A Fast Quantum Particle Swarm Optimization Algorithm for Image Denoising Problem

Eman Karam Elsayed[1]*   Dina Refaet Salem[2]   Mohammed Aly[3]

[1]Mathematics and Computer Science Department,
Faculty of Science, Alazhar University (Girls Branch), Cairo, Egypt
[2]Business Technology Department,
School of the Business, Canadian International College, CIC, Cairo, Egypt
[3]Mathematical and Computer Science Department,
Faculty of Science, Zagazig University, Zagazig, Egypt
* Corresponding author's Email: emankaran10@azhar.edu.eg

Abstract: Noise can affect images while acquired, transmitted, stored or compressed. One of the best methods for noise removal is the sparse representation algorithm (SR). The Quantum Particle Swarm Optimization (QPSO) is one of the meta-heuristic algorithms. This paper shows excellent results in noise reduction in the quick version of QPSO, which uses benefit of the SRs and meta-heuristic algorithms. This approach is known as FQPSO-MP, depending on the matching pursuit algorithm (MP). A proposed Dynamic-Multi-Swarm (DMS) and a pre-learned dictionary (FQPSO-MP) method saves the time of calculating the learning dictionary. These modifications contribute to important benefits of computing efficiency (productivity improvements of approximately 90% are achieved) without sacred image quality in comparison with the initial QPSO-MP technique (the bigger reduction relative to the PSNR indexes is lower than 0.58 dB and 0.019). The proposed FQPSO-MP method compared to the original QPSO-MP method after modification. The scientific results show that the FQPSO-MP algorithm is more effective and quicker without sacrificing image quality than the FQPSO-MP algorithm. The experimental results show, in comparison to state-of-the-art denoising algorithms, that both quantitative and image quality results are achieved with the suggested FQPSO-MP method.

Keywords: Image processing, Pre-learning dictionary, Image denoising, Meta-heuristic algorithm, Medical and biological images.

1. Introduction

In the acquisition, transmission, storage or compression of an image, contamination with noise is more obvious Images can be affected with various kinds of noise, like Gaussian, impulsive and mixed sound. Image denoising is an important pre-processing step in image processing [1]. Image processing has several steps in pre-processing. The aim of image denoising is to remove noise from the corrupted images in order to estimate their original image while preserving the relevant features edges, textures, and details [2]. In recent years, the quality of noisy images, corrupted by various kinds of noise, have been improved by various image denoising algorithms. [3, 4] demonstrate a greater summary of the denoising algorithms. Although there are different kinds of image denoising algorithms, their noise removal and running time are not optimal. Three types of image denoising algorithms are founded on the transform filtering of domains, spatial filtering and learning. Examples of the transform domain filter image denoising algorithms are a wavelet, Fourier transforms, Block-Matching and 3D filtering (BM3D) algorithms, etc. [5]. The bilateral, Gaussian, guide, and the nonlocal mean filters are

examples of the second sort of the image denoising algorithms and are focused on spatial filtering [6]. The K-Singular Value Decomposition (K-SVD) [7], Large-Scale Sparse Clustering (LSSC) [8] and the Clustering based Sparse Representation (CSR) algorithms are examples of the following third sort focused on learning. These algorithms are created to improve image quality. In [9] the compressed sensing method demonstrates a novel way to combine both inner and external data for high-performance restoration of magnetic resonance imaging (MRI) data. In [10] the PSNR and the image performance of the Patch Group Prior based Denoising (PGPD) technique is much greater than the Patch Prior based Denoising (PPD) method. Unlike most of the techniques in [11], an energy minimization function is created to mix the low- and high-frequency denoised outcomes. To achieve excellent assessments of the sparsely coding coefficients of the residual image and to make images denoise while maintaining their textures, a robust algorithm mixing gradient histogram with SR is used [12]. Completive denoising efficiency and high quality images can be achieved in comparison to other denoising techniques by an iterative weighted sparse representation (IWSR) [13]. The objective of a dual domain filter (DDF) is to enhance the SAGD image quality [14]. For low quality optical remote-sensing images [15], the progressive DDF is applied. There are two components to DDF. The first way to achieve high quality images is to use a bilateral filter as the guide filter. Then a Fourier transform is recovered and improved with an adaptive adjustment parameter by low-contrast helpful constructions. The overall weighted nuclear norm minimization (WNNM) denoising framework is provided in [16]. There's several desirable characteristics within this framework. First, the noise impact can be reduced by means of a thresholding scheme with a 2D orthogonal transformation on the similitude measures of different parts. Secondly, it requires into consideration the significance of method noise and offers an adaptive linear strategy to process noise feedback based on noise levels. A speckle removal algorithm is shown depending on the theory of Grey and WNNM algorithm. The benefits of Gray Theory can efficiently maintain the image's texture data and the WNNM can efficiently remove the speckle noise are considered in this algorithm. Concurrently, both algorithms avoid weaknesses. The method is finally implemented to the remote sensing image speckle removal.

The essential to the SR-based denoising algorithm is to know the appropriate dictionary to suit the local image structure. The SR-based denoising algorithm has been successful if the dictionary has to do with the results of sparse coding and if it suits the image features. The dictionary is split into two kinds; the first sort of the dictionary is the fixed dictionary like wavelet, Gabor, Gabor wavelet, log Gabor, log Gabor wavelet, and discrete cosine transform (DCT). The second kind is the learning dictionary like K-SVD [7], LSSC [8], principal component analysis (PCA) [17] and locally learned dictionary (KLLD) [18]. The atoms of the adaptive dictionary are iteratively created. It's still a challenge for scientists to use the learning dictionary in the denoising algorithm. In [6] the K-SVD method is a long SVD execution time and iterative. For a normal problem, the CPU needs nearly a few minutes or more for the dictionary learning. Therefore, the application of the K-SVD algorithm is encumbered in large-scale images. The use of the learning dictionary is still an issue in the denoising algorithm. Researchers face a challenge in finding an effective technique of image denoising. SR Algorithm is the most frequently used algorithm for Gaussian noise denoising images. In order to achieve the optimal quality of the regulating parameter of the total variation technique, the Particulate Swarm Optimisation (PSO) shall be used to remove a noise from image MRI [19]. The Non-locally Centralized Sparse Representation algorithm (NCSR) incorporates concepts of SR (learning dictionary) and non-local self-similitude in image patches. The NCSR incorporates these concepts in a given image.

The dictionary must be derived from the entry data, which constitutes a main principle for dictionary learning. Due to the reality that the image processed typically wishes to depict the input data using as few parts as feasible, the emerging sparse dictionary learning techniques were encouraged. Prior to that strategy, predefined dictionaries like Fourier or wavelet transforms had to be used in the general practice. In some cases, however, a dictionary trained in suiting the input data can greatly improve the sparsity of the data decomposition, compression and analysis applications used in the fields of image denoising, classification, Video and audio. Image compression, image fusion and inpainting are the applications of sparsity and over-complete

dictionaries. Most current dictionary learning techniques regard an over-complete dictionary, e.g. the K-SVD method. They often involve the resolution of a minimum issue that is extremely difficult in aspects of computational feasibility and quality. However, if the correlations between dictionary atoms are not well limited, a dictionary's redundancy does not inherently enhance sparse code efficiency. A quick orthogonal dictionary learning technique is suggested to explore sparse image representation. This technique is significantly more computationally efficient than over-complete dictionary based learning techniques, given similar results in many image recovery tasks. In an alternative minimization system, the new K-SVD method [7] teaches both an over-complete dictionary and sparse representations of the patches under that dictionary. Getting started with the set of disconnected image patches gathered from the original image, the K-SVD technique alternately executes between two sub-problems: sparse coding and dictionary updating. Almost all sub-problems in [7] are dependent on heuristic sampling techniques: the sparse coding under the over-complete dictionary is achieved via OMP and the dictionary is measured by column-wise sequential SVD changes. In [20] the over-complete dictionary was tested using the K-SVD algorithm, accompanied by the Batch-Orthogonal Matching Pursuit (Batch-OMP) rather than the OMP algorithm to improve the denoising algorithm.

This paper aims at reducing the computational complexity of the adaptive dictionary and improving the denoising efficiency of the test images. The paper offers a new speedy QPSO algorithm based on the MP and the pre-learned dictionary generated by the Translation, Libration, and Screw (TLS) model. The QPSO algorithm has two improvements that reduces the computational complexity and the execution time.

Two significant assessment indices for denoising performance are generally both noise reduction and computational efficiency. That is to say, most denoising apps involve an agreement between noise reduction and computational cost. This paper is mainly aimed at researching the effectiveness of decreasing the computational complexity of dictionary learning, one of the longest-term modules of the initial QPSO-MP algorithm. Detailed experimental findings demonstrate that, compared with their naive implementation and other state-of - the-art denoising algorithms, our fast QPSO-MP

(FQPSO-MP) method, with two technical advances, is highly competitive performance with respect to noise reduction and computational complexity.

The efficiency of BM3D is well known as a benchmark algorithm due to its ability to use block similarity and sparse representation. However, as BM3D utilizes set square blocks with a fixed scale and square shape across the image, it has restricted performances, in particular on edges with a strong contrast. The variety BM3D-SAPCA, that adopts a neighborhood adaptive shape strategy, enhances visual quality efficiency as well as significantly increases execution time. The proposed FQPSO-MP is somewhat higher than the visual quality BM3D-SAPCA in comparison with BM3D-SAPCA, but reduces the time of execution by a factor of 2-8.

The rest of this paper is structured in the following way, the proposed FQPSO-MP method with two modifications over the original QPSO algorithm is discussed in section 2. The experiment results and efficiency of the proposed FQPSO-MP algorithm and the other denoising algorithms are tested and evaluated in section 3. The paper is concluded in section 4.

2. Methodology

2.1 Matching pursuit (MP)

MP algorithm uses the signal decomposition based on a redundant dictionary. Each element of the dictionary called an atom. When MP is utilized, the best atom of the dictionary can be detected. These atoms can be found in each iteration. For an arbitrary image y of size $b \times c$, let $\{d_\gamma\}_{\gamma \in \Gamma}$ are the atoms of the dictionary D, where Γ is the set of all indexes γ and $||d_\gamma|| = 1$ [21]. The approximation of y by projecting it on a vector $d_{\gamma_0} \in D$ is the first step of the MP.

$$y = <y, d_{\gamma_0}> d_{\gamma_0} + r_y \qquad (1)$$

Where the projection of y into the atom d_{γ_0} is $<y, d_{\gamma_0}> d_{\gamma_0}$, and the residual of the original image y is r_y, where the r_y is orthogonal to d_{γ_0} [21]:

$$||y||^2 = |<y, d_{\gamma_0}>|^2 \, ||d_{\gamma_0}||^2 + ||r_y||^2 \qquad (2)$$

$$||y||^2 = |<y, d_{\gamma_0}>|^2 + ||r_y||^2, \qquad (3)$$

where $||d_{\gamma_0}||^2 = 1$

The term $\|r_y\|^2 = \|y\|^2 - |<y, d_{\gamma_0}>|^2$ should be minimized, so $d_{\gamma_0} \in D$ must be selected to maximize the term $|<y, d_{\gamma_0}>|$. The image y can be remodelled via the selected atom d_{γ_n} after N iterations

$$y \approx \sum_{n=0}^{N-1} <r_y^n, d_{\gamma_n}> d_{\gamma_n} \qquad (4)$$

The essential step in the denoising processes is constructing d. We used the Gaussian function (Eq. (5)) to construct the fixed dictionary of the proposed FQPSO-MP algorithm. The essential function is a Gaussian in one axis and the second derivative of the Gaussian in the various another axis. In this paper, the TLS model is applied the Gaussian function to construct the pre-learned dictionary $d(m, q)$. The Gaussian function is presented as the following formula:

$$d(m, q) = (2 - 4m^2)e^{-\frac{1}{4}(m^2 + q^2)} \qquad (5)$$

Eq. (5) is applied to construct the dictionary $d(m, q)$ with the TLS rotation in x direction and y direction, i.e. $T[L]_x, T[L]_y, L_x, L_y, S_{[L]}$ respectively. The TLS model applied to calculate m, q

Where, $\quad m = \frac{(cos(S_{[L]}) \times T[L]_x + sin(S_{[L]}) \times T[L]_y)}{2^{\left(\frac{stx}{NN}\right)}} \qquad (6)$

$$q = \frac{(cos(S_{[L]}) \times T[L]_y - sin(S_{[L]}) \times T[L]_x)}{2^{\left(\frac{sty}{NN}\right)}} \qquad (7)$$

Where $T[L]_x, T[L]_y, S_{[L]}, stx, and\ sty$ are a translation in x direction, y direction, screw rotation angle, arbitrary columns of the population respectively, and NN is the number of parameters of TLS model ($NN = 5$).

2.2 Quantum-behaved particle swarm optimization (QPSO)

QPSO algorithm is roused by the familiarity with quantum mechanics with the first PSO algorithm [22]. QPSO is the refreshed and upgraded version of PSO algorithm, not only in its search capacity but also in its precision. The particles of QPSO algorithm can show on any search space and these particles rely upon the delta potential. The QPSO algorithm can skip the mistake of the standard version of PSO algorithm, however the global convergence with probability 1 can't be ensured. At the same time in the quantum search space, the positions and velocities of the particles can't be calculated. The wave function

$\psi(X, t)$ used to calculate the state of the particle. In the certain position, the formula $|\psi(X, t)|^2$ used to find the probability of the particles, then the probability distribution function can be measured. Each particle must have position, where the position of each particles can be calculated and updated using the following Eq. (8):

$$X_{id} = p_{jd} \pm 0.5\ A \ln\left(\frac{1}{u}\right) \quad u \sim U(0,1) \qquad (8)$$

Where P_{jd} and $U(0,1)$ are the local attractor, and random number respectively. The value of the random number $U(0,1)$ between 0 and 1. The P_{jd} can measured by the following Eq. (9).

$$p_{jd} = \beta P_{jd} + (1 - \beta)P_{gd} \quad \beta \sim U(0,1) \qquad (9)$$

Where the best position of the jth particle is defined as $P_j = (P_{j1}, P_{j2}, P_{j3}, \ldots, P_{jd})$, the global position of all particles is defined as $P_g = (P_{g1}, P_{g2}, P_{g3}, \ldots, P_{gd})$; and β is the random number. The value of β distributed between 0 and 1. The parameter A measured by Eq. (10).

$$A = 2\alpha.|mbest_d - X_{jd}| \qquad (10)$$

Where the average optimal position of all the particles is $mbest$. The $mbest$ is measured by following Eq. (11).

$$mbest = \frac{1}{M}\sum_{j=1}^{M} pbest_j \qquad (11)$$

The parameter α is the contraction-expansion coefficient. The parameter α can be calculated as the following Eq. (12).

$$\alpha = 0.5 + 0.5 \times \frac{(L_c - C_c)}{L_c} \qquad (12)$$

Where L_c and C_c are the total number and the current number of iterations respectively.

2.3 TLS model

Cruickshank was the first to use the TLS model to describe the movements of the atomic groups. In the TLS model, the T, L and S, three matrices are defined which can define the combination of the atomic vibrations and liberations. The TLS model used in the molecular systems [23]. Matrix components should meet several requirements for transformation into vital molecular motion. The matrices may not describe concerted molecular

movements if these conditions are not met. The atoms can be given in a few bases denoted [M], and the [L] basis of the TLS matrices can be calculated using the following formulas:

$$L_{[L]} = R_{ML}^T L_{[M]} R_{ML};$$
$$T_{[L]} = R_{ML}^T T_{[M]} R_{ML};$$
$$S_{[L]} = R_{ML}^T S_{[M]} R_{ML}. \qquad (13)$$

Where R_{ML} is the transition matrix from basis [M] into basis [L], R_{ML}^T is its transpose, and $L_{[L]}$, $T_{[L]}$, and $S_{[L]}$ are liberation, translation, and screw rotation matrices in the [L] basis . In addition, the matrices of the TLS in the [M] basis can be calculated using the following formulas:

$$L_{[M]} = R_{ML}^T L_{[L]} R_{ML};$$
$$T_{[M]} = R_{ML}^T T_{[L]} R_{ML};$$
$$S_{[M]} = R_{ML}^T S_{[L]} R_{ML}. \qquad (14)$$

Where, R_{ML} is the transition matrix from basis [M] into basis [L], R_{ML}^T is its transpose , and $L_{[M]}$, $T_{[M]}$, and $S_{[M]}$ are libration, translation, and screw rotation matrices in the [M] basis.

- Liberation (L_x, L_y): the atom should be locally oriented along the contours.
- Translation (T_x, T_y): moving the atom across the image.
- Screw rotation (S): to take the smoothness of the contour.

The Eqs. (13) or (14) are applied to find h, w. In this paper, the Eq. (13) is substituted in Eqs. (6) and (7) to calculate h, w.

2.4 Pre-learned dictionary

The dictionary D is reconstruction at every outer loop iteration in the original algorithm. The denoising or intermediate noisy image extracting to training patches and these training patches are grouped into C clusters and each cluster is learned a PCA sub-dictionary. Then a compact PCA sub-dictionary is chosen to code it for a particular patch. Because constructing of dictionary from image patches takes place at runtime, which is comparatively high time complexity, particularly for large image size. In reality, some factor should be satisfied in order to construct the dictionary efficiently depicts the image content: (1) training the dictionary itself on the corrupted image (generated at runtime and iteratively updated, later referred to as the learned dictionary, taken in the original algorithm) and (2) directly training the dictionary on a high-quality image

instead of a noisy one (generated offline without updating, later referred to as pre-learned dictionary). In this paper, the pre-learned dictionary is used instead of a built-in learned dictionary runtime, which significantly decreases the runtime.   Each inner iteration in the original algorithm involves decomposing the noisy image into overlapped patches, coding the image patches sparsely, estimating the sparse coefficients, and rebuilding the denoised image from the approximate patches. The dictionary performs a significant part in these m odules. As we understand, reaching the original image equivalent to the noisy image is not practical. The dictionary should generate from another smooth image when applying the original algorithm to denoise image, where its components differ from the contents of the corrupted image. The regeneration mistake can therefore usually not be determined by the change of the representation of the image throughout the denoising method. Table 1 demonstrates the pre-learned dictionary trained to detect the value of the retrieved images on some high-quality images. In Table 1, it is obvious that all PSNR outcomes are more than 138 decibels, meaning the performance of the image being recreated is adequate. The TLS model in this paper utilizes the pre-learned dictionary instead of the constructed dictionary throughout runtime, which demonstrates why the runtime is considerably decreased. The adaptive dictionary does not always obtain the highest outcomes in regeneration. The rebuilding mistake of the sample images in the two types of dictionaries is an irrelevant change. The contents of the test images in Fig. 1 are obviously different from the contents of the test images in Figs. 2 and 3 that are also suitable for measuring the stability of the pre-learned dictionary strategy. The denoising outcomes of the original method are not susceptible to the details of the test images used to construct the dictionary. To accelerate the QPSO-MP algorithm, the TLS model is often used as a pre-learned dictionary.
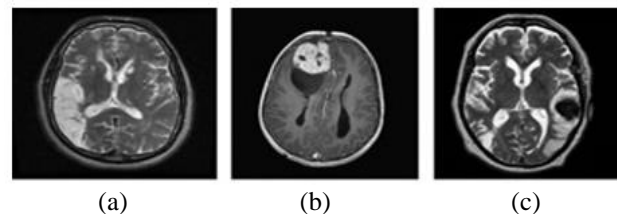

(a)   (b)   (c)
Figure.1 (a) Img1, (b) Img2, and (c) Img3. The size of all images is 512×512

(d)   (e)   (f)

Figure.2 (d) Lena, (e) Cameraman, and (f) Couple. The size of all images is 512×512

(g)   (h)   (i)
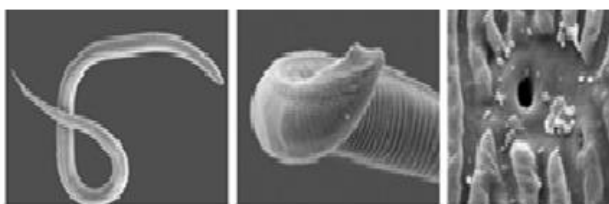
Figure.3 (g) Meloid. Kralli, (h) Meloid. Duyts, and (i) Meloid. Pore. The size of all images is 490×330
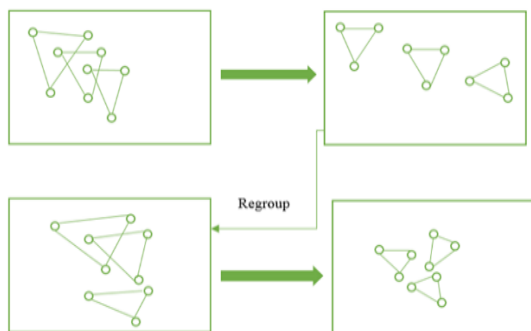

Figure.4 DMS-QPSO's search

2.5 DMS of population

The DMS-QPSO approach is based on the QPSO algorithm's normal variant. The distinction between both the QPSO and DMS-QPSO algorithms is that the swarms are dynamic and lower in size [24]. The DMS-QPSO algorithm's neighbourhood topology has two essential features; The DMS splits the entire QPSO algorithm population into tiny swarms. These swarms are repeatedly re-clustered using different regrouping plans, and these swarms data is shared between them. Fig. 4 illustrates the search for the DMS-QPSO algorithm.

2.6 The proposed FQPSO-MP algorithm

A first enhancement is to use the TLS model to construct the pre-learned dictionary from high-quality, textured images rather than the over-complete dictionary produced from the specified noisy image as its own or from the incompletely denoised image. The second enhancement is to use the DMS over the QPSO (DMS-QPSO) algorithm.

While using DMS, the population is split into tiny parts. Every group should explore the space using its own participants. Throughout this method, there will be no reduction in the diversity of an entire population as the information generated by each swam will be swapped. In R generation, the members of each swarm group will distribute randomly (i.e. R is the period of the Regroup). Table 2 provides the proposed FQPSO-MP algorithm with two significant changes to the original QPSO-MP algorithm.

3. Results and discussion

3.1 Comparison between the proposed and the original QPSO algorithms

On the BSD500 [25], BITE [26], and nematode [27] databases, we assess the suggested FQPSO-MP technique. We developed a dataset of 150 images from the database of the BSD500 and performed a statistical test on these images. These sample images include several kinds of scenes, including indoor, outdoor and portrait scenes. The BITE database objective is to communicate in vivo medical images of brain tumor patients to promote the creation and verification of latest methods for image processing. These data can be found internet on the Brain Tumor Image database of the Montreal Neurological Institute. It includes images of 14 patients with ultrasound and magnetic resonance. Every patient was subjected to a preoperative and postoperative T1-weighted magnetic resonance scan with enhanced gadolinium, and various intraoperative B-mode pictures were obtained pre and post resection. Some of the images were used for previous publications. This is why the data is split into three classes with their own properties and features; 1) Ultrasonic images pre and after resections; 2) Ultrasonic pre and after resection images and 3) MR images pre and after resection. In order to facilitate the development and validation of new image processing algorithms, the aim of the nematode database is to share bio images. The data is accessible internet for the assessment database at WAGENINGEN UNIVERSITY & RESEARCH. It includes various nematode images. To ensure reasonable contrast of the FQPSO-MP algorithm suggested to the local version of the QPSO method, all the conditions are prepared the same way, the patch size is 8×8, with similar test images all circumstances are designed in the same manner. The performance of denoised images is measured by the PSNR. The SSIM index has been evaluated in order to assess the performance of denoted images more reliably. The performance and time of execution of the denoised images must be

measured. The computational complexity of the proposed FQPSO-MP and QPSO-MP algorithms is assessed. The experiment is conducted on a DELL PC, running Windows 7 Enterprise with 2.4 GHz Intel Core (TM) i5-M520 CPU, (R), MATLAB R2016 64-bit and 4 GB RAM.

3.1.1. Complexity analysis

In the present subsection, the QPSO-MP algorithm's time complexity is analysed. The primary computational costs of the QPSO-MP method are as follows. The performance of the denoising algorithm QPSO-MP is heavily dependent on dictionary learning, particularly if it increases the size of the dictionary or the number of training patches. As we do now, one of the reasons for low efficiency of the original algorithm a non-locally estimated sparse coefficients (NESC) are strictly dependent upon a matching patch that is also time-consuming process in order to identify a patch that is sufficiently similar to that used for a diagnosis in the original algorithm. An isotropic atom that is used to build the dictionary in the original algorithm is time consuming processes, which is another reason for low efficiency of the original algorithm. Although the original algorithm is very good for reducing noise, in large-scale images with growing difficulty, it is computationally expenses and eventually restricted. In addition, the outcomes of the QPSO-MP method are not susceptible to the content of the dictionary test images. The pre-learned dictionary is therefore used to accelerate the original algorithm. The QPSO-MP denoising algorithm requires $O(dM^2)$ complexity, with d as the pixel amount for the patch, and M as the pixel amount for the image. The duration required to denoise a single image 512×512 is therefore too lengthy. For 512×512 images, the runtime for an outer loop is approximately $457\ s$ with unoptimized MATLAB codes on a PC equipped with 2.4 GHz Intel Core (TM) i5-M520 CPU and 4 GB RAM. In a single external loop, the calculation of dictionary learning and weighting costs around 70 s and 390 s, which takes around 1096 s to denoise a 512×512 image. While, the total running time of the proposed FQPSO-MP algorithm increases by 1.4 % when the size of the 512×512 test image increases by 15%. In order to decrease the computation time without too much sacrificing image quality, both changes (DMS population and pre-learned dictionary) are used to enhance computing performance of the original QPSO-MP algorithm. The role of the pre-learned dictionary construced by the TLS model is effective in reducing the running time of the proposed FQPSO-MP algorithm. Table 4 provides the execution time of

the proposed FQPSO-MP and the original QPSO-MP algorithms on the test images. The execution time of the proposed FQPSO-MP algorithm is less than the execution time of the QPSO-MP algorithm. Replacing the adaptive dictionary (Anisotropic atom) with the pre-learned dictionary is the explanation for this distinction in running time. The pre-learned dictionary can be used directly with the proposed FQPSO-MP algorithm (it doesn't take long to run) while the time required for image generation to run is more than 70 s. This implies that the FQPSO-MP algorithm's computational complexity is less than the local version of the QPSO-MP algorithm. Fig. 6 illustrates the average performance of the proposed FQPSO-MP algorithm that is surpasses the performance of the QPSO-MP algorithm. Fig. 5 displays the average execution time of the proposed FQPSO-MP algorithm, which is lower than the corresponding score of the original QPSO-MP algorithm. This demonstrates the possible advantages of the FQPSO-MP algorithm.

3.1.2. Noise reduction

The proposed FQPSO-MP and QPSO-MP algorithms are summarized in Table 4 by a PSNR values of denoised images. Because of its minimum
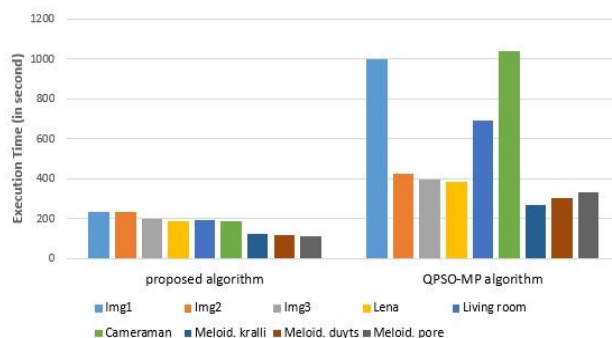

Figure.5 Average execution time on all the test images using the proposed and original QPSO-MP denoising algorithm
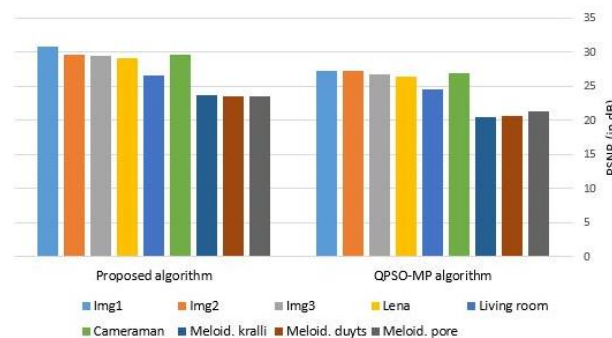

Figure.6 Average PSNR results on all test images using the proposed and original QPSO-MP denoising algorithm

running time and the optimal results of denoised images, the proposed FQPSO-MP method can be used for large images. The FQPSO-MP algorithm's average values of PSNR are higher than those of QPSO-MP algorithm. The principal cause for this is the population split into groups in the suggested FQPSO-MP algorithm. From Table 11, differences between the original QPSO-MP and proposed FQPSO-MP algorithms can observe are very small. The largest decline is less than 0.58 dB and 0.019 with respect to PSNR and SSIM indices, respectively. The distinction is not easy to discern between denoised results. When visual inspection is carried out, the results of the two algorithms are very near. This means that the proposed FQPSO-MP algorithm works as well as the original QPSO-MP algorithm and reduces running time significantly. The differences between the denoised results shown in Figs. 7 and 8 are between the proposed FQPSO-MP and QPSO-PM algorithms at the same sigma σ and same image. The proposed FQPSO-MP algorithm implements the QPSO-MP algorithm as well as reducing the runtime. The summary of this test indicates that, instead of the adaptive dictionary acquired in the corrupted image throughout runtime, the best-denoised images can be obtained by the pre-learned dictionary. The TLS model can be applied to construct the pre-learned dictionary that suits most the medical, biological and the natural images. In brief, the final results demonstrate that the proposed FQPSO-MP algorithm is great for the removal of noise (noise reduction) and running time (computational complexity).
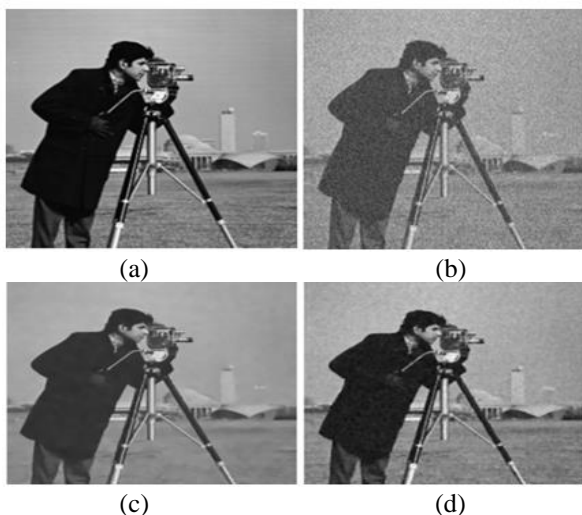

Figure. 7 Denoised results on Cameraman image with moderate noise corruption (σ = 30) using the proposed and QPSO-MP algorithms, respectively: (a) original image, (b) corrupted image, (c) denoised image using the proposed algorithm, PSNR= 30.77, and (d) denoised image using the QPSO-MP algorithm, PSNR=29.16
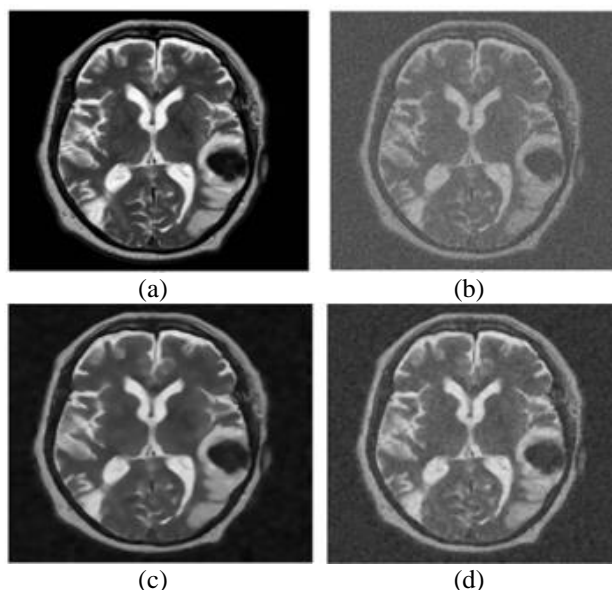

Figure. 8 Denoised results on image 3 with moderate noise corruption (σ= 50) using the proposed and QPSO-MP algorithms, respectively. (a) Original image; (b) corrupted image; (c) denoised image using proposed algorithm, PSNR= 28.38 (d) denoised image using the QPSO-MP algorithm, PSNR=25.68

3.2 Comparison between the proposed FQPSO-MP and the state-of-the-art algorithms

The denoising performance of the proposed FQPSO-MP algorithm is comparable to the state-of-the-art denoising algorithms, 1) K-SVD [7], 2) Bilateral [5], 3) Wiener [28], 4) Total Variation (TV) [29], 5) Median [30], and 6) Bayes [31]. The software package of the earlier algorithms is used in this paper to produce denoising performance results. To guarantee a suitable compare between both the proposed FQPSO-MP algorithm and the state-of-the-art algorithms, the tested images [25-27] are the same, having the same size and damaged by the additive Gaussian noise (GN) $N(0, \sigma^2)$, where σ^2 is the estimated noise deviation with noise levels $\sigma = 10, 30, 50, 70$ and 90. Six separate (the state-of-the-art) methods just under the distinct Gaussian noise rates are summarized in Tables (5, 6, and 7) in denoising outcomes. The proposed algorithm FQPSO-MP provides the highest outcomes for performance. Fig. 9 shows the general average output of the algorithms being evaluated. Figs. 10 and 11 provide the distinction between the denoised results of the suggested FQPSO-MP and the state-of-the-art algorithms at the same sigma σ and at the same image. The test findings of this paper demonstrate that there is the shortest implementation time for the suggested FQPSO-MP algorithm. But at the other hand, the
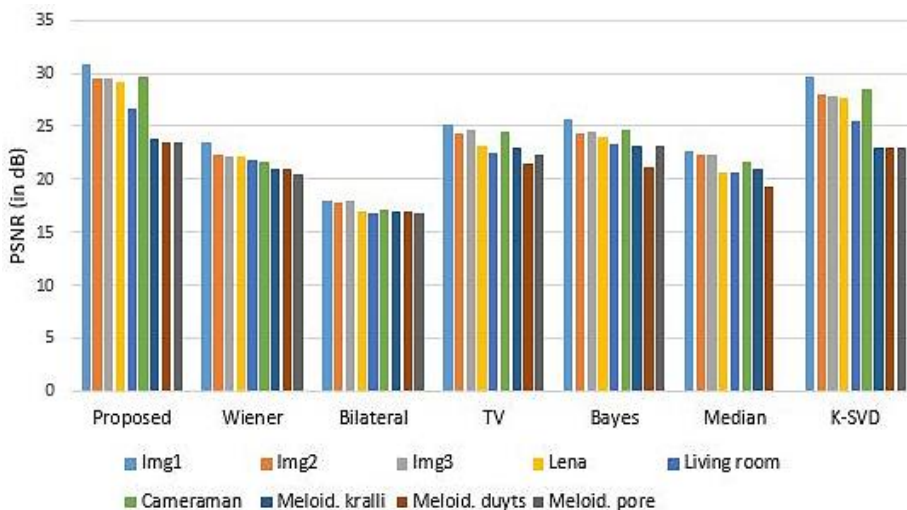
Figure.9 The average PSNR results on the tested images using the proposed and the state-of-the-art approaches
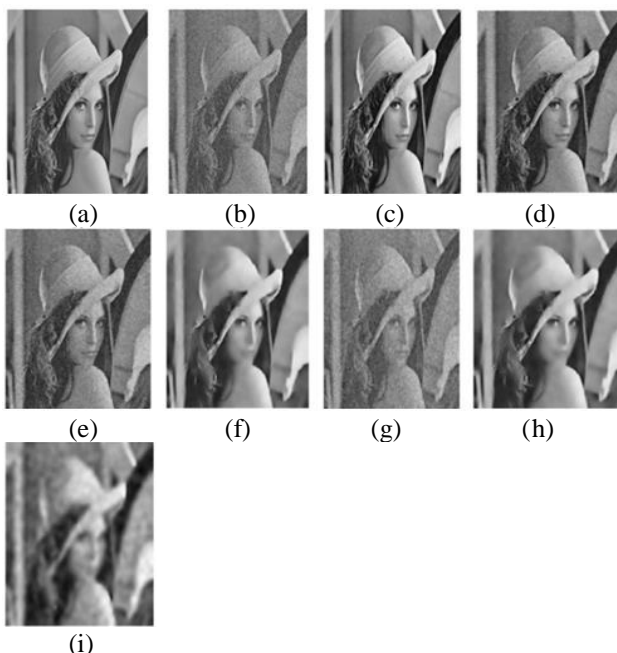

Figure. 10 Denoised results on the Lena image with moderate noise corruption ($\sigma = 30$) using the proposed and state of-art approaches: (a) original image, (b) corrupted image, (c) denoised image using the proposed algorithm, PSNR= 30.61, (d) denoised image using the K-SVD algorithm, PSNR=29.24, (e) denoised image using the TV algorithm, PSNR= 26.97, (f) denoised image using the Bayes filter, PSNR=26.03, (g) denoised image using the Median filter, PSNR=22.69, (h) denoised image using the Wiener filter, PSNR=24.59, and (i) denoised image using the Bilateral filter, PSNR= 18.70

Bilateral and the Wiener filters are faster than the FQPSO-MP algorithm, whereas the FQPSO-MP proposed average performance is greater than the other algorithms. The proposed FQPSO-MP algorithm achieves a certain balance among the performance of denoised images and the running time. The proposed FQPSO-MP algorithm is implemented to medical, biological and natural images, to fulfill and test various kinds of images. Also, the proposed FQPSO-MP algorithm is implemented to different extension types of image such as jpeg, bmp, png, and tif. Generally, it has been shown that the suggested FQPSO-MP algorithm is a stable algorithm.

Table 8 illustrates the average SSIM performance using seven different algorithms under the different noise levels on the 150 test images. The overall outcomes of the Tables (3, 4, 5, 6, 7, 8, and 9) indicated that, as compared with the remaining algorithms for separate images, a suggested FQPSO-MP achieved superior outcomes in terms of runtime and noise reduction. Notions on the performance results of FQPSO-MP are comparable to other state-of-the-art algorithms, 1) BM3D [12], 2) LSSC [8], 3) NCSR [17], 4) edge-preserving image denoising (EPID) [11], 5) WNNM [16], 6) SNLM [32], 7) BM3D-SAPCA [33], 8) FastNLM [34], and 9) FNCSR [65]. Table 9 illustrates the average PSNR performance using nine different state-of-the-art algorithms under different noise levels on the 150 test images. In addition, this part compares the execution time and the performance outcomes between the proposed FQPSO-MP algorithm with seven denoising algorithms such as SNLM [32], BM3D [12], BM3D-SAPCA [33], FastNLM [34], FNCSR [35], K-SVD [7], and WNNM [16] to demonstrate the efficiency of the proposed FQPSO-MP algorithm in large-scale image verity. We had produced 150-image from the BSD500 [25] to guarantee a suitable comparison. Table 10 shows six levels of GN on each image, i.e., $\sigma = 5, 10, 15, 20, 50$ and 100. The average execution time of the completion algorithms on the 150 test images is shown in Table 10. The FastNLM is most successful algorithm in relation to

Table 1. Summary of the reconstruction error results on the test images in terms of PNSR. The images to be reconstructed are listed in the table's rows, whereas each column names an image used for constructing the dictionary

Images	Img1	Img2	Img3	Lena	Living room	Cameraman	Meloid. kralli	Meloid. duyts	Meloid. pore
Img1	**145.54**	142.02	147.82	140.25	145.02	140.59	145.97	145.84	139.96
Img2	140.13	**144.92**	150.3	143.64	142.97	147.83	143.44	140.97	141.56
Img3	143.32	142.54	**153.49**	141.99	140.85	142.95	149.37	146.09	147.67
Lena	138.09	141.03	144.19	**146.51**	147.95	144.87	145.14	149.01	149.28
Living room	139.63	141.31	141.77	144.39	**149.09**	148.6	142.62	152.13	150.22
Cameraman	142.31	142.54	143.82	143.98	145.53	**152.61**	143.65	138.45	147.82
Meloid. kralli	138.69	143.98	151.87	138.94	144.98	139.84	**154.87**	149.33	140.59
Meloid. duyts	142.07	140.76	148.69	141.47	148.52	145.26	149.56	**153.76**	148.87
Meloid. pore	144.89	140.78	146.67	145.79	147.64	150.74	147.94	150.5	**150.98**

The bold values denote the dictionary trained on itself

Table 2. Shows pseudo-code of the proposed FQPSO-MP algorithm based on the MP algorithm with two major improvements to the original QPSO algorithm

The proposed (**FQPSO-MP**) algorithm.
Input: Noisy image y and the iteration number N.
-   R : Regroup Period (R=3)
-   m : Population size of each group (m=5)
-   n : Number of groups (n=5)
Initialization Stage:
(a) Set the initial estimate as $y^{(0)} = y^*$.
(b) Divide a random initial population into groups (DMS).
(c) Use TLS model to generate pre-learned dictionary D_1 using Eq. (5).
(d) Use DMS-QPSO to select the best atom of pre-learned dictionary D.
(e) Use MP to find the sparse representation of an image.
Iterative Denoising Stage:
Outer loop $l = 1,2, \dots, L$.
(f) Use the pre-learned dictionary D for sparse coding throughout the iterative denoising process.
Inner loop: iterate on $j = 1,2, \dots, J$.
(g) Calculate weighting coefficient $W_{j,q}$.
(h) $y^{(j+1/2)} = y^j + \delta(y - y^j)$, where δ is a constant.
(i) Sparsely code $y^{(j+1/2)}$ with respect to D to obtain $\alpha^{(j+1/2)}$.
(j) According to equation $\gamma_j = \sum_{q \in Q_j} W_{j,q}\alpha_{j,q}$ calculate $\gamma^{(j+1/2)}$ from $\alpha^{(j+1/2)}$. Where $\alpha_{j,q}$ is the sparse patch within set Q_j, $W_{j,q}$ is corresponding weight.
(k) According to equation $\alpha_i^{j+1} = S_\vartheta(\alpha_i^j - \gamma_i^j) + \gamma_i^j$ calculate $\alpha^{(j+1)}$. Where $S_\vartheta(.)$ is the classic soft-thresholding operator, and ϑ is the soft threshold parameter.
(l) According to equation $x \approx DO\alpha_x = (\sum_{i=1}^N R_i^T R_i)^{-1} \sum_{i=1}^N R_i^T D\alpha_{x,i}$ compute intermediate estimate $X^{(j+1)} = DO\alpha^{(j+1)}$ where α_x denotes the concatenation of all $\alpha_{x,i}$.
End of inner loop
End of outer loop

the execution time it is also the worst noise reduction can be found From the Table 10. The proposed FQPSO-MP algorithm achieves the best noise reduction. The running time of the WNNM is the worst-case. The other algorithms achieve a certain balance between running time and visual quality.

Due to his ability to use block similarity and sparse representation, the efficiency of BM3D is popularly referred to as a benchmark algorithm. Because BM3D uses set square blocks with a fixed scale and square shape across the image, its performance is limited, notably in particular on edges

with a strong contrast. The BM3D-SAPCA variety adopts a neighborhood adaptive shape approach and improves visual efficiency and increases execution time. Compared to BM3D-SAPCA, the proposed FQPSO-MP approach reduces the execution time by two-eight factors because of using the pre-learning dictionary that saves the time of calculating the learning dictionary, FQPSO-MP proposed is slightly greater than the visual performance BM3D-SAPCA because of the proposed FQPSO-MP approach uses benefit of the SRs and meta-heuristic algorithms (DMS with QPSO algorithm).

Table 3. PSNR (in decibel) comparison of the proposed and the QPSO-MP algorithms under different noise levels from 10 to 90. The bold values denote the highest PSNR values

Images	$\sigma = 10$		$\sigma = 30$		$\sigma = 50$		$\sigma = 70$		$\sigma = 90$	
	FQPSO-MP	QPSO-MP	FQPSO-MP	QPSO-MP	FQPSO-MP	QPSO-MP	FQPSO-MP	QPSO-MP	FQPSO-MP	QPSO-MP
Img1	**37.9**	34.99	**32.31**	29.59	**29.68**	26.16	**27.88**	23.71	**26.37**	21.88
Img2	**36.26**	35.2	**31.22**	29.33	**28.59**	26.04	**26.63**	23.58	**25.24**	21.75
Img3	**36.55**	34.38	**31.02**	28.95	**28.38**	25.68	**26.41**	23.34	**24.86**	21.57
Lena	**35.4**	32.7	**30.61**	28.68	**28.01**	25.69	**26.4**	23.42	**25.32**	21.72
Living room	**33.26**	28.82	**27.78**	26.25	**25.2**	24.13	**23.81**	22.37	**22.85**	20.87
Cameraman	**36.72**	34.42	**30.77**	29.16	**28.5**	25.95	**26.77**	23.46	**25.31**	21.62
Meloid. Kralli	**30.33**	21.64	**23.77**	21.23	**22.20**	20.57	**21.54**	19.73	**20.88**	18.88
Meloid. Duyts	**29.78**	21.92	**23.42**	21.48	**22.01**	20.77	**21.41**	19.94	**20.93**	19.08
Meloid. Pore	**29.56**	22.86	**23.50**	22.26	**22.29**	21.41	**21.45**	20.40	**20.73**	19.43
Average	**33.97**	29.66	**28.27**	26.33	**26.10**	24.04	**24.70**	22.22	**23.61**	20.76

Table 4. Execution time (in seconds) comparison of the proposed and the QPSO-MP algorithms under different noise levels from 10 to 90. The bold values denote the shortest execution time

Images	$\sigma = 10$		$\sigma = 30$		$\sigma = 50$		$\sigma = 70$		$\sigma = 90$	
	FQPSO-MP	QPSO-MP	FQPSO-MP	QPSO-MP	FQPSO-MP	QPSO-MP	FQPSO-MP	QPSO-MP	FQPSO-MP	QPSO-MP
Img1	**273**	1052.2	**221**	991.6	**224**	1096.1	**225**	940.8	**231**	917.2
Img2	**229**	452.3	**229**	423.4	**223**	387.1	**228**	393.6	**253**	468.7
Img3	**229**	402.1	**210**	387.8	**212**	379.7	**195**	425.4	**147**	374.1
Lena	**196.4**	403.4	**181.9**	377.9	**182.2**	387.1	**183.4**	377	**184.3**	376.2
Living room	**193.9**	955.1	**190.8**	955	**186.5**	224.13	**200.8**	906.2	**197.1**	400.6
Cameraman	**189.8**	977.3	**185.6**	973	**181.2**	1146	**183.4**	1120.8	**184.3**	976
Meloid. Kralli	**128.9**	246.9	**121.7**	238	**114.5**	278.7	**124.8**	279.8	**113.4**	299.5
Meloid. Duyts	**114.1**	374.7	**114.4**	297.5	**120**	312.2	**138.8**	233	**111.4**	309.9
Meloid. Pore	**110.7**	291	**110**	555.8	**111.7**	325.3	**117.9**	237.1	**116.3**	241.5
Average	**184.98**	572.78	**173.82**	577.78	**172.79**	504.04	**177.46**	545.97	**170.87**	484.86

Table 5. PSNR (dB) results of different denoising algorithms on test images (noise level $\sigma = 10$). The bold values denote the highest PSNR values

Images	K-SVD	Median	Bayes	TV	Bilateral	Wiener	FQPSO-MP
Img1	37.68	32.26	34.13	35.09	29.08	35.17	**37.9**
Img2	36.27	32.24	33.15	34.02	28.8	34.51	**36.29**
Img3	36.81	31.36	32.95	34.33	29.23	34.38	**36.95**
Lena	35.03	30.58	32.79	34.39	28.13	34.26	**35.4**
Living room	32.94	29.27	30.31	31.97	28.16	33.17	**33.26**
Cameraman	36.03	31.06	32.61	33.71	28.29	32.17	**36.72**
Meloid. kralli	30.24	27.29	28.71	29.09	28.06	29.16	**30.33**
Meloid. duyts	28.53	26.91	28.07	29.18	28.15	29.13	**29.78**
Meloid. pore	28.32	26.73	28.19	29.32	28.19	29.15	**29.56**
Average	33.54	29.74	31.21	32.34	28.45	32.34	**34.02**

Table 6. PSNR (dB) results of different denoising algorithms on test images (noise level $\sigma = 30$). The bold values denote the highest PSNR values

Images	K-SVD	Median	Bayes	TV	Bilateral	Wiener	FQPSO-MP
Img1	31.33	25.08	28.07	29.92	20.08	29.64	**32.31**
Img2	29.87	23.87	26.71	25.86	19.76	24.63	**31.22**
Img3	29.77	22.02	24.18	25.08	19.94	24.62	**31.02**
Lena	29.24	22.69	26.03	26.97	18.70	24.59	**30.61**
Living room	26.58	23.31	24.52	25.33	18.73	24.61	**27.78**
Cameraman	29.97	23.82	25.03	26.93	19.02	24.63	**30.77**
Meloid. kralli	23.28	22.98	24.29	25.87	18.74	23.61	**23.77**
Meloid. duyts	22.81	19.74	20.83	21.98	18.71	22.98	**23.42**
Meloid. pore	23.21	23.06	24.09	25.35	18.72	23.23	**23.50**
Average	27.34	22.95	24.86	25.92	19.16	24.73	**28.27**

Table 7. PSNR (dB) results of different denoising algorithms on test images (noise level $\sigma = 90$). The bold values denote the highest PSNR values

Images	K-SVD	Median	Bayes	TV	Bilateral	Wiener	FQPSO-MP
Img1	24.86	16.43	20.31	18.39	11.48	15.06	**26.37**
Img2	22.96	16.93	19.71	18.06	11.35	15.09	**25.24**
Img3	22.33	17.03	20.29	19.85	11.45	14.21	**24.86**
Lena	23.31	14.92	17.31	15.09	10.71	14.08	**25.32**
Living room	21.65	14.75	19.24	16.83	10.64	14.06	**22.85**
Cameraman	23.75	15.95	19.41	17.76	10.85	13.91	**25.31**
Meloid. kralli	19.99	16.47	19.82	18.86	10.82	15.08	**20.88**
Meloid. duyts	20.77	14.92	17.82	16.61	10.76	15.07	**20.93**
Meloid. pore	20.06	15.47	19.26	16.41	11.63	15.27	**20.31**
Average	22.19	15.87	19.24	17.54	11.08	14.65	**23.56**

Table 8. Average SSIM comparison between the proposed and state of the art approaches on a 150-image dataset under different noise levels from 10 to 90. The bold values denote the highest SSIM values

σ	10	30	50	70	90
K-SVD	0.9548	0.9401	0.8843	0.7059	0.6582
Median	0.9034	0.8248	0.7352	0.6074	0.5233
Bayes	0.9463	0.9261	0.8906	0.6902	0.6249
TV	0.9492	0.9398	0.8868	0.6976	0.6392
Bilateral	0.8743	0.8051	0.6357	0.5543	0.4767
Wiener	0.9283	0.8856	0.8195	0.7245	0.6107
QPSO-MP	0.9582	0.9391	0.8637	0.6993	0.6674
FQPSO-MP	**0.9746**	**0.9482**	**0.9019**	**0.7286**	**0.6851**

Table 9. Average PSNR comparison between the proposed and other state of the art approaches on a 150-image dataset under different noise levels from 10 to 90. The bold values denote the highest PSNR values

σ	10	30	50	70	90
BM3D	33.29	29.73	27.49	25.39	23.41
LSSC	33.18	29.78	27.31	25.04	23.07
NCSR	33.22	29.30	27.24	24.96	22.85
WNNM	33.73	30.23	27.97	25.82	23.99
EPID	33.79	30.36	28.13	25.97	24.02
SNLM	28.66	26.72	25.04	23.83	22.19
BM3D-SAPCA	33.81	30.32	27.99	25.95	22.82
FastNLM	29.74	26.84	25.58	24.03	22.53
FNSCR	33.46	30.19	27.67	25.91	24.19
FQPSO-MP	**34.92**	**30.41**	**28.15**	**26.11**	**24.27**

Table 10. Average running time (in seconds) comparison of the competing algorithms on a 150-image dataset under different noise levels from 5 to 100

σ	5	10	15	20	50	100
SNLM [32]	80.42	80.07	78.58	82.34	81.02	82.14
BM3D [5]	3.67	3.72	3.84	3.87	4.03	4.20
BM3D-SAPCA [33]	206.31	188.04	181.10	176.27	183.52	201.33
FastNLM [34]	0.84	0.81	0.83	0.79	0.80	0.79
WNNM [16]	271.06	278.93	270.65	274.31	396.57	613.02
K-SVD [7]	93.62	137.61	144.18	164.82.	183.38	372.19
FNSCR [35]	23.81	35.12	33.74	45.86	79.06	86.41
FQPSO-MP	17.94	28.61	27.03	38.52	69.73	74.07

Table 11. The decline in PSNR (dB) and SSIM on test images with different noise levels from 5 to 100

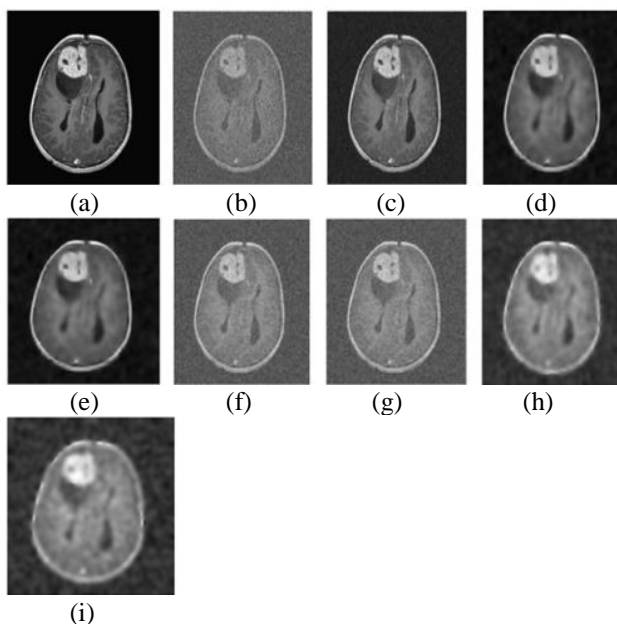| Algorithms	Original (QPSO-MP)				Proposed (FQPSO-MP)			
σ	5	30	50	100	5	30	50	100
Img1	0.21	0.26	0.19	0.32	0.0217	0.0354	0.0350	0.1401
Img2	0.47	0.42	0.28	0.35	0.0218	0.1617	0.1602	0.1850
Img3	0.53	0.40	0.23	0.19	0.0218	0.0647	0.0634	0.1793
sLena	0.30	0.25	0.22	0.18	0.0326	0.1096	0.1098	0.0103
Living room	0.15	0.88	0.58	0.35	-0.0263	0.1099	0.1082	0.05932
Cameraman	0.21	0.15	0.22	0.22	0.0313	0.0419	0.0231	0.0082

Figure. 11 Denoised results on the Img1 image with moderate noise corruption (σ = 70) using the proposed and state-of-art approaches: (a) original image, (b) corrupted image, (c) denoised image using the proposed algorithm, PSNR= 27.88, (d) denoised image using the K-SVD algorithm, PSNR=26.47, (e) denoised image using the TV algorithm, PSNR= 20.01, (f) denoised image using the Bayes filter, PSNR=21.92, (g) denoised image using the Median filter, PSNR=18.75, (h) denoised image using the Wiener filter, PSNR=17.26, and (i) denoised image using the Bilateral filter, PSNR= 13.31

BM3D's output is satisfying due to its ability to use block resemblance and SR as a popular benchmark. Since BM3D utilizes blocks with a fixed square shape and a fixed scale throughout an image, its efficiency is restricted when dealing with edges, particularly for edges with strong contrast. Its new version, i.e. BM3D-SAPCA that uses an adaptive neighborhood approach, improves visual quality performance and significantly increases running time. The proposed FQPSO-MP algorithm, in comparison with BM3D-SAPCA and K-SVD, is better in terms of quality and running time than BM3D-SAPCA and K-SVD. In addition, the running time of the proposed FQPSO-MP algorithm is better and faster than FNCSR algorithm. The proposed FQPSO-MP algorithm enhances visual performance efficiently at an appropriate time. In general, the outcomes of Tables 9 and 10 show that the proposed FQPSO-MP algorithm provides superior outcomes in noise reduction and in terms of running time compared to current competitive image algorithms.

3. Conclusion

This paper presents a pre-learned dictionary using the TLS model. This dictionary is obtained by the high-quality images rather than by the corrupted image. Furthermore, DMS is used to increase the image's performance. These changes contribute to significant computing effectiveness advantages (performance gains of approximately 90% are

achievable) without sacrificing too much image quality compared to the original QPSO-MP method (the larger decrease is less than 0.58 dB and 0.019 compared with the PSNR indexes). These improvements have significant advantages over their performance (PSNR and SSIM) and the execution time compared to the current QPSO-MP algorithm. Comprehensive experimental outcomes also show that the efficiency of the proposed algorithm in visual qualitative assessment and execution time measurements for different images and noise levels are inferior to other competing algorithms. The final performance results of the original QPSO-MP algorithm is not susceptible to image components used for dictionary construction. For this reason, the pre-learned dictionary approach applied to speed up the original QPSO-MP algorithm. The experimental results demonstrate that the FQPSO-MP algorithm's performance goes beyond the original and the state-of-art algorithms. The proposed FQPSO-MP algorithm is less running time than the original, the K-SVD, Bayes, TV, LSSC, NCSR, WNNM, EPID, median, SNLM, BM3D-SAPCA, and FNCSR approaches. The proposed FQPSO-MP algorithm achieves an important role in a certain balance between the performance of the denoised images and its execution time. In brief, two improvements have been made to this paper. One is the pre-learned dictionary, which uses the TLS model to decrease the running time relative to the original algorithm rather than learning dictionary. Where, the total running time of the proposed FQPSO-MP algorithm increases by 1.4 % when the size of the 512×512 test image increases by 15%. The second is the DMS, which improves the best atom search for images and increases their efficiency of the denoised images. In future, we will improve our proposed FQPSO-MP approach to be suitable for more types of images such as X-ray cardiovascular angiogram and remote sensing images.

Acknowledgement

References

[1] E. Elsayed, D. Salem and M. Aly, "A Fast Image Denoising Algorithm Based on TLS model and Sparse Representation", *International Journal of Imaging and Robotics*, Vol. 19, No. 2, 2019.

[2] W. Y, B. Tracey, P. Natarajan, and J. Noonan, "James–Stein type center pixel weights for non-local means image denoising", *IEEE Signal Processing Letters*, Vol. 20, No. 4, pp. 411-414, 2013.

[3] D. Shin, R. Park, S. Yang, and J. Jung, "Block-based noise estimation using adaptive Gaussian filtering", *IEEE Transactions on Consumer Electronics,* Vol. 51, No. 1, pp. 218-226, 2005.

[4] S. Li, "Non-negative sparse coding shrinkage for image denoising using normal inverse Gaussian density model", *Image and Vision Computing*, Vol. 26, No. 8, pp. 1137-1147, 2008.

[5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering", *Image Processing, IEEE Transactions*, Vol. 16, No. 8, pp. 2080-2095, 2007.

[6] S. Zhang, Q. He, K. Ouyang, and W. Xiong, "Multi-bearing weak defect detection for wayside acoustic diagnosis based on a time-varying spatial filtering rearrangement", *Mechanical Systems and Signal Processing*, Vol. 100, pp. 224-241, 2018.

[7] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD and its non-negative variant for dictionary design", *International Society for Optics and Photonics Wavelets XI*, Vol. 5914, 2005.

[8] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration", *ICCV*, Vol. 29, 2009.

[9] M. Zhang, D. Christian, and Z. Caiming, "Atlas-based reconstruction of high performance brain MR data", *Pattern Recognition*, Vol. 76, pp. 549-559, 2018.

[10] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng, "Patch group based nonlocal self-similarity prior learning for image denoising", *In: Proc. of International Conference on Computer Vision*, 2015.

[11] F. Guo, Z. Caiming, and Z. Mingli, "Edge-preserving image denoising", *IET Image Processing,* Vol. 12, No. 8, pp. 1394-1401, 2018.

[12] M. Zhang and D. Christian, "Image denoising based on sparse representation and gradient histogram", *IET Image Processing,* Vol. 11, No. 1, pp. 54-63, 2016.

[13] Z. Huang, Q. Li, T. Zhang, N. Sang, and H. Hong, "Iterative weighted sparse representation for X-ray cardiovascular angiogram image denoising over learned dictionary", *IET Image Processing*, Vol. 12, No. 2, pp. 254-261, 2017.

[14] Z. Huang, Y. Zhang, Q. Li, T. Zhang, and N. Sang, "Spatially adaptive denoising for X-ray cardiovascular angiogram images", *Biomedical Signal Processing and Control*, Vol. 40, pp. 131-139, 2018.

[15] Z. Huang, Y. Zhang, Q. Li, T. Zhang, N. Sang, and H. Hong, "Progressive dual-domain filter for enhancing and denoising optical remote-sensing images", *IEEE Geoscience and Remote Sensing Letters*, Vol. 15, No. 5, pp. 759-763, 2018.

[16] X. Liu, X. Jing, G. Tang, F. Wu, and Q. Ge, "Image denoising using weighted nuclear norm minimization with multiple strategies", *Signal Processing*, Vol. 135, pp. 239-252, 2017.

[17] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration", *IEEE Transactions on Image Processing*, Vol. 22, No. 4, pp. 1620-1630, 2012.

[18] R. Dhannawat and B. Archana, "A New Dictionary Learning Approach Using SVD with Kmeans and its appliance as regularization method for Image Deblurring using Non-Locally Centralized Sparse Representation", *International Journal of Applied Engineering Research*, Vol. 14, No.7, pp. 1672-1682, 2019.

[19] L. Singh and J. Rekhram, "Image Denoising Techniques: A Brief Survey", *Harmony Search and Nature Inspired Optimization Algorithms Springer*, pp. 731-740, 2019.

[20] R. Rubinstein, Z. Michael, and M. Elad, "*Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit*", No. CS Technion report CS-2008-08 Computer Science Department, Technion, 2008.

[21] C. Chen, J. Liang, B. Qu, and B. Niu, "Using dynamic multi-swarm particle swarm optimizer to improve the image sparse decomposition based on matching pursuit", In: *Proc. of International Conference on Intelligent Computing Springer*, 2013.

[22] P. Jia, D. Shukai, and Y. Jia, "An enhanced quantum-behaved particle swarm optimization based on a novel computing way of local attractor", *Information*, Vol. 6, No. 4, pp. 633-649, 2015.

[23] P. Afonine, P. Adams, and A. Urzhumtsev, "From deep TLS validation to ensembles of atomic models built from elemental motions. II. Analysis of TLS refinement results by explicit interpretation", *Acta Crystallographica Section D: Structural Biology,* Vol. 74, No. 7, pp. 621-631, 2018.

[24] Z. Yang and A. Wu, "A non-revisiting quantum-behaved particle swarm optimization based multilevel thresholding for image segmentation", *Neural Computing and Applications*, pp. 1-21, 2019.

[25] A. Das, and S. Aji, "A Fast and Efficient Method for Image Splicing Localization Using BM3D Noise Estimation", *Integrated Intelligent Computing, Communication and Security Springer*, pp. 643-650, 2019.

[26] Neuro Imaging and Surgical Technologies Lab. (Accessed on 30 March 2018). Available: http://nist.mni.mcgill.ca/?page_id=672.

[27] Wageningen university and research. (Accessed on 23 February 2018). Available: https://www.wur.nl/en/Research-Results/Chair-groups/Plant-Sciences/Laboratory-of-Nematology/Nematode-in-the-picture/Pictures.html

[28] W. Liu, J. Liu, L. Huang, Q. Du, and Y. Wang, "Performance analysis of reduced-dimension subspace signal filtering and detection in sample-starved environment", *Journal of the Franklin Institute*, Vol. 356, No. 1, pp. 629-653, 2019.

[29] M. Burger, Y. Korolev, C. Schönlieb, and S. Stollenwerk, "A total variation based regularizer promoting piecewise-Lipschitz reconstructions", *arXiv preprint,* Vol. 1903, No. 05079, 2019.

[30] R. Gonzalez, and R. Woods, "Digital image processing [M]", *Publishing House of Electronics Industry,* Vol. 141, No. 7, 2002.

[31] M. Salloum, K. Johnson, J. Bishop, J. Aytac, D. Dagel, and B. Van, "Adaptive wavelet compression of large additive manufacturing experimental and simulation datasets", *Computational Mechanics*, Vol. 63, No. 3, pp. 491-510, 2019.

[32] X. Li, H. He, R. Wang, and J. Cheng, "Superpixel-guided nonlocal means for image denoising and super-resolution", *Signal Processing*, Vol. 124, pp. 173-183, 2016.

[33] V. Katkovnik, A. Foi, K. Egiazarian, and J. Astola, "From local kernel to nonlocal multiple-model image denoising", *International Journal of Computer Vision*, Vol. 86, No. 1, pp. 1, 2010.

[34] O. Seynnes and N. Cronin, "Simple Muscle Architecture Analysis (SMA): an ImageJ macro tool to automate measurements in B-mode ultrasound scans", *arXiv preprint arXiv,* Vol.1905, No. 09490, 2019.

[35] S. Xu, X. Yang, and S. Jiang, "A fast nonlocally centralized sparse representation algorithm for image denoising", *Signal Processing*, Vol. 131, pp. 99-112, 2017.