An Efficient Shape Adaptive Techniques for the Digital Image Denoising

Surya Prasada Rao Borraa[a], Vishal Moyal[b], Shafqat Nabi Mughal[c], Praful Nandankard[d], Er. S. John Pimo[e] and Asadi Srinivasulu[f]

[a] Associate Professor, Dept. of ECE, Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada, suryaborra1679@gmail.com
[b] Assistant Professor, SVKMs Institute of Technology, Maharashtra, vishalmoyal@gmail.com
[c] Assistant Professor, Dept.of Electrical Engineering, Baba Ghulam Shah Badshah University, Rajouri, snmughal.bgsbu@gmail.com
[d] Assistant Professor, Electrical Engineering Dept. Government College of Engineering, pppful@gmail.com
[e] Assistant Professor, Department of CSE, St. Xavier's Catholic College of Engineering, Tamil Nadu, johnpimo@sxcce.edu.in
[f] Head Research, Professor of IT, Blue Crest University, Monrovia, Liberia, head.research@bluecrestcollege.com

Article History Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: The existing state-of-the-art in image denoising is reflected by patch-based approaches such as Block Matching and 3D collaborative filtering (BM3D) algorithms. BM3D, however, still suffers from performance degradation in smooth areas as well as loss of image information, especially in the presence of high noise levels. Integrating shape adaptive techniques with BM3D increases the denoising effect, including the denoted image's visual quality; and retains image information as well. We proposed a system in this study that generates multiple images using different shapes. These images were aggregated for both stages in BM3D at the pixel or patch levels and, when properly aggregated, resulted in an average of 1.15 dB better denoising output than BM3D.

Keywords: Digital Image Processing, Image Restoration, Additive White Gaussian Noise, Image Denoising**.**

1. Introduction

The field of digital image processing began to develop when it was affiliated with the newspaper industry in the early 1920s. The quality of digital images is one of the biggest issues in this area. Figure (1) shows the first image sent between New York and London, UK, via a submerged cable in the Atlantic Ocean. There is a disturbance in all of the image's grey intensity levels. While the processing of digital images has advanced dramatically since this picture, it still faces many challenges today.

Figure 1: Digital Image produced in 1921 [1]

Either during image processing or transmission or both, the primary source of noise in digital images appears. Relay for image acquisition on imaging sensors sensitive to bright light. The function of the light receptors located within the human eye inside the retina membrane is the same. For example, Charge-coupled (CCD) imaging equipment, which is a type of camera sensor, contains a sensor presented as a 2-D array of several million small solar cells. In this grid, every cell corresponds to a pixel in the digital image. The light from the object is reflected in the grid of the sensor, enabling the solar cells to estimate the projected number of photons in each cell. Finally, each of these calculated charges is converted into a digital pixel value by the analog-to-digital converter (ADC). In more detail, a high photon estimated value, which is converted to a high intensity value, is provided by the cell in the sensor subjected to more light. The pixel will have a value similar to 255 (white) in grey scale images, and vice versa. Logically, to perfectly fit the scene, increasing the number of cells in the sensor improves the resolution and accuracy of the information in the image. However, because of many factors, this is not a hundred percent true.

On the one side, a number of variables, such as light levels and sensor efficiency, are involved in the image acquisition environment. These factors influence the calculation of the final pixel values in the images, causing the appearance of noise and artefacts in the image. On the other hand, image transmission through wired or wireless networks, due to atmospheric conditions or usually flows in the medium of transformation, induces variations in the real pixel values.

Noise is split into two types: dependent and autonomous noise. Like multiplicative noise, based noise de- pends on the pixel value. Although independent noise, including white Gaussian noise, Rayleigh noise, erlang (gamma) noise, exponential noise, and salt-and - pepper noise, is not associated with the pixel strength. By generating an array that is close in size to the image given, different noise models are simulated. Except for salt and pepper noise, the strength values are stochastic numbers with a particular probability density function.

Dealing with noisy images is a much harder issue in real-world applications. Next, the captured images typically involve several models of noise. Images obtained from satellite imagery, for example, include speckle noise, Gaussian noise, and impulse noise. Secondly, it needs prior knowledge of the noise model(s) present in the picture to choose the best denoising method. Finally, assuming that the noise model is understood, several of the existing methods of denoising cause either loss, blurring, disruption in smooth areas or ringing artefacts around the edges in some of the image information.

One of the most common forms of noise in images is additive white Gaussian noise (AWGN). AWGN is distributed over images randomly. A state-of - art patch-based algorithm is block matching and 3D collaborative filtering (BM3D). BM3D is an algorithm of two stages. Block matching is applied between neighbouring patches in both phases. The matching method does not always provide good results due to the current noise. Consequently, at high noise levels, BM3D still suffers from degradation in denoising efficiency. In smooth areas and data loss in edge and texture areas, the deterioration occurs as a disturbance.

2. Background

The additive white Gaussian noise is clarified in this section. Then, a brief description of methods for assessing image quality. Finally, some of the strategies suggested by studies are discussed and contrasted to demonstrate the increase in image denoising efficiency.

2.1 Additive White Gaussian Noise

Because of changes in the values of the original signal, Gaussian noise happens in pictures. White noise is what it is called. The name was derived from the fact that this noise model's Fourier spectrum is constant. In other words, it has the physical features of white light, and is made up of an equal proportion of the colour spectrum. In grey images, AWGN is simulated as a disruption of the image pixel's grey values caused by adding random values to the pixel values, as seen in Equation (1), of the

$$I_{nse} = I + \sigma \times rand\ (m \times n) \quad (1)$$

Inse is the image of the AWGN, σ is the standard deviation, m x n represents the image size. Rand is a function that produces distributed normal (Gaussian) variables that can be defined by the function of the following probability density function (pdf). Fig. (2), displays the noise and the related PDF curve. of the

$$f(x)=1/2\pi\sigma^2\ exp^{-(x-\mu)^2}\ /\ 2\sigma^2 \quad (2)$$

Where x in the image corresponds to a pixel, μ is the sum of all pixels, then σ2 is the variance. The mean is equal to zero in the case of Gaussian noise. The Due to the increase in standard deviation, the x-axis reflects the increase in the range.

2.2 Image Denoising Methods

Image denoising has applications of immense importance. Medical image denoising [2][3][4] and remote sensing image denoising [5], for example. In addition to denoising, for classification, segmentation, edge detection, ..., etc., it is used as pre-processing.

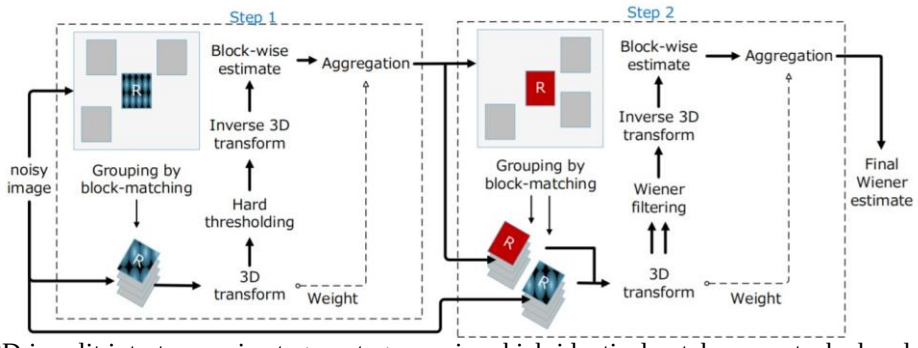
Image denoising techniques are divided into two groups in this section: spatial domain, and methods that combine denoising of spatial and transform (hybrid) domain. Later, some of the approaches suggested to enhance the efficiency of denoising are discussed. Low-level noise applies to noisy sigma images = 10, 20, and 30 in this analysis. Although high-level noise refers to sigma = 40, 50, 60, 70, 80, 90, and 100 noisy images.

2.3 Hybrid Domain Image Denoising

Image denoising techniques that combine both spatial domains and transform domain denoising as complements, such as the latest state-of-the-art BM3D, have proven to be effective denoising methods. Good quality images are generated by the spatial domain methods, but many of the small details are smoothed during the denoising process. The transform domain approach does not, however, provide the same high quality. This generates a ringing object around the edges. They often retain tiny details in the image's texture.

2.3.1 Block-matching and 3D filtering (BM3D)

BM3D[23] is a denoising philtre that, along with collective filtering, relies on self-similarities that the images



have. BM3D is split into two main stages: stage one in which identical patches are stacked and shrinkage of the wavelet is applied. Step two uses the phase one approximate performance and applies the Wiener philtre to 3D clustered patches.

Figure 2:BM3D filtering flowchart.

Stage One: Basic Estimation

In BM3D, the first stage is to stack matching or adjacent blocks into a 3D array so that the data sparsity is enhanced when the transform domain is applied. The matrix or vector in which the majority of its values are zero is known as sparsity. A wavelet transform is the required image processing.

Patch Grouping: Assuming ZxR denotes a noisy reference patch, x is the pixel 's spatial coordinates. Z(x) as a central patch pixel and N 2 is the size of the patch. Just in Fig. (4) In phase one of the BM3D algorithms, the first step begins with block matching using the normalised Euclidean distance calculation between Z_x blocks specified by Equation (3).

$$d(Z_{xR}, Z_x) = \frac{||\Ÿ\left(\tau_{hard}^{2D}(ZxR)\right) - \text{⍰}\left(\tau_{hard}^{2D}(ZxR)\right)||_2^2}{N_1^{hard}} \quad (3)$$

After 2D transformation, as referred to by the unitary transform operator, hard thresholding (t_{hr}) is applied to the patches identical to ZxR to reduce the noise, then hard threshold operator ⁇ is applied with λ2Dσ. To remove the less identical patches, patch grouping is performed by including patches with distances less than the threshold value. As the following Equation (4) shows:

$$\Upsilon(\lambda, \lambda_{thr}) = \begin{cases} \lambda & \text{if } |\lambda| > \lambda_{thr} \\ 0 & otherwise \end{cases} \quad (4)$$

λ_{thr} is the maximum distance between two patches that allow the patch to join the set of similar patches called Z_{S_X}.

Collaborative Hard Threshold:

Given a 3D array Z_{S_XR} with $N_1 \times N_1 \times |S_xR|$ dimension. 3D transform consists of 2D transform over the patches and 1D Walsh-Hadamard transform over the third dimension. The Walsh- Hadamard transform requires an even number of similar patches such that N^{hard} is always apower of two. If an even number of patches is not accessible then we decrease the number of extracted patches until a power of two can be obtained. The noise is removed using hard threshing. τ^{-1}. Finally, the denoised block is estimated as the inverse 3D transform value, as defined in Equation (5)

$$\hat{Y}_{SxR} = \tau_{3D}^{-1}\left(\Upsilon(\tau_{3D}(Z_{SxR}), \lambda_{thr3D\sigma}\sqrt{2\log(N_1^2)})\right) \quad (5)$$

Projecting from the wavelet transform domain to the spatial domain of the image is not perfect because if positioned in several positions in the original image, a single patch may be a valid solution.

Patch Aggregation:

Unlike areas that mix several distinct textures, the weighting approach gives more weight to homogeneous patches, causing edge preservation and preventing ringing artefacts around the edges. Currently, each pixel has more than one estimated average weight due to patch overlap, which is why aggregation is used to estimate the correct weight that contributes to the final pixel value. The weights of the patches in original BM3D are determined as the product of the estimated patch after inverse 3D transformation and the Kaiser window, the weight of a pixel in the original BM3D is inversely proportional to the calculation of the total block variance. In the final set of patches, weight is specified for each block using the following equation. (6).

$$w_{x_R} = \begin{cases} \frac{1}{N_{har}} & if\ N_{har} \geq 1 \\ 1 & otherwise \end{cases} \quad (6)$$

N_{har} is the number of transform coefficients with values other than zero after applying a thresh- old. Meaning that the more different the 3D block is the less this block contributes to the weight. The final estimate of the pixel yˆ is given by Equation (7)

$$\hat{y} = \frac{\sum_{x_R \epsilon X}\sum_{x_m \epsilon S_{x_R}} w_{x_R} \hat{Y}_{xm}^{xR}(x)}{\sum_{x_R \epsilon X}\sum_{x_m \epsilon S_{x_R}} w_{x_R} X_{xm}^{xR}(x)}, \quad \forall\, x \epsilon X \quad (7)$$

Stage Two: Wiener Filter:

In this step, a better-denoised approximation of the image is obtained using the basic approximation of step one and the noisy image. By applying Wiener philtre shrinkage, the final denoised image is obtained by applying the 3D transformed domain stack extracted from the basic image.

Patch Grouping:

The second stage in BM3D involves applying the Wiener filter as shown in Fig. (4). The first step is to find the matching blocks within the estimated image resulting from stage one using a hard threshold on d-distance measure defined by Equation (8).

$$S_{x_R} = x \epsilon X \mid \frac{||\left(E_{x_r} - E_{x_r}^{-}\right) - \left(E_x - E_x^{-}\right)||_2^2}{N_1^{thr}} \quad (8)$$

Ex_r and Ex are two matching with E¯x and, E¯x referring to the mean values respectively. The mean is subtracted to prevent the patch similarity searching to be biased.

3. Methodology

Block-matching and 3D filtering (BM3D) is considered as a prominent state-of-art denoising algorithm. In Fig. (5) smooth areas in BM3D suffer from distortion. While edge areas loss details (Lena's hat). BM3D still has room for improvement. In this chapter, a framework is proposed to add different kernel shapes to the original BM3D. BM3D originally uses the Kaiser (Gaussian) kernel to determine the denoised pixel value. Instead, the suggested kernel shapes shown in Fig. (6) are used.

The reason for choosing these shapes is that the edge areas in images can be quantized into four angles 0, 45, 90, and 135 degrees. Besides, using these shapes allows BM3D to preserve image details while maintaining smooth areas [25]. This framework can be used with more shapes that can improve the denoising performance. However, for this study these shapes are sufficient. The multiple kernels are used to produce various images with different performances. The combination of these images is the most crucial part of the framework.

3.2 Framework Analysis

An experiment is conducted to show the benefit of using various kernel shapes. The kernels are used to estimate the final weights of the pixels from multiple patches. The weights are based on the distances from the reference patch. The multiple kernels are applied in the aggregation step after stages one and two of the original BM3D. The output after utilizing the six shapes are images, as shown in Fig. (7) after BM3D stage one and Fig. (8) after stage two. The images show an improvement in the PSNR between the noisy image and the denoised images. However, the improvement is less than the original BM3D results in most cases (except for square kernel in some cases when the noise is low). The combination of the resulting images using either pixel-based or patch-based methods is a significant improvement in PSNR values and consequently, on the quality of the image.

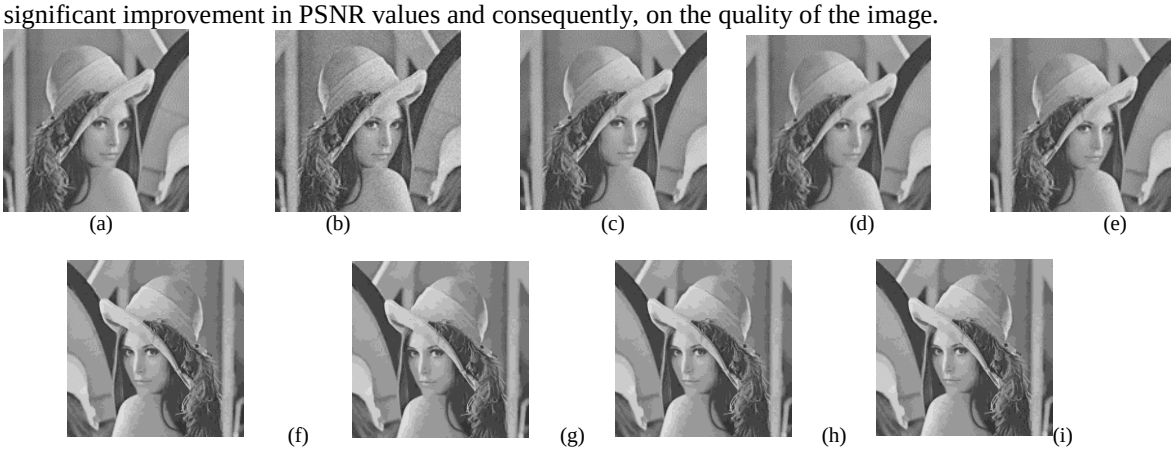


Figure 3: Lena image denoised by BM3D using different shaped kernels for stage one: (a) Original image. (b) Noisy image with sigma = 20 (PSNR = 22.10). (c) Original BM3D stage one denoised image (PSNR = 32.30). (d) square kernel (PSNR = 32.19). (e) Circular kernel (PSNR = 32.12). (f) Horizontal kernel (PSNR = 32.16). (g) Vertical kernel (PSNR = 32.15). (h) Diagonal with angle 135 kernel (PSNR = 32.20). (i) Diagonal with angle 45 kernel (PSNR=32.16).

Figure 4: Lena image denoised by BM3D using different shaped kernels for stage two: (a) Original image. (b) Noisy image with sigma = 20 (PSNR = 22.10). (c) Original BM3D stage two denoised image (PSNR = 32.83). (d) Square kernel (PSNR = 33.43). (e) Circular kernel (PSNR = 33.44). (f) Horizontal kernel (PSNR = 33.44). (g) Vertical kernel (PSNR = 33.42). (h) Diagonal with angle 135 kernel (PSNR = 33.44). (i) Diagonal with angle 45 kernel (PSNR=33.45).

3.3 Proposed Framework

It is assumed that the given image is corrupted by AWGN, i.e., the noise and image are uncorrelated. Our proposed framework replaced the square Gaussian kernel used in BM3D aggregation step by six various flat kernels. Shown in Fig. (9), Fig. (10). BM3D originally uses a window of size 8 × 8 for noise levels less than 40. Otherwise, it uses a window of size 12 × 12.

The input of the aggregation step is 3D stacked patches. These patches should be placed back in their original locations in the image. However, due to overlapping and similarities between image patches, each pixel has multiple evaluations. In original BM3D, the evaluation of a normalized sum of the pixel values (in each patch) multiplied by the kernel values is used to give final pixel estimation. as shown in Equation (9),

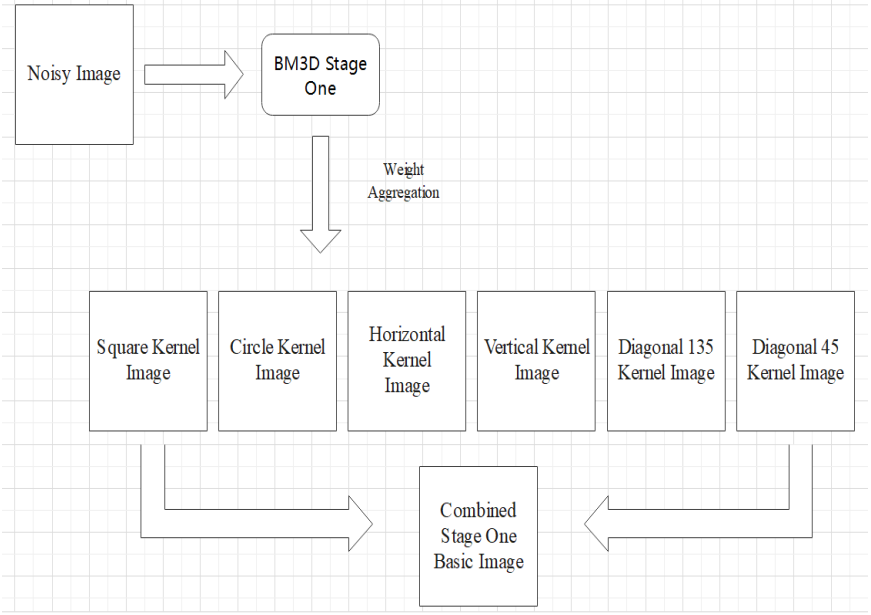


Figure 5: Stage one for the proposed framework.

Where Yˆ refers to the pixel value in patch x. ωxR refers to the weight of the pixel obtained from evaluating its location from the center pixel using the kernel in the original BM3D. The proposed framework applies to Equation (9) six times, each time with a different kernel.

$$\hat{y} = \frac{\Sigma_{x_R \epsilon X} \Sigma_{x_m \epsilon S_{x_R}} w_{x_R} \hat{Y}_{xm}^{xR}(x)}{\Sigma_{x_R \epsilon X} \Sigma_{x_m \epsilon S_{x_R}} w_{x_R} X_{xm}^{xR}(x)} \quad , \quad \forall x \epsilon X \qquad (9)$$

The flat kernel gives similar weights to all the pixels in the given patch area, thus reducing the computations. Instead of multiplying each pixel by a different value in the kernel depending on its distance from the pixel at the patch center, all the pixels in the patch are multiplied by the same value. Another reason for using a flat kernel is its ability to fit the underlying structure of the image, i.e., instead of relying on the distance from the central pixel, we rely on the shape of the extracted patch and how much it is similar to the structures surrounding it.

The most crucial step in the framework is aggregating the output images into one final denoised image. Using the original noiseless image, we combine the results. The framework does not rely on the original image or consider it as part of the algorithm, it is used for feasibility only. The original image allow use to estimate the optimal results for the framework and prove that there is still a room for improvement by using adaptive shapes. Even though, in our future work, we are looking for other approaches the can be used to aggregate the image automatically.

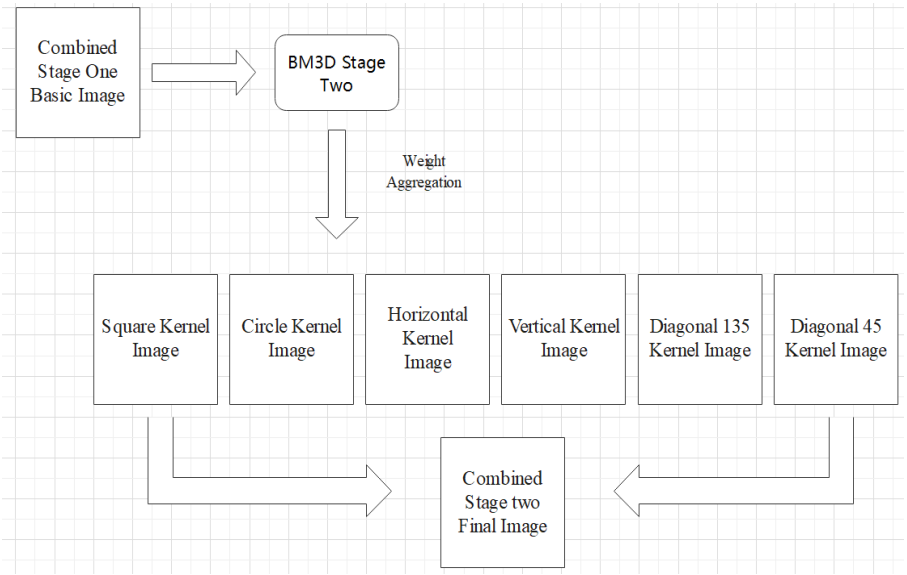
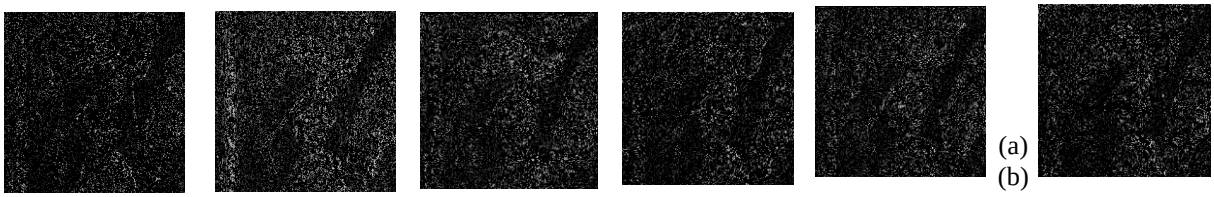


Figure 6: Stage two for the proposed framework.

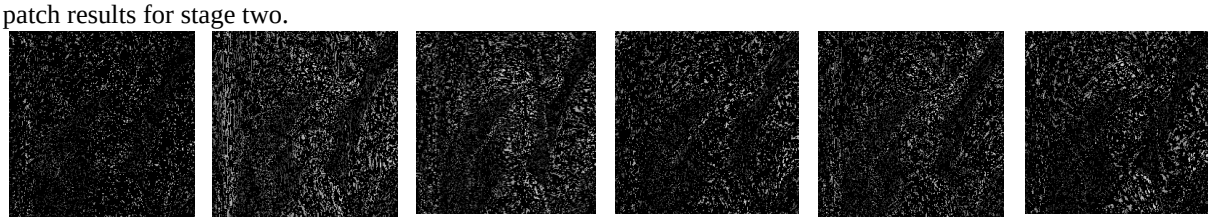
Given the six images that represent the solution space for each pixel or patch and the original image, the optimal solution is the one with the least pixel or patch error. The least error choices are shown in the following Figure (11) for stage one and Figure (12) for stage two.



(a) (b) (c) (d) (e) (f)

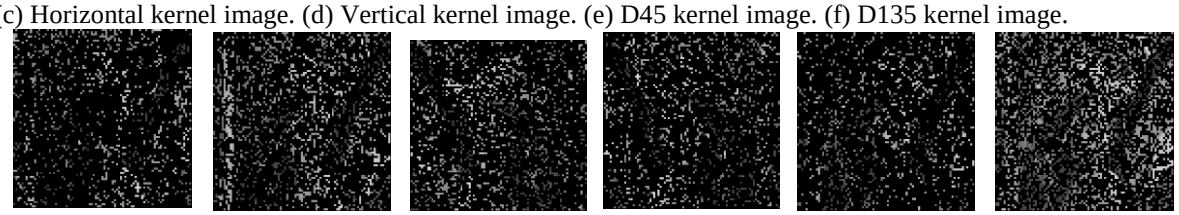
Figure 7: Lena image, after stage one, comprising six images that represent the locations of the pixels that should be taken from each image to form the optimal denoised image. (a) Square kernel image. (b) Circular kernel image. (c) Horizontal kernel image. (d) Vertical kernel image. (e) D45 kernel image. (f) D135 kernel image.

Another approach to determine the relationship between the six images is to used patches comparison to the original image patches. Extract the corresponding patches from the six images obtained from different kernel weights, then compare it to the corresponding original image patch and choosing the absolute minimum difference patch results for stage two.



(a) (b) (c) (d) (e) (f)

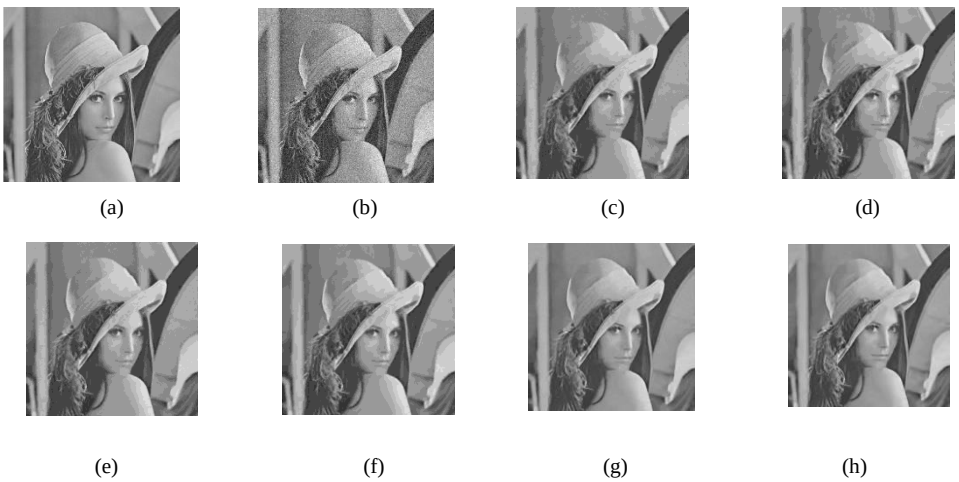
Figure 8: Lena image, after stage two, comprising six images that represent the locations of the pixels that should be taken from each image to form the optimal denoised image. (a) Square kernel image. (b) Circular kernel image. (c) Horizontal kernel image. (d) Vertical kernel image. (e) D45 kernel image. (f) D135 kernel image.



(a) (b) (c) (d) (e) (f)

Figure 9: Lena image, after stage one, comprising six images that represent the locations of the patches that should be taken from each image to form the optimal denoised image. (a) Square kernel image. (b) Circular kernel image. (c) Horizontal kernel image. (d) Vertical kernel image. (e) D45 kernel image. (f) D135 kernel image.

The patch size chosen is 3 × 3, as when the size of the patch increases the improvement declines due to the increase of the mismatching between the estimated value and the kernel. The kernel used to calculate the final pixel value is not necessarily the same kernel used for all its neighbouring pixels. Finally, the results of aggregated pixels are for stage one and stage two.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 10: Denoised Lena image results using the proposed framework. (a) Original image. (b) Noisy image with sigma = 20 (PSNR = 22.14). (c) Original BM3D after stage one. (PSNR = 32.23) (d) Original BM3D after stage two (PSNR = 33.77). Pixel based results, (e) Proposed framework after stage one (PSNR = 33.69). (f) Proposed

framework after stage two (PSNR = 33.99). Patch based results, (g) Proposed framework after stage one (PSNR = 32.74). (h) Proposed framework after stage two (PSNR = 33.26).

3 Results

In this section, we will discuss the results of the proposed framework. The performance of the framework is compared to the original BM3D after stage one and two. The dataset of images used in the evaluation is shown in Fig. (2). Finally, we will conclude the overall numerical and visual results.

3.1 Numerical Results

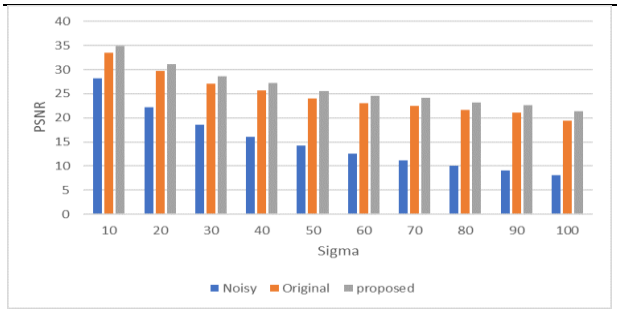
We have applied the proposed framework after both stages one and two of the BM3D scheme on multiple images for various noise levels 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100. The results include the PSNR of the noisy image, the original BM3D image after stage one and stage two, the images obtained from using the different kernels (square, circle, horizontal, vertical, diagonal 134 and diagonal 45) and the image generated from the proposed framework. For Stage one results, see tables 4.1, 4.3, 4.5, 4.7, 4.9, 4.11, 4.13, 4.15. For Stage two results, see tables 4.2, 4.4, 4.6, 4.8, 4.10, 4.12, 4.14, 4.16. The results show that the performance of the proposed framework surpasses the performance of the original BM3D scheme. This is true, although the denoising performance of each kernel is slightly less than the original BM3D and in some cases.

Table 1: Barbara image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework

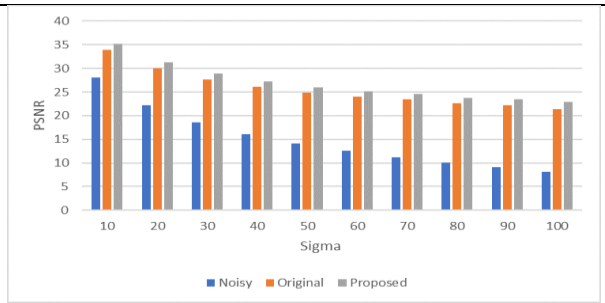
Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal 135	Diagonal 45	Proposed	Improv.
10	28.12	33.54	33.55	33.15	33.30	33.43	33.47	33.17	34.89	1.35
20	22.14	29.73	29.66	29.18	29.53	29.59	29.63	29.44	31.14	1.41
30	18.58	27.13	27.04	26.67	27.00	27.03	27.00	26.90	28.57	1.44
40	16.11	25.65	25.54	25.15	25.48	25.51	25.50	25.40	27.18	1.53
50	14.18	24.01	23.90	23.57	23.86	23.85	23.89	23.76	25.57	1.56
60	12.52	23.06	22.98	22.61	22.89	22.91	22.93	22.80	24.53	1.47
70	11.24	22.53	22.44	22.06	22.35	22.39	22.37	22.21	24.11	1.58
80	10.10	21.69	21.63	21.35	21.56	21.51	21.64	21.42	23.15	1.46
90	9.05	21.09	21.02	20.80	20.88	20.96	20.98	20.80	22.65	1.56
100	8.15	19.45	19.50	18.85	18.98	19.29	19.29	18.65	21.40	1.95

Table 2: Barbara image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two from 10 to 100 sigma noise levels.

Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal 135	Diagonal 45	Proposed	Improv.
10	28.12	33.97	34.38	34.38	34.38	34.39	34.40	34.44	35.21	1.24
20	22.14	30.01	30.52	30.49	30.49	30.51	30.51	30.52	31.32	1.31
30	18.58	27.62	28.23	28.17	28.17	28.21	28.20	28.17	28.93	1.31
40	16.11	26.09	26.65	26.63	26.64	26.65	26.65	26.64	27.16	1.07
50	14.18	24.87	25.49	25.46	25.49	25.49	25.47	25.46	25.98	1.11
60	12.52	24.00	24.63	24.60	24.62	24.63	24.62	24.60	25.11	1.11
70	11.24	23.48	24.16	24.09	24.15	24.14	24.15	24.09	24.64	1.16
80	10.10	22.61	23.25	23.20	23.24	23.22	23.26	23.19	23.74	1.13
90	9.05	22.22	22.92	22.88	22.91	22.91	22.90	22.86	23.42	1.20
100	8.15	21.32	22.38	22.31	22.36	22.38	22.34	22.33	22.96	1.64



(a)



(b)

Figure 11: Barbara image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.

Table 3: Baboon image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one from 10 to 100 sigma noise levels.

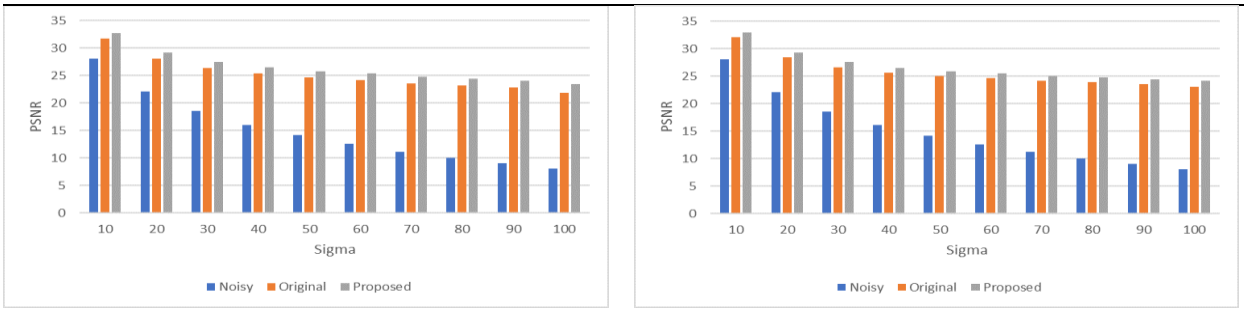
Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal 135	Diagonal 45	Proposed	Improv.

							135	45		
10	28.12	33.54	33.55	33.15	33.30	33.43	33.47	33.17	34.89	1.35
20	22.14	29.73	29.66	29.18	29.53	29.59	29.63	29.44	31.14	1.41
30	18.58	27.13	27.04	26.67	27.00	27.03	27.00	26.90	28.57	1.44
40	16.11	25.65	25.54	25.15	25.48	25.51	25.50	25.40	27.18	1.53
50	14.18	24.01	23.90	23.57	23.86	23.85	23.89	23.76	25.57	1.56
60	12.52	23.06	22.98	22.61	22.89	22.91	22.93	22.80	24.53	1.47
70	11.24	22.53	22.44	22.06	22.35	22.39	22.37	22.21	24.11	1.58
80	10.10	21.69	21.63	21.35	21.56	21.51	21.64	21.42	23.15	1.46
90	9.05	21.09	21.02	20.80	20.88	20.96	20.98	20.80	22.65	1.56
100	8.15	19.45	19.50	18.85	18.98	19.29	19.29	18.65	21.40	1.95

Table 4: Baboon image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two from 10 to 100 sigma noise levels.

Sigma Noisy Original Square Circle Horizontal Vertical Diagonal Proposed Improv.

							135	45		
10	28.10	32.06	32.53	32.52	32.51	32.53	32.53	32.50	32.98	0.92
20	22.15	28.39	28.94	28.91	28.89	28.92	28.92	28.86	29.30	0.91
30	18.56	26.65	27.20	27.16	27.16	27.19	27.18	27.13	27.55	0.90
40	16.06	25.69	26.23	26.21	26.20	26.22	26.22	26.17	26.53	0.84
50	14.18	25.04	25.55	25.53	25.53	25.54	25.54	25.50	25.86	0.82
60	12.57	24.65	25.20	25.17	25.17	25.19	25.18	25.14	25.54	0.89
70	11.20	24.16	24.71	24.68	24.67	24.68	24.71	24.65	25.06	0.90
80	10.05	23.87	24.42	24.40	24.38	24.40	24.41	24.37	24.79	0.92
90	9.05	23.51	24.06	24.03	24.04	24.05	24.05	24.03	24.45	0.94
100	8.13	23.13	23.79	23.74	23.77	23.78	23.77	23.75	24.22	1.09



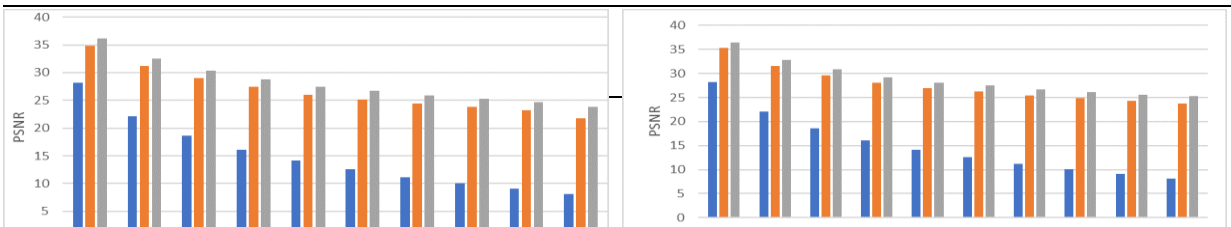
(a) (b)

Figure 12: Baboon image PSNR comparison between noisy, original BM3D and the proposed frame- work at various noise levels (Sigma).

Table 5: Boat image PSNR for noisy, original BM3D, six kernel images, the pixel proposed frame- work

Sigma Noisy Original Square Circle Horizontal Vertical Diagonal Proposed Improv.

							135	45		
10	28.16	34.79	34.76	34.56	34.56	34.73	34.69	34.54	36.10	1.31
20	22.10	31.16	31.08	30.99	31.00	31.07	31.07	31.01	32.50	1.34
30	18.60	29.03	28.93	28.85	28.88	28.90	28.95	28.86	30.39	1.36
40	16.10	27.42	27.35	27.16	27.25	27.28	27.34	27.16	28.83	1.41
50	14.14	26.05	25.99	25.75	25.84	25.94	25.94	25.71	27.48	1.43
60	12.59	25.21	25.15	24.89	25.02	25.09	25.09	24.84	26.68	1.47
70	11.20	24.46	24.42	24.16	24.25	24.34	24.37	24.11	25.93	1.47
80	10.10	23.82	23.77	23.58	23.63	23.72	23.70	23.52	25.27	1.45
90	9.06	23.18	23.14	22.97	22.91	23.07	23.08	22.84	24.64	1.46
100	8.16	21.83	21.92	21.32	21.25	21.67	21.69	20.92	23.78	1.95

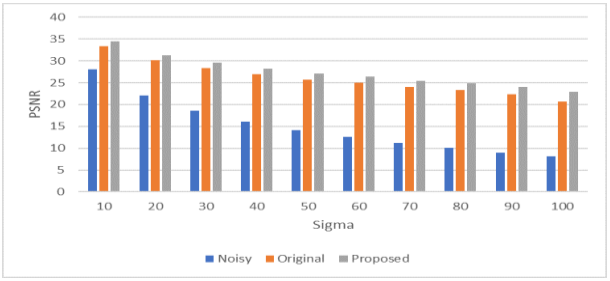


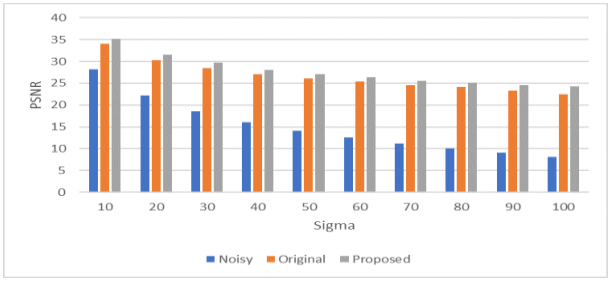
(a) (b)

Figure 13: Boat image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.

Table 6: Cameraman image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one from 10 to 100 sigma noise levels.

Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal	Proposed	Improv.
						135	45		
10	28.14	33.33	33.40	32.98	33.03	33.29	33.26	32.87	34.56	1.23
20	22.13	30.10	30.09	29.85	29.87	30.02	30.01	29.79	31.34	1.24
30	18.60	28.30	28.25	28.09	28.09	28.21	28.19	28.04	29.59	1.29
40	16.09	26.93	26.85	26.74	26.79	26.84	26.81	26.73	28.28	1.35
50	14.14	25.75	25.65	25.55	25.59	25.64	25.62	25.55	27.12	1.37
60	12.57	24.96	24.86	24.84	24.74	24.83	24.86	24.75	26.39	1.43
70	11.23	24.03	23.96	23.80	23.84	23.92	23.91	23.76	25.43	1.40
80	10.06	23.32	23.24	23.13	23.12	23.20	23.21	23.07	24.81	1.49
90	9.02	22.34	22.31	22.02	22.07	22.23	22.21	21.89	23.97	1.63
100	8.10	20.62	20.71	20.00	20.05	20.49	20.53	19.63	22.90	2.28





(a) (b)

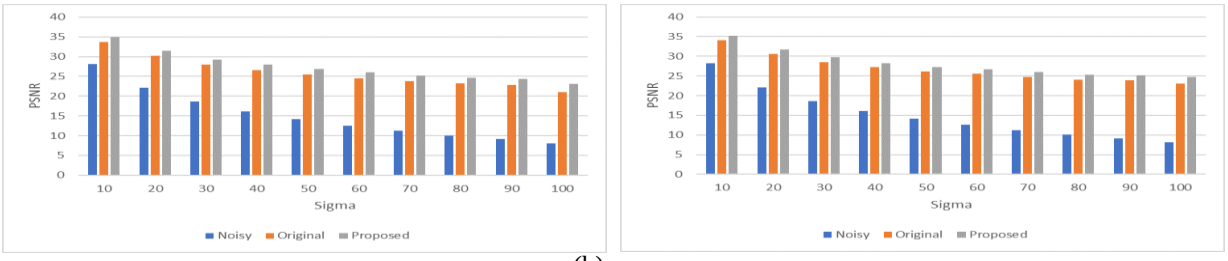
Figure 14: Cameraman image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.

Table 7: Goldhill image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one from 10 to 100 sigma noise levels.

Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal	Proposed	Improv.
						135	45		
10	28.16	33.71	33.69	33.43	33.49	33.65	33.63	33.44	34.97	1.26
20	22.08	30.17	30.11	29.94	29.99	30.08	30.07	29.93	31.49	1.32
30	18.57	27.93	27.86	27.66	27.79	27.85	27.83	27.73	29.25	1.32
40	16.07	26.54	26.45	26.33	26.38	26.43	26.44	26.38	27.97	1.43
50	14.18	25.47	25.40	25.25	25.30	25.38	25.34	25.25	26.88	1.41
60	12.57	24.56	24.50	24.30	24.41	24.41	24.46	24.27	25.99	1.43
70	11.26	23.75	23.70	23.49	23.55	23.66	23.61	23.46	25.20	1.45
80	10.04	23.24	23.20	22.97	23.06	23.15	23.11	22.92	24.70	1.46
90	9.09	22.81	22.78	22.58	22.55	22.66	22.73	22.51	24.31	1.50
100	8.08	21.00	21.06	20.48	20.43	20.85	20.80	20.15	23.09	2.09

Table 8: Goldhill image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two from 10 to 100 sigma noise levels.

Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal 135	Diagonal 45	Proposed	Improv.
10	28.16	34.11	34.59	34.57	34.57	34.60	34.57	34.56	35.18	1.07
20	22.08	30.56	31.11	31.06	31.09	31.11	31.09	31.06	31.70	1.14
30	18.57	28.50	29.12	29.06	29.10	29.13	29.09	29.08	29.69	1.19
40	16.07	27.28	27.84	27.80	27.82	27.83	27.83	27.81	28.26	0.98
50	14.18	26.20	26.81	26.75	26.79	26.78	26.79	26.75	27.23	1.03
60	12.57	25.63	26.27	26.23	26.24	26.23	26.28	26.24	26.72	1.09
70	11.26	24.77	25.47	25.40	25.46	25.43	25.47	25.44	25.93	1.16
80	10.04	24.10	24.83	24.79	24.81	24.82	24.81	24.80	25.31	1.21
90	9.09	23.93	24.70	24.65	24.64	24.65	24.70	24.63	25.17	1.24
100	8.08	23.02	24.12	24.06	24.08	24.11	24.10	24.08	24.70	1.68



(a) (b)

Figure 15: Goldhill image PSNR comparison between noisy, original BM3D and the proposed frame- work at various noise levels (Sigma). (a) After stage one. (b) After stage two.

The previously discussed results show an improvement in the PSNR between the original BM3D and the final result of the proposed framework. Despite the fact that the six different kernel images have lower PSNR in stage one. For instance, the average table (7) the square kernel, circular kernel, horizontal kernel, vertical kernel, diagonal 135, and diagonal 45, for all sigma values, are less than original BM3D stage one result by 0.04, 0.26, 0.22, 0.11, 0.11, and 0.31 respectively. Also, the square kernel in some cases gives the same or better results than original BM3D such as Cameraman, Lake images.

The framework stage one output image is the input of stage two. Consequently, stage two six kernels give higher results than stage two original BM3D. As shown in average table (8), the square kernel, circular kernel, horizontal kernel, vertical kernel, diagonal 135, and diagonal 45, for all sigma values, are more than original BM3D stage two result by 0.64, 0.61, 0.62, 0.63, 0.63, and 0.61 respectively. Finally, the combined image produced after stage two gives a significant increase with respect to the PSNR. The average improvement between the original BM3D and the proposed work for the given dataset is approximately 1.15 dB.

4 Conclusion

In this section, we will summarise the conclusion of our framework and the possible approaches that can be considered to apply the combination of various kernel images automatically. In this framework, we conduct an experiment to improve denoising performance using shape adaptive shapes. First, the results show, for low noise images (sigma = 10, 20, and 30), the average improvement of the pixel-based over the eight images is 1.18 approximately. As for high noise images (sigma = 40, 50, 60, 70, 80, 90, and 100), the average improvement of the pixel-based over the same images is 1.13 approximately. Second, the pixel-based combination approach is better than the patch-based approach. However, the patch approach has less computational requirements than the pixel approach. In the proposed approach, an expansion to the BM3D algorithm is introduced. This method outperforms the original BM3D scheme, where the minor details are preserved while maintaining the smoothness of texture-less areas. The proposed geometric kernels enable shape adaptivity to the inner image structure details.

References:

a. R. C. Gonzalez and R. E. Woods, Digital Image Processing (3rd Edition). Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006, ISBN: 013168728X.

b. Y. E. Gokdag, F. S¸ansal, and Y. D. Gokdel, “Image denoising using 2-d wavelet algorithm for gaussian-corrupted confocal microscopy images,” Biomedical Signal Processing and Control, vol. 54, p. 101 594, 2019, ISSN: 1746-8094. DOI: https://doi.org/10.1016/ j. bspc.2019.101594. [Online]. Available: http://www.sciencedirect.com/science/ article/pii/S1746809419301740.

c. H. Koyuncu and R. Ceylan, “Elimination of white gaussian noise in arterial phase CT images to bring adrenal tumours into the forefront,” Computerized Medical Imaging and Graphics, vol. 65, pp. 46–57, 2018, Advances in Biomedical Image Processing, ISSN: 0895- 6111. DOI:

https://doi.org/10.1016/j.compmedimag.2017.05.004. [Online]. Avail- able: http://www.sciencedirect.com/science/article/pii/S0895611117300472.

d. H. V. Bhujle and B. H. Vadavadagi, "Nlm based magnetic resonance image denoising–a review," Biomedical Signal Processing and Control, vol. 47, pp. 252–261, 2019.

e. X. J. He, Y. Q. Wang, Y. Li, and A. G. Xu, "A novel local variance-based filtering method for denoising remote sensing images," Remote Sensing Letters, vol. 10, no. 8, pp. 736– 745, 2019. DOI: 10.1080/2150704X.2019.1606469. eprint: https://doi.org/10.1080/ 2150704X.2019.1606469. [Online]. Available: https://doi.org/10.1080/2150704X. 2019.1606469.

f. C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), Jan. 1998, pp. 839–846. DOI: 10.1109/ICCV.1998.710815.

g. B. Goyal, A. Dogra, S. Agrawal, and B. S. Sohi, "A three stage integrated denoising approach for grey scale images," Journal of Ambient Intelligence and Humanized Computing, pp. 1–16, 2018. DOI: 10.1007/s12652-018-1019-5. [Online]. Available: https: //app.dimensions.ai/details/publication/pub.1106706026.

A. Buades, B. Coll, and J. M. Morel, "On image denoising methods" CMLA Preprint, vol. 5, 2004.

h. A. Buades, B. Coll, and J.-M. Morel, "Non-local means denoising," Image Processing On Line, vol. 1, pp. 208–212, 2011.

i. "Self-similarity-based image denoising," Commun. ACM, vol. 54, no. 5, pp. 109–117, May 2011, ISSN: 0001-0782. DOI: 10.1145/1941487.1941513. [Online]. Available: http://doi.acm.org/10.1145/1941487.1941513.

j. M. H. Alkinani and M. R. El Sakka, "Patch-based models and algorithms for image denoising: A comparative review between patch-based images denoising methods for additive noise reduction," EURASIP Journal on Image and Video Processing, vol. 2017, no. 1, pp. 1–27, 2017.

k. C.A. Deledalle, V. Duval, and J. Salmon, "Non-local methods with shape-adaptive patches (nlm-sap)," J. Math. Imaging Vis., vol. 43, no. 2, pp. 103–120, Jun. 2012, ISSN: 0924-9907. DOI: 10.1007/s10851- 011- 0294- y. [Online]. Available: http://dx.doi. org/10.1007/s10851-011-0294-y.

l. H. Bhujle and S. Chaudhuri, "Accelerating non-local denoising with a patch based dictionary," in Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing, ser. ICVGIP '12, Mumbai, India: ACM, 2012, 16:1–16:8, ISBN: 978-1-4503-1660-6. DOI: 10.1145/2425333.2425349. [Online]. Available: http://doi. acm.org/10.1145/2425333.2425349.

m. S. Sreehari, S. Venkatakrishnan, L. Drummy, J. Simmons, and C. A. Bouman, "Rotationally- invariant non-local means for image denoising and tomography," in 2015 IEEE International Conference on Image Processing (ICIP), IEEE, 2015, pp. 542–546.

2. Ram, M. Elad, and I. Cohen, "Image denoising using nl-means via smooth patch ordering," May 2013, pp. 1350–1354. DOI: 10.1109/ICASSP.2013.6637871.

a. J. Salmon and Y. Strozecki, "Patch reprojections for non-local methods," Signal Pro- cessing, vol. 92, no. 2, pp. 477–489, 2012, ISSN: 0165-1684. DOI: https://doi.org/10. 1016/j.sigpro.2011.08.011. [Online]. Available: http://www.sciencedirect.com/ science/article/pii/S0165168411002878.

b. K. N. Chaudhury and A. Singer, "Non-local euclidean medians," IEEE Signal Processing Letters, vol. 19, pp. 745–748, 2012.

c. K. Leng, "An improved non-local means algorithm for image denoising," in 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP), Aug. 2017, pp. 149–153. DOI: 10.1109/SIPROCESS.2017.8124523.

d. B. Chen, Y. Yan, L. Wang, M. Chen, C. Wang, and H. Ma, "One dimension nlm denoising method based on hasudorff distance and its application in otlc," in 2019 IEEE Asia Power and Energy Engineering Conference (APEEC), Mar. 2019, pp. 75–79. DOI: 10 . 1109 / APEEC.2019.8720666.

e. D. Van De Ville and M. Kocher, "Sure-based non-local means," IEEE Signal Processing Letters, vol. 16, no. 11, pp. 973–976, Nov. 2009, ISSN: 1558-2361. DOI: 10. 1109/ LSP. 2009.2027669.