

Until now, learning NSS prior is still an open problem because too many models can be used for

this task. In our opinion, the reason that using GMM gives a great performance, is due to the nature property of a clean image. As mentioned in [18], the gradient of a clean image obeys heavy-tailed distribution[19], which actually can be modeled by mixture Gaussian distributions. So GMM is a suitable and reasonable choice for denoising.

Because of the fact that the authors of this paper only give the learning result, we cannot discuss much in this part and have no idea what parameters can be tuned. But based on the paper, we are able to describe the PG-GMM algorithm. In order to acquire the mean and variance of GMM, we first model a objective function:

$$F = \sum_{n=1}^N \ln \left(\sum_{k=1}^K \pi \prod_{m=1}^M \mathcal{N}(\hat{x}_{n,m} | \mu_k, \Sigma_k) \right). \quad (30)$$

Once acquiring this function, EM algorithm[20] is applied to find the optimization values. The final result of learning part is to provide a set of covariance matrix Σ for different GMM components, which provide dictionaries as well as regularization parameters for denoising phase later.

In denoising part, the size of the patches size is fixed to 8×8 . Once we acquire a PG Y , we subtract them with the mean of the PG. The subtracted PG is denoted by \bar{Y} . We can notice that in Figure 1, before subtraction, two PGs have very different local structures. After subtraction, they have similar variations, and that's why the possible number of patterns are extremely reduced, which can benefit GMM learning phase as well. Based on the Many patches which originally have different local patterns may become similar after group mean subtraction.

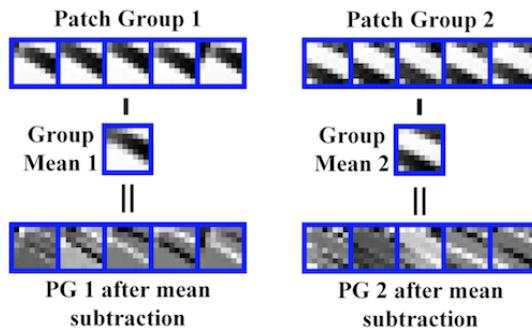


Fig. 1. Different patch groups (PG) share similar PG variations.

In order to find the best fit Gaussian component model for each PG, we check the posterior probability that \bar{Y} belongs to the k th Gaussian component, denoted by $P(k|\bar{Y})$. Finally we simply choose the component which has the highest $P(k|\bar{Y})$ for this \bar{Y} .

After selection for \bar{Y} , SVD is been applied to get the eigenvector matrix \mathbf{D} of Σ which will be used as the dictionary.

$$\Sigma = \mathbf{D} \wedge \mathbf{D}^T, \quad (31)$$

where \wedge represents the significance of these eigenvectors (Figure 2). As we mention before, each patch in one PG has really similar variation after subtraction. So the \mathbf{D} actually represents the variation of the corresponding \bar{Y} .

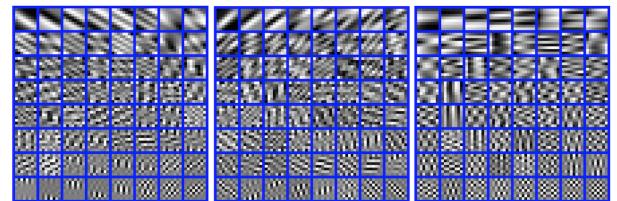


Fig. 2. Eigenvectors \mathbf{D} of 3 different Gaussian components.

After \mathbf{D} is acquired, we sparsely encode each patch \bar{y} in the PG like below:

$$\bar{y} = \mathbf{D} * \alpha + \mathbf{v}, \quad (32)$$

α is the sparse coding coefficient. Since we assume noise \mathbf{v} as white Gaussian noise and α as i.i.d Laplacian distribution. And we already know \bar{y} and \mathbf{D} , so it is not that difficult to get α .

Once acquiring α , we simply get the denoised patch in the PG by $x = \mathbf{D}\alpha + \mu$, where μ is the mean we subtract at the beginning of grouping patches. The big picture is below:

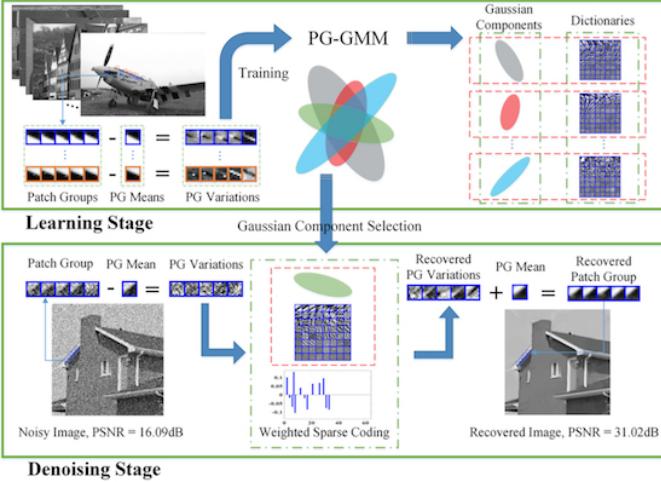


Fig. 3. Whole process of PGPD method.

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. 4 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. 4. 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 II.

TABLE II
CHARACTERISTICS OF THE DIFFERENT TYPES OF NOISE.

Type of noise	Characteristics
Gaussian	$\mu=0$, $\sigma=0.1$
Rician	$\mu=0.05$, $\sigma=0.1$
Uniform	$\mu=0$, $\sigma=0.1$
Salt and Pepper	5% salt, 5% pepper
Speckle	$\mu=0$, $\sigma=0.04$

After applying the described noises in the original images seen in Fig. 4, the obtained noisy images are the ones shown in Fig. 5.



Fig. 5. Noised images. 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. 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 [21].

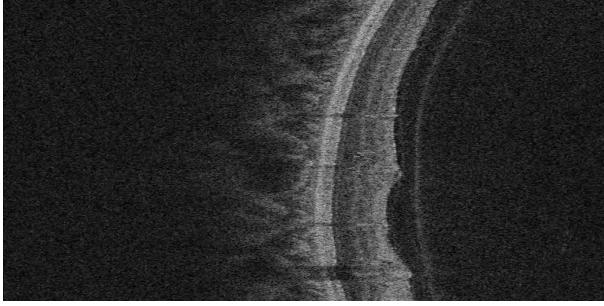


Fig. 6. Retinopathy image.

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 [22]. Finally, the outcome of the segmentation is compared against the segmentation of the noisy retinopathy volume, which can be seen in Fig. 7.

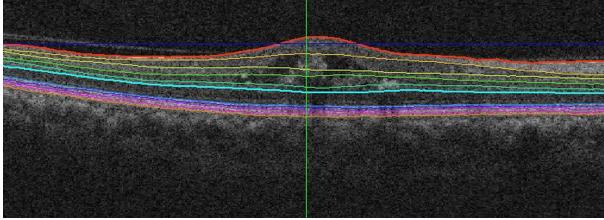


Fig. 7. Segmentation of a noisy retinopathy volume.

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 AND EVALUATION

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. Specific evaluations

Some of the considered algorithms require parameter configuration. Thus, the influence of the parameters as well as their importance for the process is presented in this section.

1) Subspace: The performance of the SDC as a function of the size of the image, as well as its capability in representing real SAR images, is investigated.

Since the proposed SDC is a global technique that operates directly on the whole image, it is essential to investigate its performance with the size of the image. This experiment is conducted by adding speckle noise to the synthetic image in Fig. 8 at different noise levels and by calculating the PSNR gain from the despeckled image.

The results obtained from the average values of 100 trials for the mix image in Fig. 4(a) are shown in Fig. 4(b). Here, the image size is varied from 500×500 to $10k \times 10k$ pixels. The results indicate better performance by the SDC with the growing size of the image. In general, the SDC performance improves by approximately 1.5 dB when the size increases from 500×500 to 1000×1000 and by 0.6 dB when the size increases from 1000×1000 to 2000×2000 . Similar results are obtained with portrait images, like Lena, Barbara, and boat. Generally speaking, the better performance of the SDC with the increased size of the image is mainly attributed to the improved structure of the estimated noise covariance matrix in terms of its diagonality. Close observation of the noise covariance matrix shows nonzero off-diagonal values when the size of the image is small and closer structure to diagonal as the image grows in size.

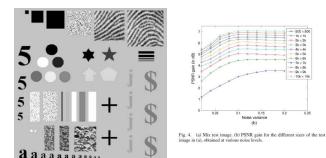


Fig. 8. Synthetic image

To test the capability of the SDC in preserving SAR image details with r -dimensional principal subspace, the algorithm is run with the four images in Fig. 9(a), and the root-mean-square error (RMSE) metric is used to indicate its performance. In this

experiment, the rank values are increased from 500 to around 8000 with a step size of 100, and the RMSE values are calculated and shown in Fig. 9. The results in Fig. 9 clearly show the capability of the SDC in preserving the details of SAR images with a reduced-rank model. It is quite clear that the SDC can represent the four SAR images with an $\text{RMSE} < 0.1$ using rank values of $r \approx 4000$ for the mostly urban area.

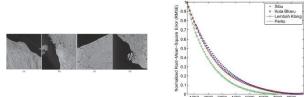


Fig. 9. Results of the SDC on SAR images

2) *K-SVD*: K-SVD algorithm, as presented in Section II-G, 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 influence was the following:

- 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.

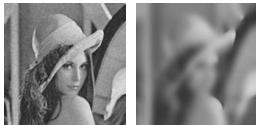


Fig. 10. Denoised images with K-SVD algorithm using different block sizes. On the left the block size is 4 while on the right is 32.

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

- Trainnum: This parameter is related 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.
- 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.
- 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.
- Noise: 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 an homogenous 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.

3) *PGPD*: Besides, we want to talk about denoising using PGPD in SD-OCT images. In the experiment we vary the number of patches per group with initial value for number of patches being 10. Recall that PG is considering not the patches themselves, but patches minus mean of all the patches in the group. In the algorithm we find the correspondence between patch of our noisy image and the group of averaged patches. The results of algorithm performance are as follows:

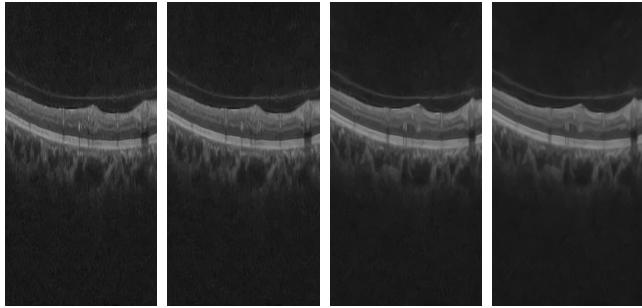


Fig. 12. Retina denoising results with number of patches in group being from left to right : 7, 11, 20, 40.

We can see that with value of patches per group being equal to 15, we reach saturation point where adding more images to group does not give any additional information, but instead makes the image blurry. We can look at this in the following way: when we increase the number of patches we reach the point when average of all patches is no longer similar to patches themselves which ruins data in PG and therefore match is no longer precise.

In this section we are trying to estimate the affect of the PG search window on precision of patch matching. PGPD algorithm searches for a patch of fixed size in the window $N \times N$ and we vary N . The results are as follows:

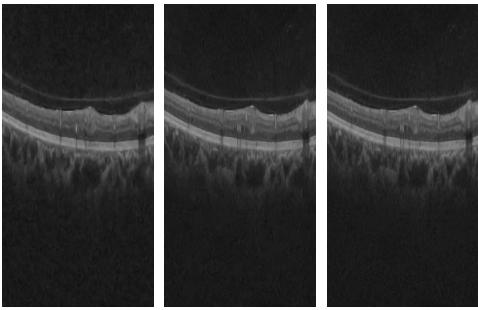


Fig. 13. Patch search window variation from left to right: 5×5 , 15×15 , 30×30 patches around initial patch.

In picture set we can observe that with increasing of the window size we get better results. But the drawback of such improvement is the computational time. We can observe that when reaching the parameter of N being 15, the further improvement of quality is miserable compared to needed computational time.

4) NLM: It is essential to determine the effect of various NLM filter parameters on denoised output. The three parameters that were tested with are filter

size (f), search window (t) and degree of filtering (σ). The search window and filter size do not have any major visual effect on the output. As expected, increasing these parameters increases the execution time. However, σ used in the filter should be close to the original σ of the noise. Choosing very low values for σ parameter can lead to poor denoising, whereas choosing very high values would have the effect of image blurring.

B. Synthetic images

1) Mean filter: Mean filtering is one of the simplest kernel-based smoothing techniques, which simply calculates the average value of pixels within its search window. However, this simplicity did not translate to good denoising results, underperforming most of the advanced methods. This is because it acts as a low-pass filter, discarding higher frequency components with no distinction between noise and signal. In addition, mean filtering affects the quality of the image by smoothing out edges, which degrades parts of the image without much noise. Given its extreme simplicity, the mean filter offers a good benchmark to compare with modern techniques.

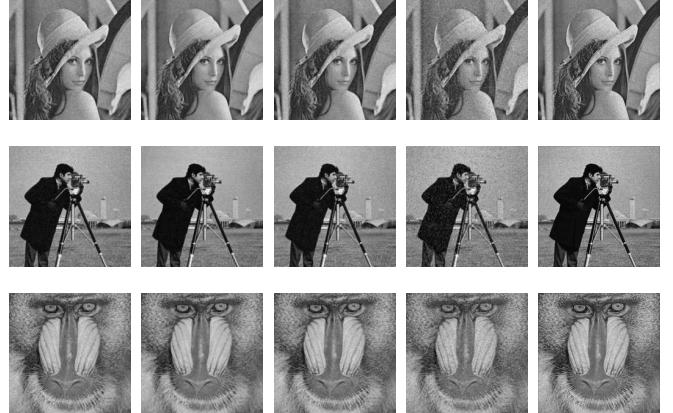


Fig. 14. Denoised images with the mean filter. From left to right: removing Gaussian, Rician, uniform, salt and pepper and speckle noise, respectively.

2) Median filter: This specialized technique is about as simple as mean filtering, but offers far superior results in special cases. The median filter is particularly well suited for noise which varies greatly from image values, such as salt-and-pepper noise. As we can see in the results, the median filter outperformed every other method when applied to pure salt-and-pepper noise, because evaluating the

median always removes outlying values. Unfortunately, median filtering is much less effective on other forms of noise which are more similar to most real-world noise sources. Except for its outstanding results on salt-and-pepper noise, the median filter is inferior to most other techniques on all of the test images.



Fig. 15. Denoised images with the median filter. From left to right: removing Gaussian, Rician, uniform, salt and pepper and speckle noise, respectively.

3) LS filter: When we compare the performance of LS filter with other filters, the results actually are not good at all. it turns out that all the PSNR of denoised output are worse than other advanced methods (expect the other 3 basic filters). This is reasonable because LS filter is at least 25 years older than all advanced methods and technology developed faster year by year.

However, even when we put the results from LS filter and results from other basic filters in the same table, it is still a little disappointing. Expect for wavelet filter which is the worst, LS filter does not outperform mean and median filter. Only for uniform noise, LS filter do a bit better job than median, but still worse than mean filter.

When we consider the reason why this filter cannot perform very well, we find out that, in the flat area, the method can perform as well as mean filter. But edge preservation is a very important characteristics of LS, which is better than other basic filters. And out of the same reason, it would keep the noise around edges as well. That's probably why it is not better than mean filter in denoising area.

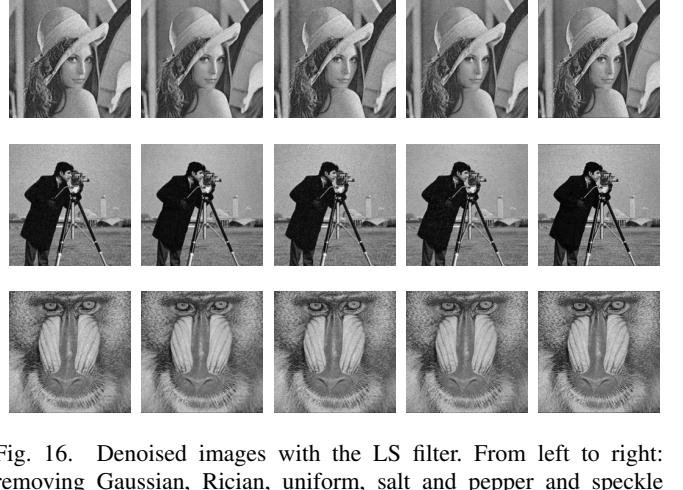


Fig. 16. Denoised images with the LS filter. From left to right: removing Gaussian, Rician, uniform, salt and pepper and speckle noise, respectively.

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

5) Subspace filter: Fig.18 shows the results of the proposed images after the subspace filter is applied.

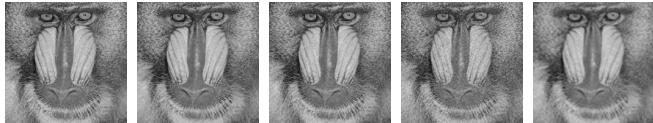


Fig. 18. Denoised images with the subspace filter. From left to right: removing Gaussian, Rician, uniform, salt and pepper and speckle noise, respectively.



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

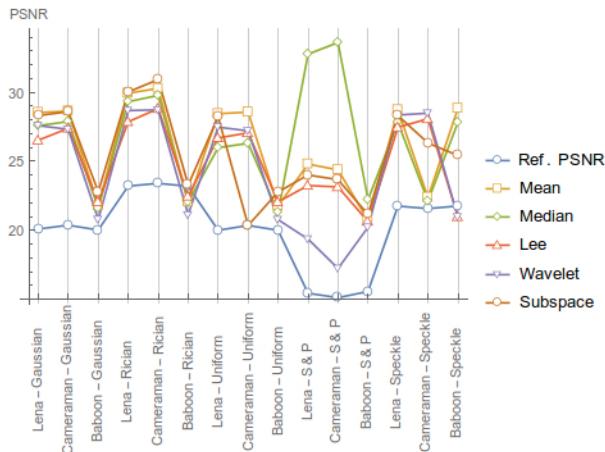


Fig. 19. Graphical comparison of the subspace filter with traditional filtering techniques.

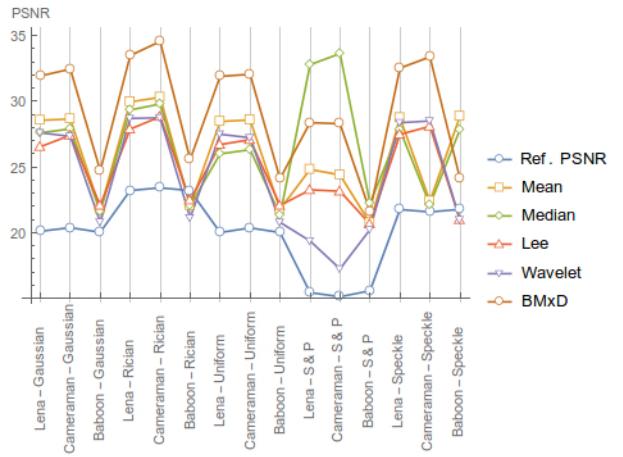


Fig. 21. Comparison of the BMxD filter with traditional filtering techniques.

The denoised images appear sharp and well reconstructed after passing through BM3D algorithm. Only salt and pepper noised images look a bit pixelated.

Values of PSNR are quite important for Gaussian, Rician and Uniform noise. The results exceed those from Mean, Median and Wavelet methods on all the synthetic images. The BM3D method get nevertheless worth PSNR than Median filter on the Salt and Pepper noise, which is a known technique for this type of noise.

Results of PSNR on "baboon" noised images appear less high than on "Lena" and "cameraman", because there is no big uniform regions and no repetitive pattern in this one, which are regions where this algorithm performs good results.

6) *BMxD filter*: Fig.20 shows the results of the proposed images after the BMxD filter is applied.

7) *K-SVD filter*: Fig. 22 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. 22. 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. 23. 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.

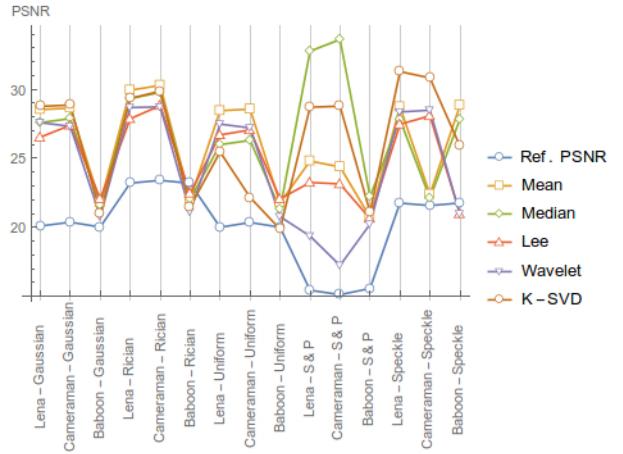


Fig. 23. Comparison of the K-SVD filter with traditional filtering techniques.

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.

8) *NLM filter*: Fig.24 shows the results of the proposed images after the NLM filter is applied.



Fig. 24. Denoised images with the NLM and OBNLM filter. From left to right: removing Gaussian, Rician, uniform, and salt and pepper noise with NLM and speckle noise with OBNLM, respectively.

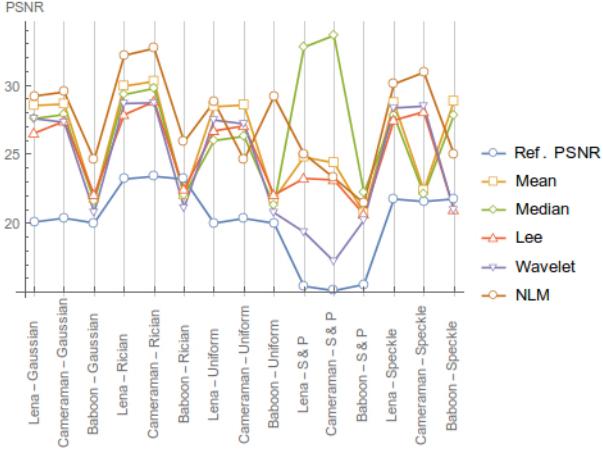


Fig. 25. Comparison of the NLM filter with traditional filtering techniques.

The results for NLM and OBNLM filters are presented in Fig. 24. The human eye is the best testing mechanism to measure the quality of the restored images. The NLM filters produce good qualitative results except falling second to median filters in case of salt and pepper noise. OBNLM produces similarly good results while denoising images with speckle noise.

The NLM filters are not associated with any geometrical shapes. In other words there are no geometrical structures in the residual images. This property allows these filters to work irrespective of the nature of the original image. So, the edges will always be retained which is a big advantage of NLM over several other filters.

The best results for NLM are obtained on periodic and textured images as is the case in the image of the baboon. However, in case of cameraman image, one can see the presence of artifacts in the smooth background of the image. This owes greatly to the fact that for every pixel, the algorithm can find a large set of similar pixels in the neighborhood to compute the new value.

9) *PGPD filter*: Fig.26 shows the results of the proposed images after the PGPD filter is applied.



Fig. 26. Denoised images with the PGPD filter. From left to right: removing Gaussian, Rician, uniform, salt and pepper and speckle noise, respectively.

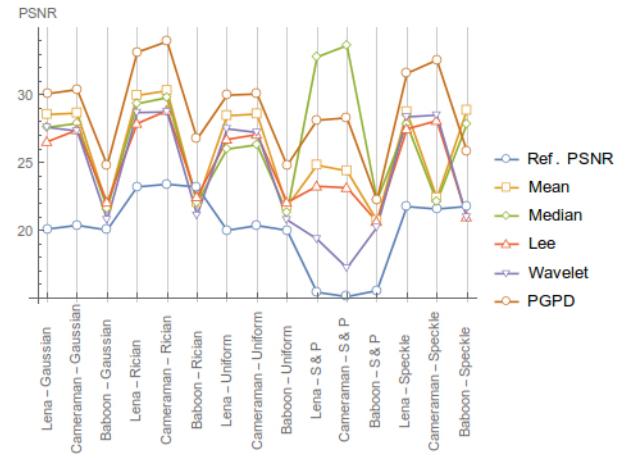


Fig. 27. Comparison of the PGPD filter with traditional filtering techniques.

The PGPD filter have better performance than the median filter in terms of removing noises like Guassian noise, uniform noise and Rician noise. But in case of salt and pepper noise the median filter is showing better performance than the PGPD.

When we compare the PGPD filter with mean filters the PGPD filter is better in removing noises, Gaussian noise, uniform noise, and Rician noise.

In all types of noises we use, the PGPD filter results have better PSNR than the Lee filter. And the image filtered by PGPD looks better than the images filtered by Lee, so we can draw the conclusion that PGPD filter preserves edges better than Lee filter.

PSNR of each de-noised image using PGPD is greater than the one de-noised by the Hard- and soft-thresholding in wavelet domain, except for the

Salt and pepper noise. When we compare the images visually, images de-noised by the PGPD appear better, because of better edge preserving than the Wavelet domain.

We can draw a conclusion that PGPD de-noising performance depends on the type of the noise in the images. The PGPD de-noising techniques provides a greater PSNR than other filters when the noise is Uniform noise, Rician noise and Gaussian noise. When it comes to salt and pepper noise it becomes hard to align patches due to the nature of the noise: salt and pepper give the 100% error per pixel while matching 2 patches which reduces the efficiency of the algorithm.

C. Retinopathy images

In this section the obtained results when denoising retinopathy images are shown and evaluated.

1) *Subspace filter:* -

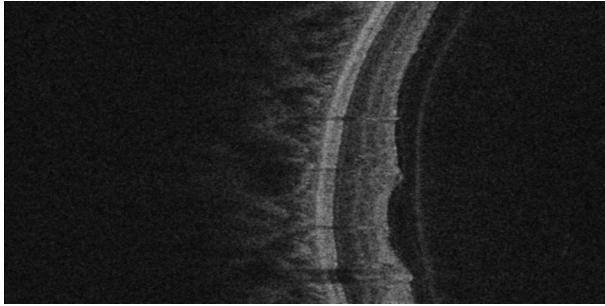


Fig. 28. Denoised retinopathy image with the subspace filter.

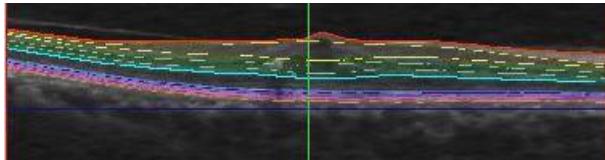


Fig. 29. Segmentation of a retinopathy volume after denoising using the subspace filter.

2) *BMxD filter:* Comparing the results of the denoised retinopathy image with other techniques, the method is still one of the most successful. However, there are less gap between the PSNRs obtained from the retinopathy image denoised with this method and the others, than the PSNRs obtained from the denoised synthetic images with the same methods.

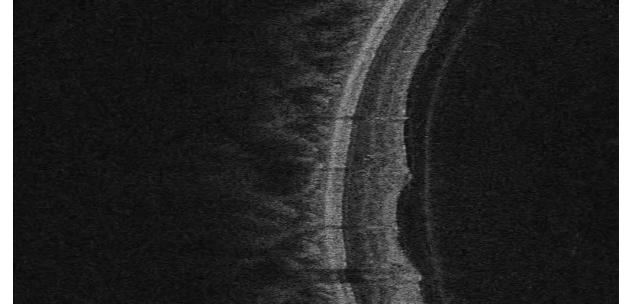


Fig. 30. Denoised retinopathy image with the BMxD filter.

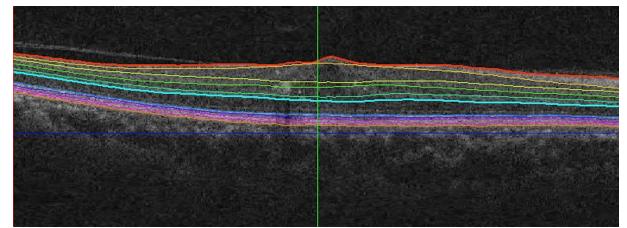


Fig. 31. Segmentation of a retinopathy volume after denoising using the BMxD filter.

3) *K-SVD filter:* 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.

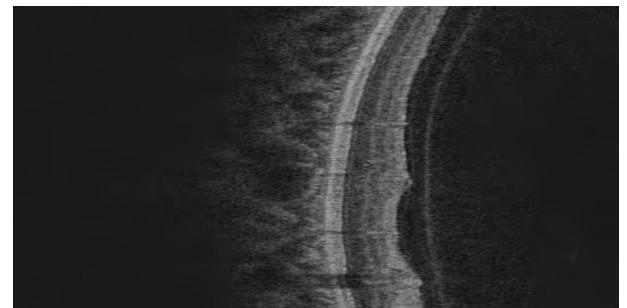


Fig. 32. Denoised retinopathy image with the K-SVD filter.

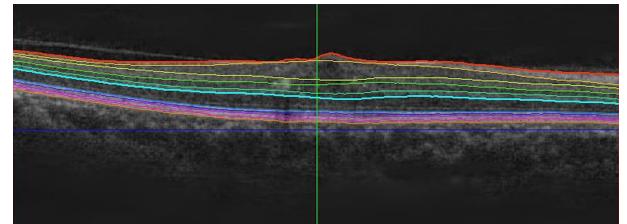


Fig. 33. Segmentation of a retinopathy volume after denoising using the K-SVD filter.

4) *NLM filter*: We denoised the images using Non-Linear Means (NLM) filter, However, when we segmented the images, the output was not much different in comparison to the segmentation of original noisy images.

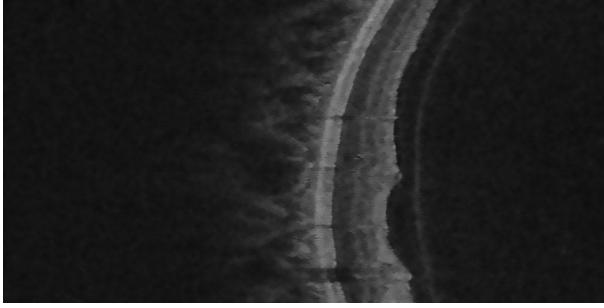


Fig. 34. Denoised retinopathy image with the NLM filter.

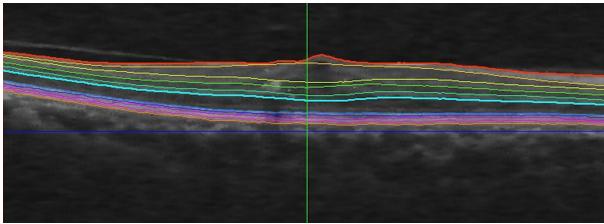


Fig. 35. Segmentation of a retinopathy volume after denoising using the NLM filter.

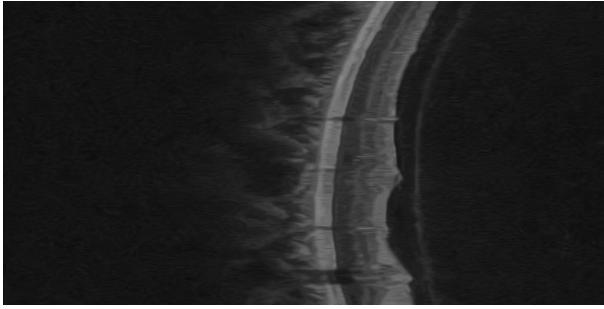


Fig. 36. Denoised retinopathy image with the PGPD filter.

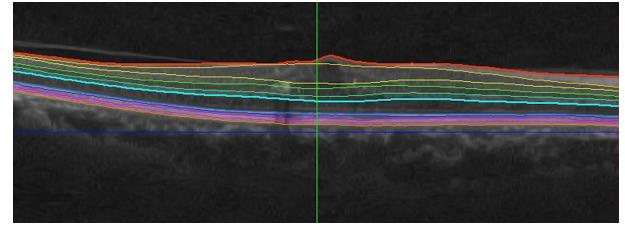


Fig. 37. Segmentation of a retinopathy volume after denoising using the PGPD filter.

6) *SD-OCT Evaluation*: 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. 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. The results are presented in Table III. In terms of this approach, 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.

TABLE III
PSNR FOR DENOISING ALGORITHMS CONSIDERING
RETINOPATHY IMAGES

Technique	Retinopathy
Subspace	24.47
BMxD	25.73
K-SVD	26.68
NLM	23.57
PGPD	23.55

V. FINAL REMARKS

A. Subspace

A subspace-based denoising technique for SAR images has been presented and tested. The proposed technique (SDC) is based on linear estimator and reduced-rank subspace model to estimate the clean image from the corrupted one with speckle noise. The validity of the reduced-rank model in representing different SAR images has been verified and used to enhance the performance of the linear estimator. The capability of the proposed SDC technique in efficiently representing SAR images with reduced-rank values has been discussed and verified. Next, the performance of the SDC has been tested with simulated three-look amplitude data and compared

with Lee and wavelet. The results indicate less noise variance reduction capability by SDC than Lee and wavelet but with less blur, less artifacts, and better preservation of the radiometric edges of the targets.

B. BMxD

The BM3D method was showing good results on synthetic images. The results of the PSNR registered in appendix after denoising synthetic images show most of the time higher values than the other algorithms described in this paper.

Images of the OCT volume were denoised with BM3D. Less noise can be seen on the images. However, values of the PSNR do not show as good results as on synthetic images and we can think that the BM4D algorithm can achieve to better denoise the whole volume since it performs denoising on 3D patches.

C. K-SVD

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 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.

D. NLM

Non Local Means filter is used for denoising images with different noise types like uniform, Gaussian, Rician, and salt and pepper noise. The NLM algorithm performed well when compared with some standard denoising techniques both in terms of the visual quality of the denoised image and the peak signal to noise ratio values.

Optimized Bayesian NLM filter is used for denoising images with speckle noise type. Similar satisfactory qualitative and quantitative results were obtained.

SD-OCT images were denoised using the algorithm. Although the denoising was successful, the subsequent OCT volume segmentation did not show much improvement compared to the segmentation results obtained before the denoising operation.

E. PGPD

PGPD method has better denoising results most of current state-of-art denoising method. In the following, we will analyze the reason.

First of all, based on the successful utilizations of nonlocal self-similarity (NSS) prior in many recent work, it is reasonable for getting good result by using NSS. What's more, instead of simply following the previous work about NSS, PGPD method try to learn NSS prior of clean image rather than noisy image. This could be one of the main distributions to the great performance of PGPD.

Meanwhile, from our point of view, the reason that using GMM gives a great performance is like what we mentioned before, GMM reflects the nature property of clean images. When a method always follows the essence of an object, there is no doubt that they will obtain something quite impressive.

VI. APPENDICES

This appendix includes all the PSNR values of the images shown in Section IV-B. In order to easily compare the performance of the different methods, such results have been grouped according to the different kind of noise that have been analyzed.

TABLE IV
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
Subspace	28.32	28.60	22.79
BMxD	31.92	32.44	24.70
K-SVD	28.77	28.86	21.02
NLM	29.22	29.52	24.62
PGPD	30.07	30.37	24.80
Input noise	20.79	20.37	20.02

TABLE V
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
Subspace	30.03	30.93	23.28
BMxD	33.47	34.54	25.57
K-SVD	29.35	29.87	21.50
NLM	32.17	32.68	25.91
PGPD	33.12	33.92	26.73
Input noise	23.20	23.40	23.19

TABLE VI
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
Subspace	28.23	20.35	22.78
BMxD	31.89	32.04	24.10
K-SVD	28.72	28.78	21.10
NLM	28.81	29.19	24.61
PGPD	29.96	30.05	24.78
Input noise	20.00	20.35	20.00

TABLE VII
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
Subspace	24.01	23.67	21.15
BMxD	28.36	28.28	21.62
K-SVD	25.47	22.16	19.86
NLM	25.02	23.29	21.45
PGPD	28.11	28.29	22.24
Input noise	15.43	15.11	15.56

TABLE VIII
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
Subspace	28.31	26.33	25.42
BMxD	32.51	33.37	24.12
K-SVD	31.29	30.83	25.90
OBNLM	30.10	30.94	25.03
PGPD	31.57	32.55	25.84
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 denoising. In *Wavelets and Statistics, Lecture Notes in Statistics*. New York, pages 125–150, 1995.
- [4] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736–3745, 2006.
- [5] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. *Multiscale Model. Simul.*, 4(2):490–530, 2005.
- [6] A. Efros and T. Leung. Texture synthesis by non parametric sampling. *Proc. Int. Conf. Computer Vision*, 2:1033–1038, 1999.
- [7] M. Lindenbaum, M. Fischer, and A. Bruckstein. On gabor contribution to image enhancement. *Pattern Recognition*, 27:1–8, 1994.
- [8] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Patt. Anal. Mach. Intell.*, 12:629–639, 1990.
- [9] L. Yaroslavsky. *Digital Picture Processing - An Introduction*. Springer Verlag, 1985.
- [10] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D.*, 60:259–268, 1992.
- [11] D. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41:613–627, 1995.
- [12] P. Coupe, P. Hellier, C. Kervrann, and C. Barillot. Nonlocal means-based speckle filtering for ultrasound images. *IEEE Trans. Image Proc.*, 18(3), 2009.
- [13] C. Kervrann, J. Boulanger, and P. Coup. Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal,. *Proc. Conf. Scale-Space and Variational Methods, Ischia, Italy*, pages 520–532, 2007.
- [14] Jun Xu, Lei Zhang, Wangmeng Zuo, David Zhang, and Xiangchu Feng. Patch group based nonlocal self-similarity prior learning for image denoising. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 244–252, 2015.

- [15] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.
- [16] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2862–2869. IEEE, 2014.
- [17] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *Image Processing, IEEE Transactions on*, 16(8):2080–2095, 2007.
- [18] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics (TOG)*, 25(3):787–794, 2006.
- [19] Lev Borisovič Klebanov. *Heavy tailed distributions*. Matfyzpress, 2003.
- [20] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
- [21] Joan Massich Guillaume Lemaitre. Initiative for collaborative computer vision benchmarking. <http://visor.udg.edu/dataset/>.
- [22] Milan Sonka Kimberly A Glynn. Iowa reference algorithms: Human and murine oct retinal layer analysis and display. <http://www.iibi.uiowa.edu/>, 2007.