

IMAGE DENOISING IN SPATIAL AND TRANSFORM DOMAINS

by

Mina Sharifymoghaddam

Bachelor of Science, Sharif University of Technology, 2012

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the Program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2015

©Mina Sharifymoghaddam 2015

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my dissertation may be made electronically available to the public.

Image Denoising in Spatial and Transform Domains

Master of Applied Science 2015

Mina Sharifymoghaddam

Electrical and Computer Engineering

Ryerson University

Abstract

Image denoising is an inseparable pre-processing step of many image processing algorithms. Two mostly used image denoising algorithms are Nonlocal Means (NLM) and Block Matching and 3D Transform Domain Collaborative Filtering (BM3D). While BM3D outperforms NLM on variety of natural images, NLM is usually preferred when the algorithm complexity is an issue. In this thesis, we suggest modified version of these two methods that improve the performance of the original approaches.

The conventional NLM uses weighted version of all patches in a search neighbourhood to denoise the center patch. However, it can include some dissimilar patches. Our first contribution, denoted by Similarity Validation Based Nonlocal Means (NLM-SVB), eliminates some of those unnecessary dissimilar patches in order to improve the performance of the algorithm. We propose a hard thresholding pre-processing step based on the exact distribution of distances of similar patches. Consequently, our method eliminates about 60% of dissimilar patches and improves NLM in terms of Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM).

Our second contribution, denoted by Probabilistic Weighting BM3D (PW-BM3D), is the

result of our thorough study of BM3D. BM3D consists of two main steps. One is finding a basic estimate of the noiseless image by hard thresholding coefficients. The second one is using this estimate to perform wiener filtering. In both steps the weighting scheme in the aggregation process plays an important role. The current weighting process depends on the variance of retrieved coefficients after denoising which results in a biased weighting. In PW-BM3D, we propose a novel probabilistic weighting scheme which is a function of the probability of similarity of noiseless patches in each 3D group. The results show improvement over BM3D in terms of PSNR for an average of about 0.2dB.

Acknowledgements

I would like to thank my supervisor, Dr. Soosan Beheshti, for her knowledge, patient guide, and support. It was my privilege and honor to work under Dr. Beheshti's supervision. I would also like to thank my examination committee members: Dr. Karthi Umapathy, Dr. Lian Zhao, and Dr. Ebrahim Bagheri for their valuable comments and feedback. Last but not least, I am truly grateful and would like to thank my family, especially my beloved sisters Sahel and Sayeh Sharifymoghaddam, for their understanding, love, and support.

Contents

<i>Declaration</i>	ii
<i>Abstract</i>	iii
<i>Acknowledgements</i>	v
<i>List of Tables</i>	viii
<i>List of Figures</i>	ix
<i>List of Appendices</i>	xii
1 Introduction	1
1.1 Literature review	1
1.2 Thesis contributions	4
2 Background	6
2.1 Nonlocal Means algorithm	7
2.2 BM3D image denoising	8
2.2.1 First Step: Basic Estimate	9
2.2.2 Second Step: Wiener Filtering	11
2.3 Noise Invalidation Technique	12
2.4 Performance Evaluation Criteria	15
2.4.1 Peak Signal to Noise Ratio	15
2.4.2 Structural Similarity Index	16
3 Similarity Validation Based NLM	17
3.1 Problem Formulation	17

3.2	Proposed Method	18
3.2.1	Step One: Patch Similarity Validation	18
3.2.2	Step Two: Weighting Process	19
3.2.3	Step Three: Smoothing Process	22
3.3	Simulation Results	22
4	Probabilistic Weighting BM3D	30
4.1	Grouping	31
4.1.1	Block Matching on Noisy Image	31
4.1.2	Block Matching on Basic Estimate	32
4.2	Denoising	32
4.2.1	Hard Thresholding	32
4.2.2	Wiener Filtering	33
4.3	Aggregation	35
4.3.1	Weighting in Hard Thresholding	35
4.3.2	Weighting in Wiener Filtering	36
4.4	Proposed Weighting	36
4.5	Simulation Result	37
5	Conclusions and Future Work	42
	References	51

List of Tables

2.1	Notations and default parameters of BM3D algorithm proposed by authors in [1]	13
3.1	Performance Comparison for Different Values of λ $k_{NIDe} = 3, k = 2$ & $\sigma = 25$ On Test Image Boat (512×512)	23
3.2	Performance Comparison (PSNR) for Different Patch Sizes for $\lambda = 3$ & $\sigma = 25$ On Test Image Boat (512×512)	24
3.3	Percentage of the eliminated patches by hard thresholding	24
3.4	Performance Comparison Over Test Images of Boat and Man (PSNR/SSIM) . .	26
3.5	Performance Comparison For Noise Standard Deviation of 30 (PSNR/SSIM) . .	26
4.1	Performance Comparison Over Test Image of Barbara (PSNR)	38
4.2	Performance Comparison Over Test Image of Cameraman (PSNR)	41
4.3	Performance Comparison For Noise Standard Deviation of 30 (PSNR)	41

List of Figures

2.1	Example of self similarity in an image. Similar pixel neighborhoods give a large weight, $w(p, q1)$ and $w(p, q2)$, while much different neighborhoods give a small weight $w(p, q3)$. [2]	8
2.2	Block Matching and 3D Collaborative Filtering Schematic [1].	9
2.3	Effect of sorting the absolute value of data: the top figure is 100 runs of a white Gaussian noise, the bottom figure is the sorted data [3]	14
2.4	Solid line is the noisy data, dashed lines are the noise only confidence boundaries. The portion of noisy data inside the boundary with high probability belongs to noise coefficients [3]	15
3.1	Similarity validation based nonlocal means (NLM-SVB)	18
3.2	Three scenarios of search neighbourhood S_i : (a) flat, (b) edge, (c) pattern ($\sigma=25$). Little red square in the middle is P_i . Right column: sorted distances of candidate patches, $d_{i,j}$ s, and pre-calculated probabilistic boundaries in (1.9) . . .	20
3.3	For search neighbourhood S_i s in Figure 3.2, First column: weights of PNLM, second column: weights of hard thresholding+PNLM, third and fourth columns: denoised versions of the images by PNLM and hard thresholding+PNLM respectively.	21
3.4	Test images: (a) boat (512×512), (b) man (512×512), (c) cameraman (256×256), (d) house (256×256), (e) barbara (512×512), (f) couple (512×512)	23
3.5	Estimation of the noiseless Cameraman (size 256×256) with noise standard deviation of 20 by NLM method. The PSNR is 29.17	27

3.6	Estimation of the noiseless Cameraman (size 256×256) with noise standard deviation of 20 by NLM-PET method. The PSNR is 28.65	27
3.7	Estimation of the noiseless Cameraman (size 256×256) with noise standard deviation of 20 by NLM-SAP method. The PSNR is 29.55	27
3.8	Estimation of the noiseless Cameraman (size 256×256) with noise standard deviation of 20 by Fast-NLM method. The PSNR is 29.49	27
3.9	Estimation of the noiseless Cameraman (size 256×256) with noise standard deviation of 20 by PNLM method. The PSNR is 29.51	28
3.10	Estimation of the noiseless Cameraman (size 256×256) with noise standard deviation of 20 by NLM-SVB method. The PSNR is 29.58	28
3.11	Estimation of the noiseless Barbara (size 512×512) with noise standard deviation of 30 by NLM method. The PSNR is 27.26	28
3.12	Estimation of the noiseless Barbara (size 512×512) with noise standard deviation of 30 by NLM-PET method. The PSNR is 26.91	28
3.13	Estimation of the noiseless Barbara (size 512×512) with noise standard deviation of 30 by NLM-SAP method. The PSNR is 27.90	29
3.14	Estimation of the noiseless Barbara (size 512×512) with noise standard deviation of 30 by Fast-NLM method. The PSNR is 27.50	29
3.15	Estimation of the noiseless Barbara (size 512×512) with noise standard deviation of 30 by PNLM method. The PSNR is 27.61	29
3.16	Estimation of the noiseless Barbara (size 512×512) with noise standard deviation of 30 by NLM-SVB method. The PSNR is 27.94	29
4.1	Estimation of the noiseless House (size 256×256) with noise standard deviation of 20 by BM3D method. The PSNR is 33.53	39
4.2	Estimation of the noiseless House (size 256×256) with noise standard deviation of 20 by Segmented-BM3D method. The PSNR is 33.50	39
4.3	Estimation of the noiseless House (size 256×256) with noise standard deviation of 20 by Sigmoid-BM3D method. The PSNR is 33.58	39

4.4	Estimation of the noiseless House (size 256×265) with noise standard deviation of 20 by PW-BM3D method. The PSNR is 33.63	39
4.5	Estimation of the noiseless Boat (size 512×512) with noise standard deviation of 30 by BM3D method. The PSNR is 29.03	40
4.6	Estimation of the noiseless Boat (size 512×512) with noise standard deviation of 30 by Segmented-BM3D method. The PSNR is 29.06	40
4.7	Estimation of the noiseless Boat (size 512×512) with noise standard deviation of 30 by Sigmoid-BM3D method. The PSNR is 29.05	40
4.8	Estimation of the noiseless Boat (size 512×512) with noise standard deviation of 30 by PW-BM3D method. The PSNR is 29.14	40

List of Appendices

1	Calculation of boundaries for validating patches	45
---	--	----

Chapter 1

Introduction

Image denoising is a main problem in image processing and is defined as a process aiming to recover an original clean image from its observed noisy version. Removing noise is an essential and the most fundamental pre-processing step in majority of image processing techniques such as medical and radar image analysis, image segmentation, visual tracking, classification and 3D object recognition where obtaining a good estimate of the clean image is crucial for strong performance, or it can only be used for the purpose of improving images visual quality.

1.1 Literature review

During the past few decades, several denoising techniques have been proposed. One of the earliest examples is median filter, where the value of the corrupted pixel is been replaced by the median value in a window, in order to estimate the noiseless version of the target pixel. The other one is the linear mean filter implemented by a convolution mask which replaces each noisy pixel with the average of itself and pixels around it in a neighbourhood [4]. The goal in some of these methods is to find a scheme to do weighting average instead of calculating simple mathematical mean. Weights are based on similarity between pixels. In general case, the spatial distance (relative location of the pixels) and the photometric distance (the difference in intensity values of the pixels) both affect this similarity measure. How to take into account these two impacts introduces different denoising algorithms. The classical one is Gaussian

smoothing filter. They compute weights only by spatial Euclidean distance between pixels in form of a Gaussian kernel. Lack of considering the structural (photometric) similarity in the image is the major drawback of this method. Another method is known as bilateral filters [5]. Authors proposed to consider both kind of distances in a separable manner. Weights are multiplication of two Gaussian kernels with two adaptable decaying parameters, one for spatial distance and the other for photometric distance. This approach has advantages over the previous one, however it is been shown that this filter still does not have good performance in low signal to noise ratio cases [6]. Another group of image restoration methods are through Bayesian filters. The main idea is to find the true image given the prior information of the noise and the observed noisy image. The challenge in this method is to find an appropriate prior [6].

Some methods known as patch based methods attempt to find those weights as a function of similarity between pre-defined shape patches around the target pixel rather than pixel-wise calculations. There are two categories in those methods, local and non-local methods. Most local methods only consider a local patch around the target pixel, assuming adjacent pixels tend to have similar patches. On the other hand, non local approaches take advantages of existence of a pattern or similar features in including the non-adjacent pixels in the denoising process [2]. Non-local means (NLM) originally introduced in 2005 by Budaes et al., exploits self-similarities in the search neighbourhood to estimate the true value of the noisy pixels. Due to its relative simplicity, NLM is the most well known and used spatial domain denoising methods, specifically when algorithm complexity is an issue [7]. This method is one of the concentrations of this work and will be introduced in the next chapter with more details.

Since the introduction of NLM, many other variations have been proposed to further improve the method from various perspectives. For example, nonlocal means with shape adaptive patches (NLM-SAP) is examined in [8]. They adaptively use shapes such as pie or quarter pie slices in addition to regular square patches. The main advantage of this approach is reducing the noise halo produced in high contrast edges. Another recent improvement, probabilistic nonlocal means (PNLM)[9], implements a new weight function based on the distribution of the distances of similar patches. This weighting scheme outperforms the Gaussian kernel weights in traditional NLM. The mathematical assumptions beyond this method is also explained more

in the next chapter.

In probabilistic early termination (NLM-PET) [10] the complexity of calculation in nonlocal means is decreased by reducing the number of patches involved in weighted averaging step by a pre-processing hard thresholding. When the partial sum of pixels starting from inner regions exceeds a probabilistic threshold they remove that candidate pixel from weighted averaging step. However, the overall performance of this method is worse than that of the traditional NLM due to not accurate calculation of those probabilistic thresholds.

Transforming signal to other bases may lead to better understanding the structures and features of it. Therefore, in some cases denoising can be achieved easier and more efficient by separating noise and signal coefficients in transform domains rather than spatial domain. In image denoising this is also popular. There are methods that transform image to other bases for the purpose of denoising such as wavelet or curvelet based methods [11]. The wavelet transform helps to analyze the signal in its different scales, details and approximation scales. Noise coefficients are characterized by low amplitude values spread across the wavelet coefficients. Therefore, thresholding in wavelet domain can remove noise coefficients. In [12], Donoho has introduced a method that performs denoising by soft thresholding wavelet coefficient on image affected by AWGN. They apply Haar wavelet transform to the noisy image. Soft threshold detailed coefficient and applied inverse transform. This threshold is a function of standard deviation of noise and the length of the wavelet coefficients. [13]

However usually natural images are not really sparse specially in the presence of textures or sharp transitions. This issue makes it impossible for any fixed 2D transform to achieve good sparsity for all cases. Recently, a new transform domain image denoising is introduced which benefits from enhanced sparse representation due to block matching similar fragments and grouping them in 3D blocks. Denoted by Block Matching and 3D Transform Domain Collaborative Filtering (BM3D), this method can separate noise by shrinking coefficients much better than 2D transform based methods and to the best of our knowledge is the state of the art of denoising methods [1], [14].

In BM3D with Sigmoid Shrinkage [15], authors suggest a flexible thresholding function based on mathematical sigmoid function with adjustable parameters instead of hard thresholding for

the first step of BM3D. They can change the thresholding parameters based on the sparsity of 3D blocks coefficients which is a function of noise level. This approach which does not have the discontinuities result from hard thresholding shows improvement over BM3D specially for higher noise levels. In BM3D with region growing segmentation [16], image is segmented in two partitions based on the intensity value of pixels. Block matching is done only on the involved segment. Then a Sobel edge detector is applied to improve the sharpness of edges.

1.2 Thesis contributions

In nonlocal means methods regardless of choice of the weight functions, many dissimilar patches in the search neighbourhood are processed through NLM. In order to address this issue, in Chapter 3 we propose a new hard thresholding pre-processing algorithm based on the exact distribution of similar patch distances to eliminate dissimilar patches before the weighting process. By calculating tight boundaries using NIDe technique for similar patch distances, we validate similarity of the reference patch with each candidate patch to process it through averaging step. Our proposed method is faithful to the probabilistic distribution of distance of similar patches. Our simulation results confirm superiority of this approach to the traditional NLM and the variations of this method.

The state of the art of image denoising methods is BM3D. Despite its superior performance, BM3D is a relative complex method and many of the theoretical approach behind steps of BM3D has not been studied yet. We start Chapter 4 by a more in depth study of BM3D. Then we introduce two main issues with the current weighting process of BM3D in both hard thresholding and wiener filter denoising steps and propose a probabilistic weighting scheme that has been shown to outperform the existing method.

The reminder of this thesis is outlined as follows. In Chapter 2 a review of two popular image denoising methods nonlocal means and BM3D are presented. Also an overview of the original noise invalidation technique over one dimensional data set has been presented. In Chapter 3, our work on improving non local means by validating patch similarities is been introduced. The results and performance comparison are available in corresponding sub sections. In Chap-

ter 4, we described our method for improving the state of the art BM3D using probabilistic weights. The related simulation results have also been presented. Finally, Chapter 5 presents the concluding remarks and offers some suggestions for future work.

Chapter 2

Background

Additive White Gaussian Noise (AWGN) is the most common model of the noise considered in image processing. That is when the power of the noise is constant over all frequencies (a flat power spectral density), the amplitude of noise follows the probability density of Gaussian distribution and noise values are been added to the original signal. The probability distribution of the noise in this scenario can be formalized as follow:

$$f(n) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(n - \mu)^2}{2\sigma^2}\right) \quad (2.1)$$

$$y = x + n; \quad (2.2)$$

where f is the probability distribution function of the noise amplitude, μ is the mean of the noise and σ is defined as standard deviation of the noise (constant parameters). Here n is the value of noise, x and y are the original and corrupted data. Some early methods try to separate the image into two parts: smooth part (original image) and the oscillatory part (noise). However images are not truly smooth in structure. They usually have fine details and edges with high frequencies. When the high frequencies are removed some information of the original image will be lost along with the noise as well [17]. Buades et al. have developed the Nonlocal Means image denoising to overcome this issue and somehow differentiate between high frequency noise

and images fine details.

2.1 Nonlocal Means algorithm

Non-Local Means is the most well-known image denoising method and has proved its ability to challenge other powerful methods such as wavelet based approaches or variational techniques. It is relatively simple to implement and efficient in practice. It is very similar to bilateral denoising method, considering both geometric and photometric distance of pixels. However it takes advantage of similarity between pixels far from the target pixel (non local) in addition to neighbourhood (local) pixels. It process the similarity measure over a square sub-image around two candidate pixels called patch. Similar to previous methods patches with higher similarity measures will have higher weight. Figure 2.1 shows one sample of target patch and three candidate reference patches. Let's assume v_i s are pixels inside the noisy image I . Each estimated pixel $NL(v_i)$ is a weighted average of all pixels in the image:

$$NL(v_i) = \sum_{j \in I} w(i, j) v_j \quad (2.3)$$

where $w(i, j)$ s depend on the similarity between pixels at i and j , $0 < w(i, j) < 1$ and summation of $w(i, j)$ s for each reference pixel i is equal one. The similarity between pixels i and j is measured by the similarity between square neighborhood of fixed sized around them called P_i and P_j . Weight is a decreasing function of Euclidean distance between patches:

$$w(i, j) = \frac{1}{Z_i} e^{-\frac{\|N_i - N_j\|_{2,a}^2}{h^2}} \quad (2.4)$$

In this formula, $\|\cdot\|_2$ is the $\ell-2$ norm distance of vectorized patches, a is the standard deviation of Gaussian kernel multiplied to consider the geometric distance of pixels in the patch and h is the decaying parameter controls the amount of blurring. Z_i is the normalization constant to make the summation of weights for each pixel equal one [2]. To decrease the computation complexity of algorithm it has been shown that it is not necessary to investigate the whole image in order to find the most similar patches. Finding a fair amount of similar patches in a

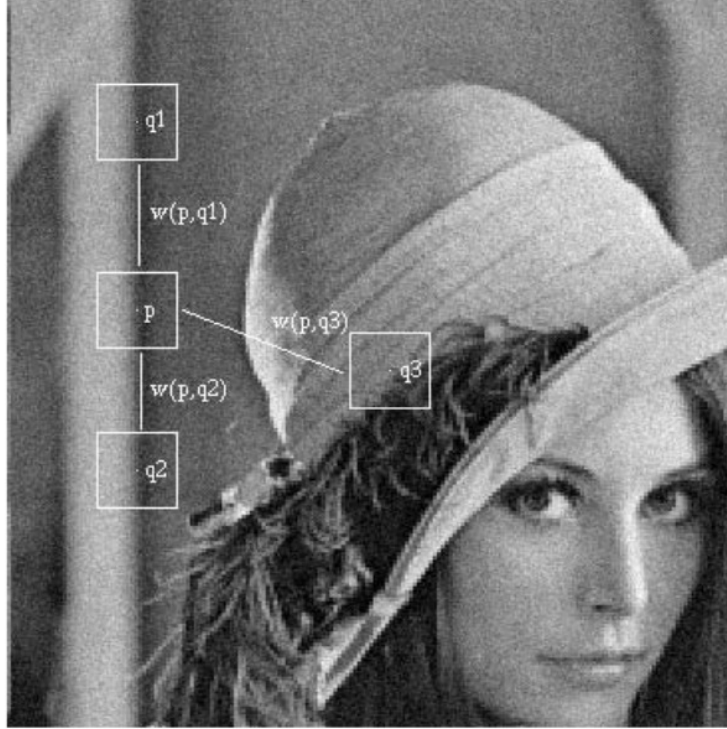


Figure 2.1: Example of self similarity in an image. Similar pixel neighborhoods give a large weight, $w(p, q1)$ and $w(p, q2)$, while much different neighborhoods give a small weight $w(p, q3)$. [2]

smaller neighbourhood called search neighbourhood can work as well as repeating the process in the whole image.

2.2 BM3D image denoising

Image Denoising by Block Matching and 3D Transform-Domain Collaborative Filtering (BM3D) is the state of the art image denoising method and is introduced by Dabov et al. in 2007 [1]. It has two main steps. The first one is a wavelet shrinkage process applied on a 3D group of similar patches called stack. The second step is using this basic estimate to process wiener filter to 3D groups. This algorithm retrieves the finest details of patches by preserving the unique features of each individual stack [18]. These steps are well illustrated in Figure 2.2.

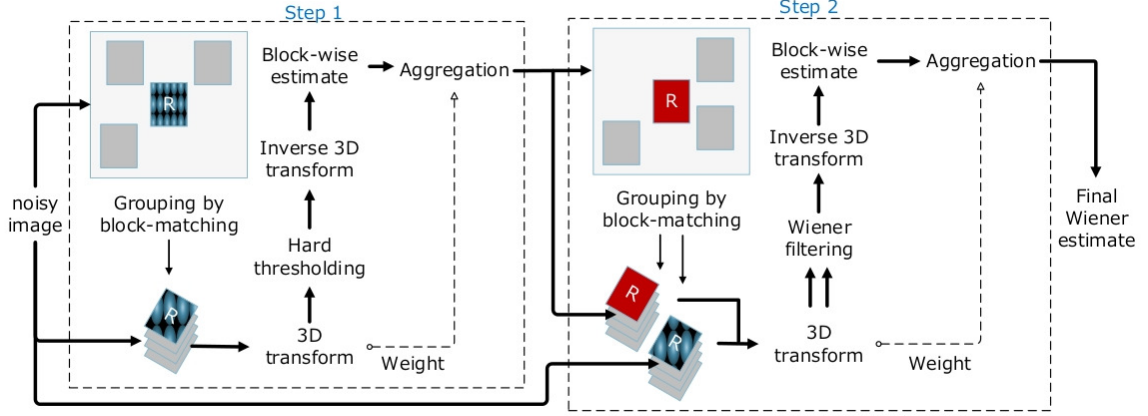


Figure 2.2: Block Matching and 3D Collaborative Filtering Schematic [1].

2.2.1 First Step: Basic Estimate

The first step in this algorithm is the dominant one. The output of this step is fairly well retrieved. Parameters in this step are shown with a superscript ht . The idea behind this step is that by grouping similar patches into 3D arrays, sparsity of data in transform domain is enhanced. In other words, the difference between signal and noise coefficients become more visible consequently thresholding leads to better estimation of noiseless data. If y is the true image and z is the observed noisy one, \hat{e} denote the basic estimate of the image by \hat{y}_{basic} and the final estimate by \hat{y}_{final} . We name the patch of size N_1^{ht} with top left corner at pixel x by Z_x .

Grouping

We call the currently processed patch the reference patch and name it Z_{x_R} . Firstly, we find the blocks that are similar to the reference one within a fixed search neighborhood. The similarity measure is a function of patches distance. We define the distance as normalized Euclidean distance of vectorized patches. In order to degrade the effect of noise with high standard deviation, it is been proposed to apply a 2D transform followed by a hard thresholding

coefficients to pre-smooth patches only for standard deviation of noise greater than 40.

$$d(Z_{x_R}, Z_x) = \frac{\|\Upsilon'(\tau_{2D}^{ht}(Z_{x_R})) - \Upsilon'(\tau_{2D}^{ht}(Z_x))\|_2^2}{(N_1^{ht})^2} \quad (2.5)$$

In this equation, τ_{2D}^{ht} is the explained 2D transform and Υ' is the hard thresholding operator with threshold $\lambda_{2D}\sigma$. The grouping is done by comparing this distance by a threshold value and keeping at most N_2^{ht} of the most similar patches in the group. The threshold value is τ_{match}^{ht} . We call the set of top left coordinates of these patches $S_{x_R}^{ht}$.

$$S_{x_R}^{ht} = \{x | d(Z_{x_R}, Z_x) < \tau_{match}^{ht}\} \quad (2.6)$$

We stack them in a 3D group and call it \mathbf{Z}_{x_R} .

Collaborative Hard-thresholding

We apply a 3D transform to the 3D group \mathbf{Z}_{x_R} , then we hard threshold noise coefficients, produce estimate of the 3D block, apply inverse transform and return them to their original places.

$$\hat{\mathbf{Y}}_{basic_{x_R}} = \tau_{3D}^{ht^{-1}}(\Upsilon(\tau_{3D}^{ht}(\mathbf{Z}_{x_R}))) \quad (2.7)$$

In this equation τ_{3D}^{ht} is the 3D transform usually composed of a 2D transform over patches and then a 1D transform over the third dimension. The inverse 3D transform is denoted by $\tau_{3D}^{ht^{-1}}$. The Υ is the hard thresholding operator with a threshold value of $\lambda_{3D}\sigma$.

Aggregation

Aggregation is defined as a process to weighted average more than one estimate of a pixel. The block-wise estimations of pixels have overlap. Consequently we need to aggregate them for each pixel. Proper selection of this weights is very critical [19]. The authors of BM3D suggested weights should be inversely proportional to the total sample variance of the corresponding block-wise estimates. More dissimilar or noisy the 3D group estimates are the less the contribution (weight) is. In this case the total sample variance would be $\sigma^2 N_{har}^{x_R}$, where N_{har} is the number

of retained (non-zero) coefficients after hard thresholding. So we assign weights as:

$$w_{x_R}^{ht} = \frac{1}{\sigma^2 N_{ht}^{x_R}} \quad (2.8)$$

In Chapter 4, the two main problems of this weighting scheme will be discussed and a new weighting scheme will be proposed to improve BM3D further.

2.2.2 Second Step: Wiener Filtering

In this step, both the basic estimate and the noisy image are used to improve the denoising. We estimate the true image using power spectrum of the basic estimate and applying wiener filter shrinkage again in 3D transform domain over stacks. Parameters in this step are shown with a superscript *wie*.

Grouping

For each patch in the basic estimate we repeat the block matching process in the first step. The only difference is that since the image is already pre-filtered the effect of noise with high standard deviation does not exists any more, so we can calculate the distance of two patches with size N_1^{wie} by N_1^{wie} directly on the spatial domain as below:

$$d(\hat{Y}_{x_R}^{basic}, \hat{Y}_x^{basic}) = \frac{\|\hat{Y}_{x_R}^{basic} - \hat{Y}_x^{basic}\|_2^2}{(N_1^{wie})^2} \quad (2.9)$$

We again compare this distance by the threshold τ_{match}^{wie} (which obviously should be smaller than the threshold in the first step) and keep at most N_2^{ht} of the most similar patches in the basic estimate. We denote this stack by $\hat{Y}_{x_R}^{basic}$. We create another 3D stack in the same order of the previous one from the noisy image and call it $Z_{x_R}^{basic}$.

Wiener Filter Shrinkage

3D transform is applied to both groups. We perform collaborative Wiener filtering on the noisy group using energy spectrum of the basic estimate as the pilot. Then apply inverse 3D

transform and return patches to their original places. We define wiener coefficients from the energy of the 3D transform coefficients of the basic estimate group:

$$\mathbf{W}_{x_R} = \frac{|\tau_{3D}^{wie}(\hat{\mathbf{Y}}^{basic}_{x_R})|^2}{|\tau_{3D}^{wie}(\hat{\mathbf{Y}}^{basic}_{x_R})|^2 + \sigma^2} \quad (2.10)$$

Then we multiply it element-wise by the 3D array of noisy image:

$$\hat{\mathbf{Y}}_{x_R}^{wie} = \tau_{3D}^{wie-1}(\mathbf{W}_{x_R} \tau_{3D}^{wie}(\mathbf{Z}^{basic}_{x_R})) \quad (2.11)$$

Aggregation

Same as the first step, we compute a final estimate of the true image by aggregating all of the obtained local estimates using a weighted average. Weights for each 3D group are defined as:

$$w_{x_R}^{wie} = \frac{1}{\sigma^2 \|\mathbf{W}_{x_R}\|_2^2} \quad (2.12)$$

There is also a major drawback in this formulation of aggregation weights which is discussed in Chapter 4. Default values of BM3D for all the mentioned parameters are presented in Table 2.1

2.3 Noise Invalidation Technique

Noise invalidation is an adaptive technique to drive a noise signature from the noise statistics in order to separate noise from data. This method was originally proposed over one dimensional data. It does not consider any particular assumption on the structure of the noise-free signal and performs better than other thresholding denoising methods. A brief overview of this method is provided here [3]:

For each candidate threshold z , we can define the signature function for one sample of a noise random variable as below:

$$g(z, v_i) = \begin{cases} 1 & \text{if } v_i \leq z \\ 0 & \text{otherwise} \end{cases} \quad (2.13)$$

Table 2.1: Notations and default parameters of BM3D algorithm proposed by authors in [1]

Parameter	Definition	Default Value
z_x	noisy image pixel located at x	
y_x	true image pixel located at x	
\hat{y}_x^{basic}	basic estimation pixel located at x	
\hat{y}_x	denoised pixel located at x	
η_x	noise value added to the pixel located at x	
σ	standard deviation of the noise	
τ_{match}^{ht}	threshold parameter for similar distances in step 1	$\sigma < 40 ? 2500:5000$
τ_{match}^{wie}	threshold parameter for similar distances in step 2	400
Z_x	Patch extracted from noisy image located at x	
\mathbf{Z}_x	block extracted from noisy image located at x	
N_1^{ht}	Patch size in step 1	8
N_1^{wie}	Patch size in step 2	8
N_2^{ht}	maximum number of patches in 3D group in step 1	16
N_2^{wie}	maximum number of patches in 3D group in step 2	32
τ_{3D}^{ht}	normalized 3D transform for denoising in step 1	2D-Bior1.5+1D-Walsh
τ_{3D}^{wie}	normalized 3D transform for denoising in step 2	2D-DCT+1D-Walsh
Υ'	hard-threshold operator with $\sigma\lambda_{2D}$ for block matching in step 1	λ_{2D} is zero for $\sigma < 40$
Υ	hard-threshold operator with $\sigma\lambda_{3D}$ for denoising in step 1	λ_{3D} is 2.7

We name the expected value and variance of this signature as $G_E(z)$ and $G_{Var}(z)$. For a vector of the noise random variable of length N , we drive the signature as:

$$g(z, v^N) = \frac{1}{N} \sum_{i=1}^N g(z, v_i) \quad (2.14)$$

It is straightforward to show that:

$$E(g(z, V^N)) = G_E(z) \quad (2.15)$$

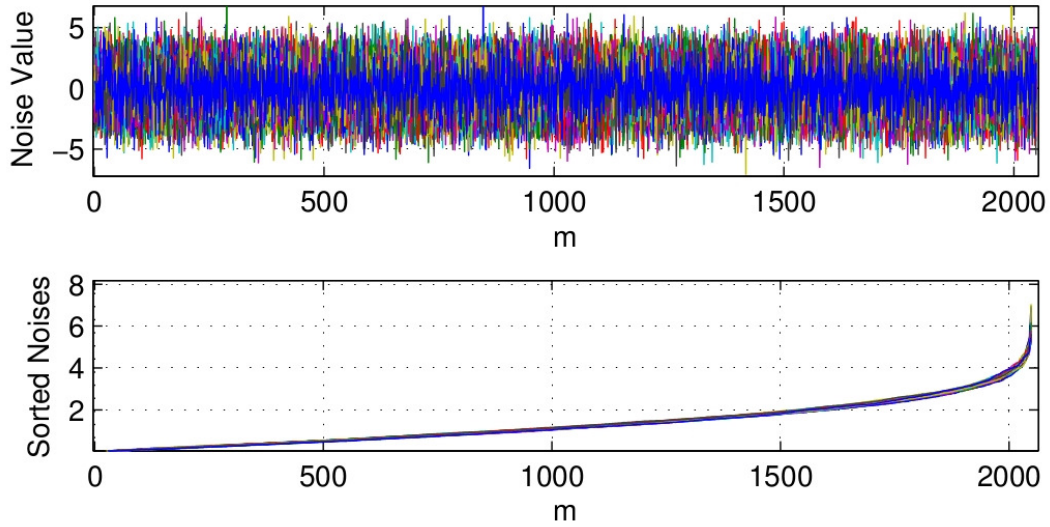


Figure 2.3: Effect of sorting the absolute value of data: the top figure is 100 runs of a white Gaussian noise, the bottom figure is the sorted data [3]

and

$$\text{Var}(g(z, V^N)) = \frac{1}{N} G_{\text{Var}}(z) \quad (2.16)$$

It can be shown that for each z , the above signature for the vector of noise samples can also be written as:

$$g(z, v^N) = \frac{m}{N} \quad (2.17)$$

where m is the number of samples with absolute value less than z , equivalently when sorting v_i s the m th value is the largest v_i less than z [20]. The effect of this signature function derived from sorting is illustrated in Figure 2.3.

As it can be seen in the figure the sorted data are in such a denser area than the unsorted ones i.e. they have less variance. This property helps us to define a tight boundary around the mean of the sorted data coefficient to detect noise coefficient which are inside that boundary with high probability. These boundaries are as below and shown in Figure 2.4. In the next chapter, a 2D version of this method is used over the vector of patch distances in order to

$$L(z) = G_E(z) - \lambda \sqrt{\frac{1}{N} G_{\text{Var}}(z)} \quad (2.18)$$

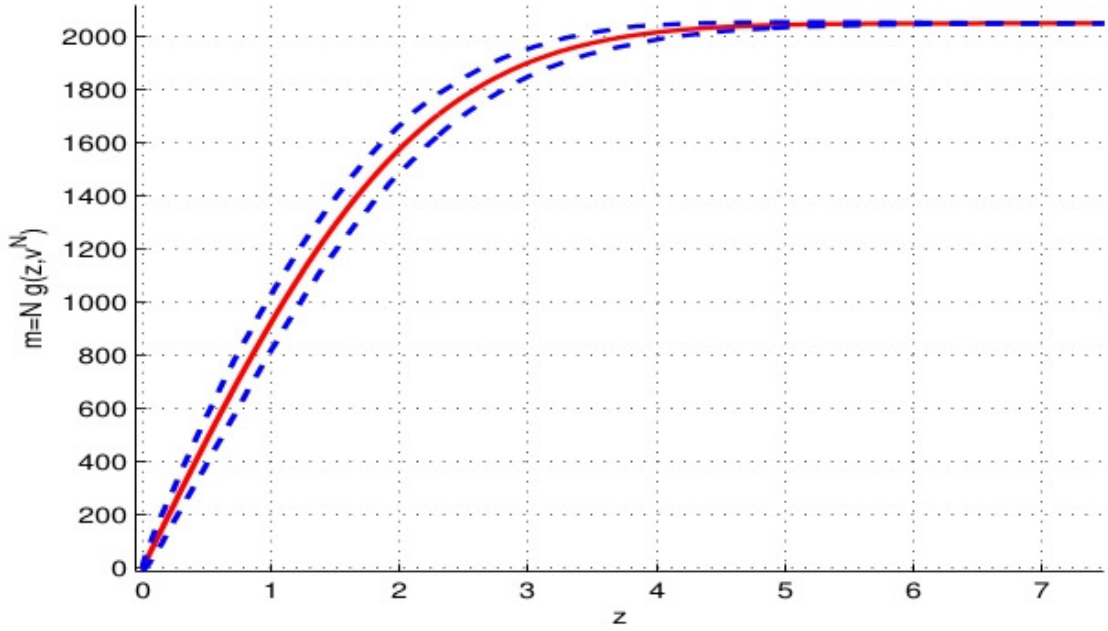


Figure 2.4: Solid line is the noisy data, dashed lines are the noise only confidence boundaries. The portion of noisy data inside the boundary with high probability belongs to noise coefficients [3]

$$U(z) = G_E(z) + \lambda \sqrt{\frac{1}{N} G_{Var}(z)} \quad (2.19)$$

2.4 Performance Evaluation Criteria

The quality measurement criteria used for performance evaluations are Peak Signal to Noise Ratio (PSNR) and (Structural Similarity Index Measure) SSIM which both compare the estimated image and the original true image.

2.4.1 Peak Signal to Noise Ratio

Peak Signal to Noise Ratio (PSNR) is the most popular quantitative metric in analysing retrieved images and it is a function of Mean Square Error (MSE). The Mean Square Error is

defined as squared Euclidean distance of two images. Larger PSNR values mean better denoising and it does not depend on any visual structure and details. With the condition of pixel intensities between zero and one we have the following equation for calculating PSNR between two images X and \hat{X} [21]:

$$PSNR(X, \hat{X}) = 20 \log_{10} \left(\frac{1}{\|X - \hat{X}\|_2} \right) \quad (2.20)$$

2.4.2 Structural Similarity Index

Structural Similarity Index Measurement (SSIM) shows edges and fine details preservance in the denoised image and it indicates more visual similarity between two images than PSNR. The following equations show the formulas for the mentioned quality criteria,[22]:

$$SSIM(X, \hat{X}) = \frac{(2\mu_X\mu_{\hat{X}} + k_1^2)(2\sigma_{X\hat{X}} + k_2^2)}{(\mu_X^2\mu_{\hat{X}}^2 + k_1^2)(\sigma_X^2\sigma_{\hat{X}}^2 + k_2^2)} \quad (2.21)$$

where μ_X and $\mu_{\hat{X}}$ are the mean values of the images, σ_X^2 and $\sigma_{\hat{X}}^2$ are the variance of the images, $\sigma_{X\hat{X}}$ is the covariance of two images and k_1 and k_2 are two constants.

Chapter 3

Similarity Validation Based NLM

3.1 Problem Formulation

We consider an image corrupted with an additive white Gaussian noise (AWGN) with zero mean and variance σ^2 , where y_i is the i th noisy pixel value and x_i is the i th true pixel value:

$$y_i = x_i + n_i; \quad \forall i : n_i \sim \mathcal{N}(0, \sigma^2) \quad (3.1)$$

The goal is to recover the noise free image from the observed noisy image. In the conventional NLM methods, each estimated pixel, \hat{x}_i , is a weighted average of other pixels in a search neighbourhood S_i :

$$\hat{x}_i = \frac{\sum_{j \in S_i} w_{i,j} y_j}{\sum_{j \in S_i} w_{i,j}} \quad (3.2)$$

where $w_{i,j}$ is the weight between square patches centered at pixels i and j . The weight is a function of squared value of ℓ_2 -norm distance between two local patches P_i and P_j with centers at pixels i and j :

$$d_{i,j} = \|P_i - P_j\|_2^2 \quad (3.3)$$

$$w_{i,j} = e^{-d_{i,j}/h} \quad (3.4)$$

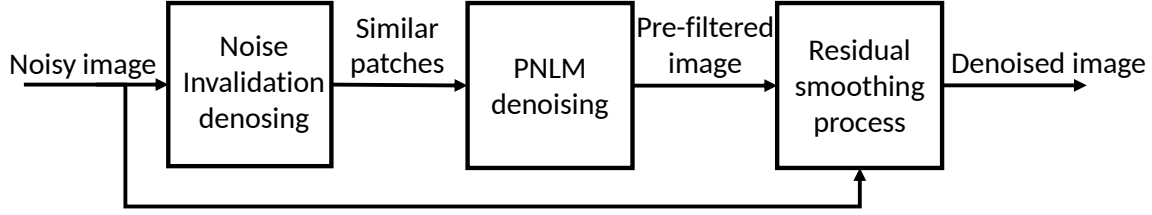


Figure 3.1: Similarity validation based nonlocal means (NLM-SVB)

Where $\|\cdot\|_2$ is Euclidean distance of vectorized patches difference. This weight, used in traditional NLM, is a Gaussian kernel weight, where h is a decaying parameter and is usually set to 10σ [2].¹

3.2 Proposed Method

Our proposed method, denoted by similarity validation based nonlocal means (NLM-SVB), consists of three steps shown in Figure 3.1. In the following these three steps are explained in detail.

3.2.1 Step One: Patch Similarity Validation

Using fundamentals of NLM, for each reference patch the distance of that patch and the patches in searching area S_i is first calculated. The goal is to keep similar patches in this area for further processing in next steps. Two patches are considered similar if their distance is only due to additive noise. Due to the nature of the distance, d_i, j in (3.3) this distance has a chi-squared distribution where the distribution for x is defined as:

$$\chi_k^2(x) = \frac{x^{(k/2-1)}e^{-x/2}}{2^{(k/2)}\Gamma(k/2)} \quad (3.5)$$

¹In the original formulation, they multiply this function by a neighbourhood Gaussian kernel to consider the geometric distance between pixels in a search neighborhood.

where Γ denotes the Gamma function and k is the order of the distribution. Motivated by this definition of similarity in the first step, our goal is to hard threshold as many dissimilar patches as possible. The procedure is as follows:

For any i th center patch, we first sort all the $d_{i,j}$ s in its search neighborhood S_i . In this case, similar patches with $d_{i,j}$ s following Chi-squared distribution fall within a probabilistic boundaries that can be pre-calculated based on that Chi-squared distribution. Details of calculation of these boundaries are provided in Appendix 1. Using this probabilistic boundaries an example of the hard thresholding, that is also explained in Appendix 1, is as follows:

Figure 3.2 shows the probabilistic bounds and sorted $d_{i,j}$ s for three cases of a flat, an edge and a pattern search neighbourhood respectively. Red squares show the reference patch P_i . Note that these boundaries are fixed for all three cases and only function of the σ and the size of S_i . Consequently, the hard thresholding process considers any j th patch with its $d_{i,j}$ out of this boundary as a dissimilar patch to the i th patch. For example, after sorting the patch distances, at index $j = 1000$ the probabilistic upperbound and lowerbound with probability 99.8% (3σ probabilistic confidence) are 0.9114 and 0.6546. As the figure shows for the flat scenario, $d_{i,j}$ at index $j = 1000$ is 0.8962, which falls within the boundaries. However, this value is 1.0116 and 1.1483 for edge and pattern scenarios respectively that are out of the boundaries. Therefore, 1000th sorted pixel is passed to step 2 for the first scenario, while being discarded (set to zero) for the second and third scenarios.

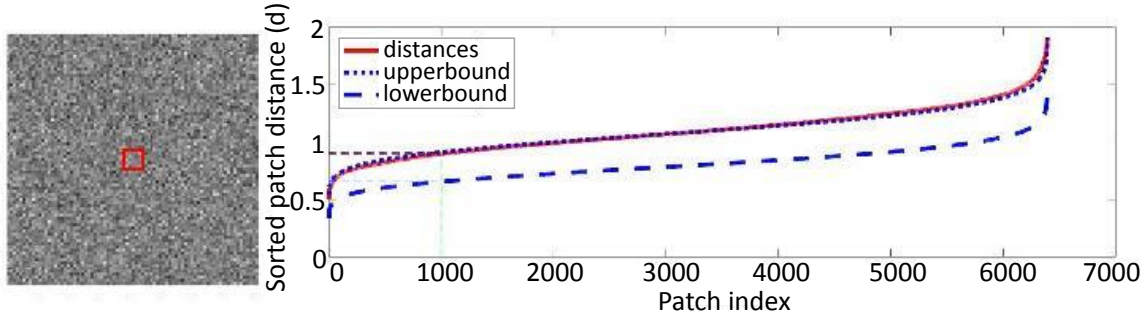
3.2.2 Step Two: Weighting Process

After elimination of dissimilar patches through the hard thresholding, the remaining patches are processed in the weighting stage. For this stage, our weights in (3.2) are consistent with the corresponding Chi-squared distribution in (3.5) [9]:

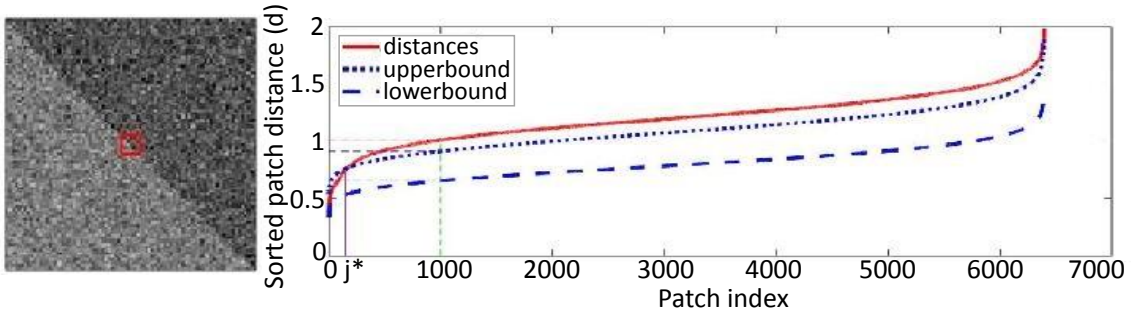
$$w_{i,j} = \chi_{\eta_{i,j}}^2(d_{i,j}^n/\gamma_{i,j}) \quad (3.6)$$

where:

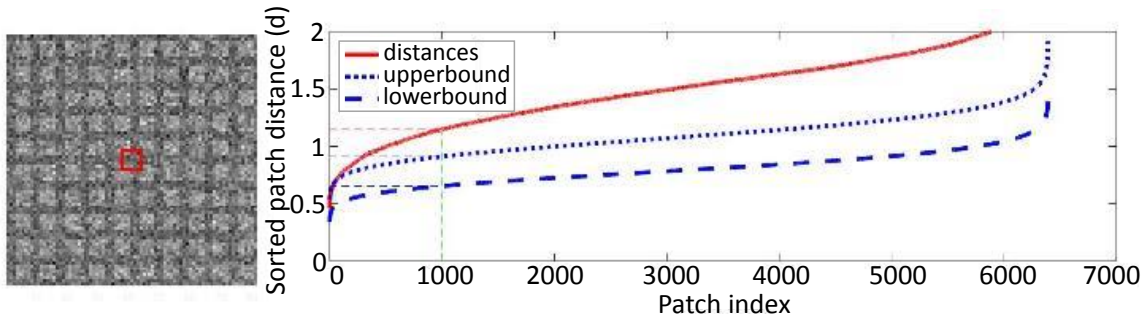
$$\gamma_{i,j} = (2|P_i| + |O_{i,j}|)/2|P_i|; \quad \eta_{i,j} = |P_i|/\gamma_{i,j} \quad (3.7)$$



(a)



(b)



(c)

Figure 3.2: Three scenarios of search neighbourhood S_i : (a) flat, (b) edge, (c) pattern ($\sigma=25$). Little red square in the middle is P_i . Right column: sorted distances of candidate patches, $d_{i,j}$ s, and pre-calculated probabilistic boundaries in (1.9)

and $|P_i|$ is the number of pixels in P_i and $|O_{i,j}|$ is the number of overlapping pixels between P_i and P_j . This step can be considered as a soft thresholding stage after a hard thresholding stage, both consistent and faithful to the exact distribution of $d_{i,j}$ s for similar patches.

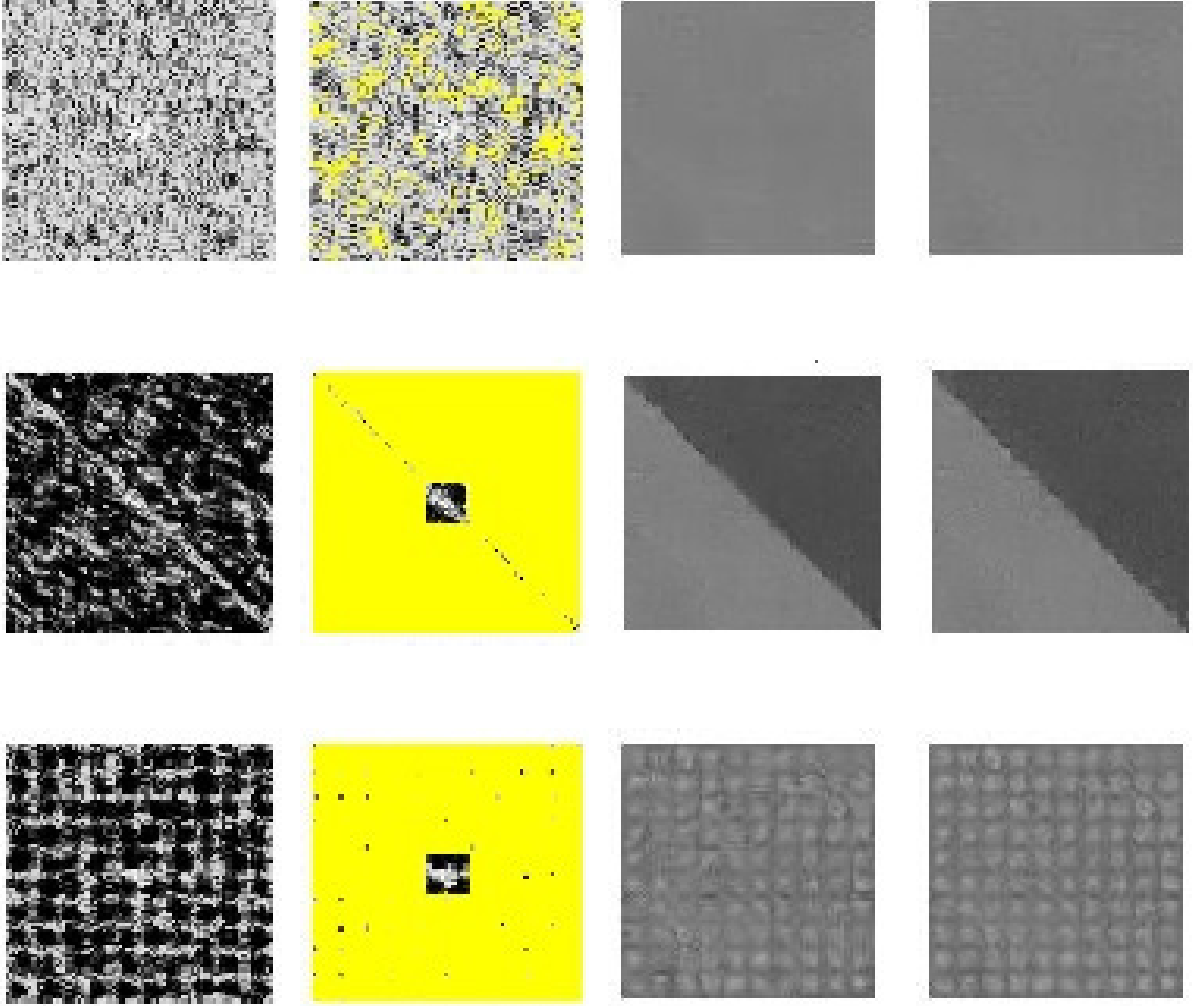


Figure 3.3: For search neighbourhood S_i s in Figure 3.2, First column: weights of PNLM, second column: weights of hard thresholding+PNLM, third and fourth columns: denoised versions of the images by PNLM and hard thresholding+PNLM respectively.

Advantages of pre-processing thresholding before weighting process

Figure 3.3 shows how our additional hard thresholding benefits the existing soft thresholding (PNLM) for the same scenarios as in Figure 3.2. The first column shows the associated weights of PNLM while the second column shows the weights for hard thresholding+PNLM. The additional zero weighted ones are shown in yellow in the second column. Comparing these two

columns, the additional hard thresholding has zeroed the weights of many dissimilar patches (18% for flat case, 95% for edge case and 96% for pattern case). As the figures show, the remaining pixels are highly related (very similar) to the center pixel. The third and forth columns show the denoised results. As these two columns show elimination of the dissimilar patches resulted better denoised image, specially for the cases of edge and pattern structure, where with the additional hard thresholding fine details are well retrieved [23].

3.2.3 Step Three: Smoothing Process

This stage uses the conventional smoothing filter [24]:

$$\hat{x}_i^{new} = \hat{x}_i + \lambda D(y_i - \hat{x}_i) \quad (3.8)$$

where D is the smoothing denoising function and λ is the added percentage of smoothed residuals. A mean filtering is applied over residuals, $y_i - \hat{x}_i$, [25]. For each pixel of the residual image, the mean value of pixels in a 3×3 neighbourhood is calculated to replace the center value and $\lambda = 10\%$ [26].

3.3 Simulation Results

Our test images are boat, man, cameraman, house, barbara and couple shown in Figure 3.4. Parameter λ in the invalidation step, shows the width of those probabilistic boundaries. If we choose a higher value not many of the pixels will be eliminated from the averaging step. Therefore, the effect of NIDe selection will become lower. If we choose a smaller value some related patches with their distance only because of the noise will be eliminated. In the following simulation, we changed the value of λ over test image of boat (512×512). Table 3.1 shows that λ around 3 gives us higher PSNR.

Parameters k_{NIDe} and k show the radius of patches in the NIDe filtering and applying probabilistic weights steps respectively. Large patches lead to unwanted smoothing and lose of many details, while too small patch sizes cause many low frequency variation in the image. Here it should be mentioned that in the proposed PNLM method authors suggest using patches

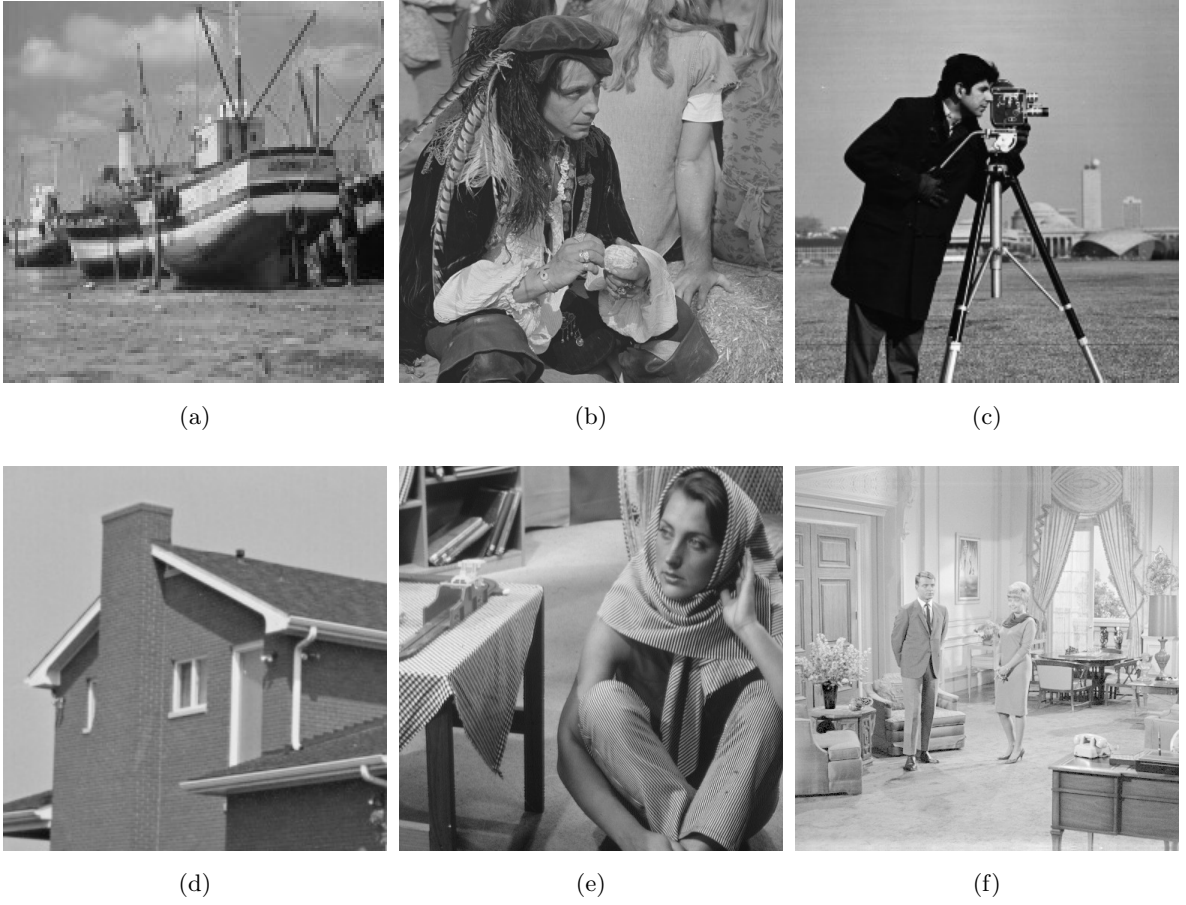


Figure 3.4: Test images: (a) boat (512×512), (b) man (512×512), (c) cameraman (256×256), (d) house (256×256), (e) barbara (512×512), (f) couple (512×512)

Table 3.1: Performance Comparison for Different Values of λ $k_{NIDe} = 3$, $k = 2$ & $\sigma = 25$ On Test Image Boat (512×512)

λ	0.5	1	1.5	2	2.5
PSNR(dB)	28.77	28.78	28.78	28.79	28.79
λ	3	3.5	4	4.5	5
PSNR(dB)	28.79	28.78	28.77	28.76	28.76

of size 3×3 . However, our results show patches of size 5×5 or k equals 2 gives us higher PSNR in all of our test images (size 512×512).

Table 3.2: Performance Comparison (PSNR) for Different Patch Sizes for $\lambda = 3$ & $\sigma = 25$ On Test Image Boat (512×512)

k/k_{NIDe}	1	2	3	4
1	28.36	28.68	28.64	28.53
2	28.52	28.75	28.79	28.73
3	28.43	28.60	28.62	28.60
4	28.18	28.32	28.35	28.34

The resulted percentage of patch elimination due to hard-thresholding for $\sigma = 25$ is provided in Table 3.3. As the table shows, on average around 60% of patches are being discarded before the weighting process. Note that this percentage is higher in images with fine details such as man and barbara, while it is lower for images with less details such as house.

Table 3.3: Percentage of the eliminated patches by hard thresholding

σ	boat	man	cameraman	house	barbara	couple
25	63.5%	65.6%	57.7%	51.4%	64.0%	61.6%

The quality measurement criteria used for the performance evaluation are PSNR [21] and SSIM [22]. The proposed method is compared to NLM and NLM-PET[10], NLM-SAP[8], Fast NLM[27] and PNLM[9] that are variations of NLM. For all these methods the tuning parameters in their referenced papers are used. While PNLM patches have a 3×3 size, our optimum patch size in combined approach is 5×5 as explained. The search neighbourhood S_i is a square window of size 21×21 .

Table 3.4 shows the results for boat and man over a wide range of noise standard deviations. The column with title "Intermediate" is our algorithm without the third step of smoothing. As

it can be seen, the performance of NLM-PET is worse than that of the traditional NLM. The hard thresholding used in this approach eliminates even some similar patches. NLM-SAP on the other hand, outperforms NLM as in this method patch shapes are adaptive.

PNLM outperforms NLM, since it uses the weight function based on the true distribution of similar patches. The Fast NLM method in [27], outperforms the NLM by removing dissimilar patches based on their gradient. As the table shows, NLM-SVB outperforms all the above methods as it takes advantage of PNLM weights, as well as exploiting the additional pre-processing hard thresholding. Comparing the last two columns shows that hard thresholding step contributes drastically more than smoothing process in the overall improvement. The results for all other images, cameraman, house, barbara and couple are similar to what is shown for boat and man. Due to the space limitation, those results are shown only for $\sigma = 30$ in Table 3.5. Figures 3.5- 3.16 show the visual quality of all studied methods. By comparing NLM-SVB with PNLM and NLM denoised images, it can be seen that sharper edges are more visible in our method. So this method can be used in scenarios when edge preservance is more important than actual denoising such as features extraction.

With ignoring different implementation methods which can speed up the algorithms and by fixing patch sizes in all the methods, our algorithm has less order of complexity than method such as NLM-SAP. However, due to sorting its complexity is comparable to methods such as NLM-PET and Fast NLM. Note that a more advanced smoothing process (a new weighting scheme) is used in [26] which is currently comparable with our method. Combining this post-processing step with our existing method in future can further improves the overall performance.

Table 3.4: Performance Comparison Over Test Images of Boat and Man (PSNR/SSIM)

σ	Noisy Image	Boat (512×512)							
		NLM	NLM-PET	NLM-SAP	Fast NLM	PNLM	Intermediate	NLM-SVB	
10	28.13/70.63	32.63/86.35	32.50/84.97	32.91/87.53	32.79/86.95	32.91/87.60	33.03/88.05	33.10/88.02	
30	18.59/32.34	27.16/70.23	27.01/70.15	27.66/75.47	27.50/73.53	27.28/70.71	27.85/74.10	27.94/76.19	
50	14.16/19.11	24.61/56.18	24.54/55.22	25.09/65.81	24.82/65.49	24.61/58.71	25.33/64.38	25.58/66.54	
70	11.23/12.92	23.08/45.89	22.96/45.19	23.61/53.05	23.39/49.84	22.98/50.05	23.75/55.76	24.01/57.12	
90	9.05/9.52	21.99/38.21	21.93/37.46	22.66/47.72	22.38/43.60	21.92/43.73	22.51/46.13	22.87/49.08	
σ	Noisy Image	Man (512×512)							
		NLM	NLM-PET	NLM-SAP	Fast NLM	PNLM	Intermediate	NLM-SVB	
10	28.14/68.60	32.56/86.81	32.10/86.26	33.13/88.98	32.92/87.55	33.16/89.18	33.56/89.53	33.64/89.55	
30	18.59/29.16	27.33/68.28	27.01/67.89	27.79/71.99	27.65/70.59	27.39/68.87	27.96/73.85	28.10/75.85	
50	14.16/16.76	25.05/54.61	24.62/54.39	25.52/60.37	25.47/59.80	24.89/56.12	25.50/61.88	25.88/64.24	
70	11.23/11.18	23.57/44.92	23.37/44.12	24.24/53.48	23.95/49.81	23.38/47.43	24.14/52.03	24.47/55.67	
90	9.05/8.21	22.47/37.64	22.13/37.29	23.36/48.71	23.07/45.18	22.36/32.14	23.09/40.17	23.43/49.33	

Table 3.5: Performance Comparison For Noise Standard Deviation of 30 (PSNR/SSIM)

Image	Noisy Image	NLM	NLM-PET	NLM-SAP	Fast NLM	PNLM	Intermediate	NLM-SVB
cameraman	18.56/31.05	27.30/76.68	27.02/76.47	27.70/81.27	27.61/79.94	27.59/78.40	27.70/80.07	27.78/82.48
house	18.56/26.00	29.62/76.89	29.50/76.95	29.99/79.78	29.81/78.14	29.57/77.83	30.38/80.01	30.53/81.54
barbara	18.59/37.40	27.26/76.00	26.91/75.13	27.90/80.57	27.50/79.84	27.61/75.51	27.54/79.30	27.94/81.35
couple	18.59/32.82	26.38/66.90	26.15/66.13	26.92/71.97	26.72/68.44	26.46/67.78	26.94/71.61	27.29/73.85



Figure 3.5: Estimation of the noiseless Cameraman (size 256×256) with noise standard deviation of 20 by NLM method. The PSNR is 29.17



Figure 3.6: Estimation of the noiseless Cameraman (size 256×256) with noise standard deviation of 20 by NLM-PET method. The PSNR is 28.65



Figure 3.7: Estimation of the noiseless Cameraman (size 256×256) with noise standard deviation of 20 by NLM-SAP method. The PSNR is 29.55



Figure 3.8: Estimation of the noiseless Cameraman (size 256×256) with noise standard deviation of 20 by Fast-NLM method. The PSNR is 29.49



Figure 3.9: Estimation of the noiseless Cameraman (size 256×256) with noise standard deviation of 20 by PNLM method. The PSNR is 29.51



Figure 3.10: Estimation of the noiseless Cameraman (size 256×256) with noise standard deviation of 20 by NLM-SVB method. The PSNR is 29.58



Figure 3.11: Estimation of the noiseless Barbara (size 512×512) with noise standard deviation of 30 by NLM method. The PSNR is 27.26



Figure 3.12: Estimation of the noiseless Barbara (size 512×512) with noise standard deviation of 30 by NLM-PET method. The PSNR is 26.91



Figure 3.13: Estimation of the noiseless Barbara (size 512×512) with noise standard deviation of 30 by NLM-SAP method. The PSNR is 27.90



Figure 3.14: Estimation of the noiseless Barbara (size 512×512) with noise standard deviation of 30 by Fast-NLM method. The PSNR is 27.50



Figure 3.15: Estimation of the noiseless Barbara (size 512×512) with noise standard deviation of 30 by PNLM method. The PSNR is 27.61



Figure 3.16: Estimation of the noiseless Barbara (size 512×512) with noise standard deviation of 30 by NLM-SVB method. The PSNR is 27.94

Chapter 4

Probabilistic Weighting BM3D

The signature method of grouping and collaborative filtering is the key of BM3D algorithm. It benefits from two principal properties of natural images very efficiently. The first one is the existence of similar patches within a neighborhood and the second one is the local correlation of pixel values in a single patch. They state that collaborative 3D filtering allows us to exploit both intra-patch correlation and inter-patch correlation based on the above two assumptions. This high degree of correlation leads to a very high sparse representation of the image in the transform domain. This sparseness can be interpreted as the fact that most of the image details are represented by few large coefficients while noise is the many small coefficients. This representation allows us to separate the noise from the true signal by applying thresholding on the coefficients in transform domain [28].

In this chapter we study BM3D in more details. We go over all parameters and design elements and try to discuss their optimality. Then we propose a new weighting scheme to improve the algorithm.

As it is been explained in Chapter 2, Block matching 3D transform domain collaborative filtering consists of two nearly identical transform domain denoising. The first one is hard thresholding coefficient by exploiting additional sparsity in coefficients of 3D blocks and creating an estimate of the noiseless image. The second step is using this estimate as pilot to perform wiener filtering again on 3D blocks. There are three different processes in these steps. One

is finding similar patches for each reference patch by a similarity measure close to Euclidean distance of patches and then transforming them into another domain (wavelet or discrete cosine transform). The other one is to shrinking the coefficients in the transformed 3D data to attenuate the noise (once by hard thresholding in the final step by wiener filter) and the last one is applying inverse 3D transform to the remaining coefficients, returning the 2D estimates of the grouped blocks to their original position and aggregating them according to a proper weighting scheme [1].

4.1 Grouping

As it has been explained in detail in the background chapter, the grouping step is performed to increase the sparsity of coefficients in transform domain. For each reference patch, they find similar patches in a search neighbourhood and group them in 3D stacks to denoise them using collaborative filtering. The similarity measure is the same as many other denoising algorithm and is based on Euclidean distance of vectorized patches.

4.1.1 Block Matching on Noisy Image

The formula for matching criteria is in (2.5). For high value of noises, Euclidean distance has been proved to be not working well which leads to inaccurate block matching and in the literatures is called "block effect" [29]. Due to this issue, authors propose a pre-filtering hard thresholding step denoted by operator Υ' of the coefficients for noise standard deviation greater than 40. They also consider two different threshold values for measure of similarity between patches. For sigma less than 40, this parameter τ_{match}^{ht} is set to 2500, for higher values of sigma it is set to 5000 due to more dissimilarity coming from additive noise. In [30], authors define three different levels of noise, low less than 30, medium between 30 and 50 and high more than 50 and adjust parameters of BM3D specifically the distance threshold for those three cases.

4.1.2 Block Matching on Basic Estimate

Block matching in the second step is not different than the first step 2.9. However it is done twice, once over the basic estimate image and the second time over the noisy observed image in exactly the same order. The similarity measure threshold τ_{match}^{wie} , is proposed to be 400.

There is always possibility to improve the accuracy of grouping by exploiting other similarity measures like the gradient of patches or average gray value as it is discussed in [27]. Or using noise invalidation approach similar to what is presented in the previous chapter to distinguish between dissimilar patches and patches that are similar in the noiseless version and are dissimilar only because of the additive noise. In this method similar to Chapter 3, one can find the distribution of distance of exactly the same patches after additive noise and compare its sorted version with distances of all candidate patches with the reference patch then eliminate unnecessary patches.

4.2 Denoising

The next step is collaborative filtering or 3D denoising. Transform domain denoising includes hard or soft thresholding of coefficients. Wiener filtering is another coefficient shrinkage method that performs well when there exists an estimate of the noiseless signal. The choice of 3D transform is a concern in this step. The DCT transform does not perform well for representing sharp edges and corners, while wavelet transform is not good for textures and smooth transitions. Thus authors suggest using two different transforms in hard thresholding and wiener filtering step to compensate each others flaws [31].

4.2.1 Hard Thresholding

Here the 3D noisy blocks are being denoised. A 3D transform over patches followed by hard thresholding coefficients is formalized in (2.7). The proposed 3D transform is a 2D wavelet on patches surface followed by a 1D Walsh-Hadamard transform on the third dimension. It seems that the choice of these transform does not affect the final result considerably [14]. However, the thresholding parameter λ_{3D} plays an important role in the amount of denoising or blurring.

Authors suggest the value of 2.7 for this parameter, it seems optimal but the reason is not clear. Changing this value leads to losing many dBs in the final PSNR.

4.2.2 Wiener Filtering

The role of wiener filtering is not presented in the original work and to the best of our knowledge it is not been discussed in any other analytical work. Therefore in this section we review this step in more details:

For each reference patch x_R we have a 3D block (consist of many similar patches) that we aim to apply collaborative wiener filtering to denoise them. That filter is a linear FIR filter which we name it \mathbf{Q}_{x_R} , a 3D matrix in spatial domain where:

$$\hat{\mathbf{Y}}_{x_R} = \mathbf{Q}_{x_R}^T \mathbf{Z}_{x_R} = (\mathbf{Z}_{x_R}^T \mathbf{Q}_{x_R})^T \quad (4.1)$$

The error would be as (4.2), another 3D matrix in spatial domain:

$$\mathbf{e}_{x_R} = \mathbf{Y}_{x_R} - \hat{\mathbf{Y}}_{x_R} \quad (4.2)$$

We aim to minimize the mean square error which is defined as:

$$C(\mathbf{Q}_{x_R}) = E[\mathbf{e}_{x_R} \mathbf{e}_{x_R}^T] = E[(\mathbf{Y}_{x_R} - \mathbf{Q}_{x_R}^T \mathbf{Z}_{x_R})(\mathbf{Y}_{x_R}^T - \mathbf{Z}_{x_R}^T \mathbf{Q}_{x_R})] \quad (4.3)$$

for simplification of notations:

$$\begin{aligned} C(Q) &= E[(Y - Q^T Z)(Y^T - Z^T Q)] \\ &= E(YY^T) - Q^T E(ZY^T) - E(YZ^T)Q + Q^T E(ZZ^T)Q \\ &= \sigma_y^2 - Q^T R_{ZY} - R_{ZY}^T Q + Q^T R_{ZZ}Q \end{aligned} \quad (4.4)$$

where R_{ZY} is the cross correlation matrix between observation and original signal and R_{ZZ} is the auto correlation matrix for the observation and σ_y^2 is the variance of original image. The mean square error is a second order equation. Therefore the equation has a global minimum

which can be identified by differentiating to Q and solving for zero:

$$\frac{dC(Q)}{dQ} = \frac{d}{dQ}(\sigma_y^2 - Q^T R_{ZY} - R_{ZY}^T Q + Q^T R_{ZZ} Q) = -2R_{ZY} + 2R_{ZZ}Q \quad (4.5)$$

The filter coefficients are easily solved. Wiener-Hopf equation:

$$Q = (R_{ZZ})^{-1} R_{ZY} \quad (4.6)$$

Since noise is additive and it is independent of signal ($R_{NY} = 0$) we can show:

$$Q = (R_{YY} + R_{NN})^{-1} R_{ZY} = (R_{YY} + \sigma^2)^{-1} R_{ZY} \quad (4.7)$$

In transform domain, auto correlation matrix will become power spectral density (PSD), so the shrinkage coefficient would be:

$$W = \tau_{3D}^{wie}(Q) = \frac{S_{YY}}{S_{YY} + \sigma^2} \quad (4.8)$$

Power Spectrum Estimation

The periodogram is an estimate of the spectral density of a signal. In practice, the periodogram is often computed from a finite-length Discrete Fourier Transform (DFT). The raw periodogram is not a good spectral estimate because of spectral bias and the fact that the variance at a given frequency does not decrease as the number of samples used in the computation increases. However, as the following proof shows, in the case of FIR signals (like our image blocks) we can calculate the exact spectrum density: Assume we have a 1D signal $x[n]$ which has values where $n = 0 : N$ and it is zero otherwise. We can vectorized any higher dimension signal and extend the theory. We define its auto correlation sequence as:

$$r_{xx}[k] = \sum_{n=0}^N x[n]x[n-k] \text{ for } k = -N : N \quad (4.9)$$

We take the DFT of the auto correlation and we get the power spectral density as:

$$S_{xx}[K] = \sum_{k=-N}^N r_{xx}[k] e^{\frac{-j2\pi Kk}{2N+1}} \text{ for } K = -N : N \quad (4.10)$$

If we substitute (4.9) in (4.10):

$$S_{xx}[K] = \sum_{k=-N}^N \sum_{n=0}^N x[n] x[n-k] e^{\frac{-j2\pi Kk}{2N+1}} \text{ for } K = -N : N \quad (4.11)$$

We define a new variable $l = n - k$:

$$S_{xx}[K] = \sum_{l=-N}^{2N} \sum_{n=0}^N x[n] x[l] e^{\frac{-j2\pi K(n-l)}{2N+1}} = \sum_{l=-N}^N x[l] e^{\frac{j2\pi Kl}{2N+1}} \sum_{n=-N}^N x[n] e^{\frac{-j2\pi Kn}{2N+1}} \quad (4.12)$$

And by definition of DFT:

$$S_{xx}[K] = X(K)X(-K) = |X(K)|^2 \quad (4.13)$$

Using a similar proof, we can show the equation holds for DCT too. Therefore, (4.8) will result in (2.10).

4.3 Aggregation

4.3.1 Weighting in Hard Thresholding

After denoising 3D groups each patch is transferred to its original location. Overlapping pixels needs to be aggregated in order to produce a unique estimate of the noisy pixel. Dabov et al. in [1] propose a scalar value for each 3D group which plays the role of the weight of each individual patch in that group on its original location. This aggregation step gives a basic estimate of the noisy image which by itself is better than some previously introduced methods.

The weighting scheme is based on the variance of patches in each 3D group in transform domain. This variance is either calculated by considering all coefficients of 3D group in transform

domain, or is estimated by simply comparing number or value of retrieved coefficients after denoising. For weighting in hard thresholding, estimating the variance by counting number of non-zero coefficients in each 3D group after hard thresholding is proposed as in (2.8).

4.3.2 Weighting in Wiener Filtering

There is a very similar approach in aggregation in Wiener filtering step and it is to estimate the variance of noiseless 3D stacks by sum of the squared coefficient values after shrinkage as in (2.12). It seems there are two main issues with this approach. The first one is that it seems low variance in the retrieved 3D stacks does not guarantee the accuracy of denoised version. For patches that are naturally with sharp edges and transitions, even in the perfect denoised scenarios this variance is still high. The second issue is ignoring the number of patches in each group. The method limits this number by maximum of 32 and a power of two. If two groups have the same variations, the number of retrieved coefficients in the bigger one would be larger consequently the assigned weight for all denoised patches in that group would be larger. This puts more weight on the patches coming from smoother regions which have more similar candidate patches and does not seem realistic. In the next section, we propose a new probabilistic framework to improve the weighting schema of BM3D.

4.4 Proposed Weighting

The idea behind this weightings scheme is that the impact of denoised 3D groups on each pixel can be approximated by the the probability that the entire 3D group are the noisy version of the same noiseless patches stacked in one group. It means there distance is only because of the additive noise. Following is the exact calculation of this probability:

If two patches centered at x and x_R are exactly the same their distance which is summation of some squared Gaussian random variables (additive noise values) follows a Chi-squared distribution. In the previous chapter and in [9] this probability which we denote it by $PS(D_{x,x_R})$

is shown to be:

$$PS(D_{x,x_R}) = \chi_{\eta_{x,x_R}}^2(D_{x,x_R}/\gamma_{x,x_R}) = \frac{(D_{x,x_R}/\gamma_{x,x_R})^{\eta_{x,x_R}/2-1} \exp(-D_{x,x_R}/2\gamma_{x,x_R})}{2^{\eta_{x,x_R}/2} \Gamma(\eta_{x,x_R}/2)} \quad (4.14)$$

where Γ denotes the Gamma function and:

$$\gamma_{x,x_R} = (2|Z_x| + |O_{x,x_R}|)/2|Z_x|; \quad (4.15)$$

$$\eta_{x,x_R} = |Z_x|/\gamma_{x,x_R} \quad (4.16)$$

and $|Z_x|$ is the number of pixels in patch Z_x and $|O_{x,x_R}|$ is the number of overlapping pixels between Z_x and Z_{x_R} . Since selection of patches in a group are independent from each other, their distances are also independent. So the probability of perfect similarity in a 3D group of noisy patches are equal to multiplication of those individual probabilities. We propose to assign the a scalar as weight for the whole group equals to this probability. To make this weight independent of the number of patches involved in the 3D stack, we normalize this probability by taking Nth root of it , where N is the number of patches in the 3D group. Note that this adaptation can happen in both two steps of BM3D. Consequently we have (2.8) and (2.12) modified as below:

$$w_{x_R}^{ht} \text{ or } w_{x_R}^{wie} = (\prod_{x \in S_{x_R}} PS(D_{x,x_R}))^{(\frac{1}{N})} \quad (4.17)$$

where \prod is the multiplication notation and PS is defined in (4.14). This weighting scheme has proved to perform better than the current variance based scheme. Next section shows the simulation results.

4.5 Simulation Result

Our test images are boat, man, cameraman, house, barbara and couple shown in Figure 3.4 the same as Chapter 3. The quality measurement criteria used for the performance evaluation is

PSNR [21].

Tables 4.1 and 4.2 shows the result for barbara and cameraman test images over a wide range of standard deviation of noise respectively. First column is the PSNR values for noisy images. Next column is the studied BM3D. Third column is a method introduced in [16] (Segmented-BM3D) . The idea is to choose the search neighbourhood a bit more wisely. They segment the image to two parts based on the intensity value of the pixels and for choose candidate similar pixels from those within the segment. It acts as a pre-filtering step over patches before calculating the distance this step does not improve the original method considerably since the intensity has already been considered in Euclidean distance somehow. However, it decrease the computational complexity by eliminating some unnecessary candidate patches. The next method Sigmoid-BM3D is introduced in [15]. Authors suggest a new thresholding function based on the mathematical sigmoid function instead of hard thresholding that causes sharp drop in coefficients in transform domain.

Our method PW-BM3D outperforms BM3D and Segmented-BM3D in most of the cases and it is compatible with Sigmoid-BM3D. The results for all other images, boat, man, house and couple are similar to what is shown for boat and man and are shown only for $\sigma = 30$ in Table 4.3. Figures 4.1- 4.8 show the visual quality of all studied methods.

Table 4.1: Performance Comparison Over Test Image of Barbara (PSNR)

σ	Noisy Image	BM3D	Segmented-BM3D	Sigmoid-BM3D	PW-BM3D
10	29.03	34.81	34.78	34.88	34.91
30	18.59	29.61	29.60	29.63	29.64
50	14.82	27.18	27.23	27.30	27.34
70	11.70	25.46	25.53	25.61	25.60
90	10.09	23.90	24.01	24.18	24.08



Figure 4.1: Estimation of the noiseless House (size 256×256) with noise standard deviation of 20 by BM3D method. The PSNR is 33.53



Figure 4.2: Estimation of the noiseless House (size 256×256) with noise standard deviation of 20 by Segmented-BM3D method. The PSNR is 33.50



Figure 4.3: Estimation of the noiseless House (size 256×256) with noise standard deviation of 20 by Sigmoid-BM3D method. The PSNR is 33.58



Figure 4.4: Estimation of the noiseless House (size 256×265) with noise standard deviation of 20 by PW-BM3D method. The PSNR is 33.63



Figure 4.5: Estimation of the noiseless Boat (size 512×512) with noise standard deviation of 30 by BM3D method. The PSNR is 29.03



Figure 4.6: Estimation of the noiseless Boat (size 512×512) with noise standard deviation of 30 by Segmented-BM3D method. The PSNR is 29.06



Figure 4.7: Estimation of the noiseless Boat (size 512×512) with noise standard deviation of 30 by Sigmoid-BM3D method. The PSNR is 29.05



Figure 4.8: Estimation of the noiseless Boat (size 512×512) with noise standard deviation of 30 by PW-BM3D method. The PSNR is 29.14

Table 4.2: Performance Comparison Over Test Image of Cameraman (PSNR)

σ	Noisy Image	BM3D	Segmented-BM3D	Sigmoid-BM3D-	PW-BM3D
10	28.98	33.99	33.96	33.94	34.12
30	18.56	28.47	28.46	28.45	28.63
50	14.01	26.12	26.19	26.15	26.30
70	11.54	24.54	24.61	24.60	24.72
90	9.89	23.39	23.51	23.58	23.64

Table 4.3: Performance Comparison For Noise Standard Deviation of 30 (PSNR)

Image	Noisy Image	BM3D	Segmented-BM3D	Sigmoid-BM3D	PW-BM3D
boat	18.59	29.03	29.00	29.05	29.14
man	18.59	28.87	28.90	28.93	29.00
house	18.56	31.91	31.88	31.99	32.05
couple	18.59	28.75	28.81	28.84	28.92

Chapter 5

Conclusions and Future Work

In this thesis, we studied the two most well-known and used image denosing techniques: Nonlocal Means (NLM) and Block Matching 3D Transform Domain Collaborative Filtering (BM3D). We started Chapter 1 with an overview of these two methods and discussed some other works that have been done on their improvements. In Chapter 2 we presented a brief overview of NLM and BM3D steps. The noise invalidation technique for data denoising was reviewed, and our criteria measures for performance evaluations, SSIM and PSNR, were discussed.

In Chapter 3 we introduced Similarity Validation Based Nonlocal Means technique which is an improvement over NLM and showed the simulation results. By adding an additional pre-processing stage in form of a hard thresholding, we have improved the performance of the traditional nonlocal means. This pre-processing step attempts to eliminate dissimilar patches prior to the weighting process. As our simulation result shows this step can eliminate about 60% of the patches that are used in traditional nonlocal means. As it is shown this percentage is lower for the flat neighbourhoods and higher for the neighbourhood with fine details. Our proposed method (NLM-SVB) considers the exact distribution of similar patches distances in both the hard thresholding step and weighting process. In addition, we use the same distribution in the weighting process. Our simulation results illustrate the advantages of the proposed method over the traditional nonlocal means, probabilistic nonlocal means, fast nonlocal means, NLM with early termination and shape adaptive NLM that are variations of NLM. The combination

of the noise invalidation technique with those variations of NLM can be considered for a future works. For example, pre-filtering techniques based on the gradient or average gray-value of the patches can be added to the noise invalidation technique to completely remove unnecessary patches.

In Chapter 4 we studied BM3D in details. This algorithm has a large number of parameters and design elements and is relatively more complex than other existing methods. We had a thorough analysis on the theoretical aspects of BM3D, specifically we studied the estimation of power spectrum of basic estimated 3D groups that are used in calculation of wiener filter coefficients and were not explained in the original paper and, to the best of our knowledge, in any other analytical works in the field of image processing. As explained in details there are many parameters that their optimal value are found by trial and error and the algorithm can be sensitive to them. Future work can analyze and discover theoretical reasoning behind the optimality of those parameters.

Then, we introduced the two main issues with the current weighting scheme which leads to a bias scheme and puts more weights on smoother patches. We suggested a new weighting scheme based on the probability distribution of the distance of noisy similar patches. Our simulation results confirmed superiority of our approach (PW-BM3D) over the traditional BM3D for an average of about 0.2dB in PSNR. There are still some aspects of BM3D that can be improved, specifically for generating the estimate in the first step. Some iterative methods can be considered to improve the basic estimate that will be used as the pilot for wiener filtering step. Also some denoising techniques in transform domain other than hard thresholding can be explored to retrieve the data coefficient of 3D blocks in a more efficient way in the future.

Related Publications

Mina Sharifymoghaddam, Soosan Beheshti, Pegah Elahi, and Masoud Hashemi. Similarity Validation Based Nonlocal Means Image Denoising. Signal Processing Letters, IEEE,22(12):21852188, 2015

Appendix 1

Calculation of boundaries for validating patches

The idea used for elimination of dissimilar patches is inspired by noise invalidation thresholding [3]. For reference patch P_i , in search neighbour S_i , we denote the non overlapping patches in S_i by \bar{S}_i . If P_i and P_j are similar and both are corrupted by AWGN, we denote their distance by $d_{i,j}^n$. We also denote the set of all $d_{i,j}^n$ s in \bar{S}_i by $\{d_{i,j}^n\}$. Defining the following function for any given z :

$$g(z, d_{i,j}^n) = \begin{cases} 1 & \text{if } d_{i,j}^n \leq z \\ 0 & \text{otherwise} \end{cases} \quad (1.1)$$

the expected value of this random variable is:

$$\begin{aligned} E(g(z, D_{i,j}^n)) &= 1 \times Pr(D_{i,j}^n \leq z) + 0 \times Pr(D_{i,j}^n > z) \\ &= Pr(D_{i,j}^n \leq z) = F(z) \end{aligned} \quad (1.2)$$

where F is the cumulative distribution function (CDF) of the random variable $D_{i,j}^n$ (we denote $d_{i,j}^n$ as a sample of random variable $D_{i,j}^n$). Note that F is a Chi-squared distribution of order

$|P_i|$ calculated in [9]. With variance of this random variable we have:

$$\begin{aligned} Var(g(z, D_{i,j}^n)) &= E(g(z, D_{i,j}^n)^2) - E(g(z, D_{i,j}^n))^2 \\ &= F(z) - F(z)^2 = F(z)(1 - F(z)) \end{aligned} \quad (1.3)$$

where:

$$\begin{aligned} E(g(z, D_{i,j}^n)^2) &= 1 \times Pr(D_{i,j}^n \leq z) + 0 \times Pr(D_{i,j}^n > z) \\ &= Pr(D_{i,j}^n \leq z) = F(z) \end{aligned} \quad (1.4)$$

and $E(g(z, D_{i,j}^n))$ is defined in (1.2). The following function:

$$\bar{g}(z, \{d_{i,j}^n\}) = \frac{1}{|\bar{S}_i|} \sum_{j \in \bar{S}_i} g(z, d_{i,j}^n) \quad (1.5)$$

calculates the number of pixels with patch distance less than z ; equivalently when sorting $d_{i,j}^n$ s, if m number of them is less than or equal to z , then $\bar{g}(z, \{d_{i,j}^n\}) = m/|\bar{S}_i|$. This function represents the index of sorted version of $d_{i,j}^n$ s [3]:

$$E(\bar{g}(z, \{D_{i,j}^n\})) = E(g(z, D_{i,j}^n)) = F(z) \quad (1.6)$$

$$\begin{aligned} Var(\bar{g}(z, \{D_{i,j}^n\})) &= \frac{1}{|\bar{S}_i|} Var(g(z, D_{i,j}^n)) \\ &= \frac{1}{|\bar{S}_i|} F(z)(1 - F(z)) \end{aligned} \quad (1.7)$$

Note that $|\bar{S}_i|$ is a considerably large number. For example, if the search neighbourhood has size of 21×21 and patches are 5×5 squares, this number is 360. This will cause the variance of $\bar{g}(z, \{D_{i,j}^n\})$ to be considerably less than its mean. In addition, \bar{g} in (1.5) is a summation of $|\bar{S}_i|$ numbers random variables. Consequently, using Central Limit Theorem we can estimate this random variable with a Gaussian distribution. Therefore the following probabilistic boundaries

hold for $d_{i,j}$:

$$Pr(L(i) < \bar{g}(d_{i,j}^n, \{D_{i,j}^n\}) < U(i)) \approx erf(\lambda/\sqrt{2}) \quad (1.8)$$

where $U(i)$ and $L(i)$ are:

$$E(\bar{g}(d_{i,j}^n, \{D_{i,j}^n\})) \pm \lambda \sqrt{Var(\bar{g}(d_{i,j}^n, D_{i,j}^n))} \quad (1.9)$$

For the choice of λ , the conventional three times standard deviations rule can be used that results in a confidence probability of 99.8%. To implement this approach, first upperbound and lowerbound in (1.9) are pre-calculated as a function of σ . We then compare whether the available values of $\bar{g}(d_{i,j}, \{d_{i,j}\})$ are within these boundaries. The threshold is the value of the point from which this number gets out of the boundaries. From that index j^* , the structure of these distances do not follow that of due to pure AWGN only. Therefore, P_j s for $j > j^*$ represent dissimilar patches. Middle plot in Figure 3.2 shows an example of such j^* to be 135.

References

- [1] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” *Image Processing, IEEE Transactions on*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [2] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in *Computer Vision and Pattern Recognition. IEEE Computer Society Conference on*, pp. 60–65, 2005.
- [3] S. Beheshti, M. Hashemi, X.-P. Zhang, and N. Nikvand, “Noise invalidation denoising,” *Signal Processing, IEEE Transactions on*, vol. 58, no. 12, pp. 6007–6016, 2010.
- [4] M. C. Motwani, M. C. Gadiya, R. C. Motwani, and F. C. Harris, “Survey of image denoising techniques,” in *Proceedings of GSPX*, pp. 27–30, 2004.
- [5] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Computer Vision. Sixth International Conference on*, pp. 839–846, 1998.
- [6] P. Milanfar, “A tour of modern image filtering: New insights and methods, both practical and theoretical,” *Signal Processing Magazine, IEEE*, vol. 30, no. 1, pp. 106–128, 2013.
- [7] M. Lebrun, M. Colom, and B. Antoni, “Secrets of image denoising cuisine,” *Acta Numerica*, vol. 21, pp. 475–576, 2012.
- [8] C.-A. Deledalle, V. Duval, and J. Salmon, “Non-local methods with shape-adaptive patches (NLM-SAP),” *Journal of Mathematical Imaging and Vision, Springer*, vol. 43, no. 2, pp. 103–120, 2012.

- [9] Y. Wu, B. Tracey, P. Natarajan, and J. P. Noonan, "Probabilistic non-local means," *Signal Processing Letters, IEEE*, vol. 20, no. 8, pp. 763–766, 2013.
- [10] R. Vignesh, B. T. Oh, and C.-C. J. Kuo, "Fast non-local means (NLM) computation with probabilistic early termination," *Signal Processing Letters, IEEE*, vol. 17, no. 3, pp. 277–280, 2010.
- [11] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.
- [12] D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.
- [13] R. C. Gonzalez, *Digital image processing*. Pearson Education India, 2009.
- [14] M. Lebrun, "An analysis and implementation of the BM3D image denoising method," *Image Processing On Line*, pp. 175–213, 2012.
- [15] M. Poderico, S. Parrilli, G. Poggi, and L. Verdoliva, "Sigmoid shrinkage for BM3D denoising algorithm," in *Multimedia Signal Processing (MMSP), IEEE International Workshop on*, pp. 423–426, 2010.
- [16] H. Rehan Afzal, H. Ali, and M. Asif, "Improved BM3D by using region growing segmentation & Sobel edge technique," in *Image and Signal Processing (CISP), IEEE International Congress on*, pp. 164–168, 2014.
- [17] D. Raghuvanshi, D. Jain, and P. Jain, "Performance Analysis of Non Local Means Algorithm for Denoising of Digital Images," *International Journal*, vol. 3, no. 1, pp. 94–100, 2013.
- [18] C. Knaus and M. Zwicker, "Dual-domain image denoising," in *Image Processing, IEEE International Conference on*, pp. 440–444, 2013.
- [19] W. Liu, Y. Zhu, T. Liu, M. Rong, and H. Zhang, "Analysis and architecture design of aggregation in BM3D," in *ASIC (ASICON), IEEE International Conference on*, pp. 703–706, 2011.

- [20] S. Beheshti, N. Nikvand, and X. N. Fernando, "Soft thresholding by noise invalidation," in *Proceedings of the 24th Biennial Symposium on Communications*, pp. 235–238, 2008.
- [21] D. S. Turaga, Y. Chen, and J. Caviedes, "No reference PSNR estimation for compressed pictures," *Signal Processing: Image Communication, Elsevier*, vol. 19, no. 2, pp. 173–184, 2004.
- [22] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.
- [23] M. Sharifymoghaddam, S. Beheshti, P. Elahi, and M. Hashemi, "Similarity Validation Based Nonlocal Means Image Denoising," *Signal Processing Letters, IEEE*, vol. 22, no. 12, pp. 2185–2188, 2015.
- [24] D. Brunet, E. R. Vrscay, and Z. Wang, "The use of residuals in image denoising," in *Image Analysis and Recognition*, pp. 1–12, 2009.
- [25] S. Kumar, "Image denoising based on non-local means filter and its method noise thresholding," *Signal, image and video processing, Springer*, vol. 7, no. 6, pp. 1211–1227, 2013.
- [26] H. Zhong, C. Yang, and X. Zhang, "A new weight for nonlocal means denoising using method noise," *Signal Processing Letters, IEEE*, vol. 19, no. 8, pp. 535–538, 2012.
- [27] M. Mahmoudi and G. Sapiro, "Fast image and video denoising via nonlocal means of similar neighborhoods," *Signal Processing Letters, IEEE*, vol. 12, no. 12, pp. 839–842, 2005.
- [28] M. M. Hasan, *Adaptive Edge-guided Block-matching and 3D filtering (BM3D) Image Denoising Algorithm*. PhD thesis, University of Western Ontario, 2014.
- [29] L. Dai, Y. Zhang, and Y. Li, "BM3D Image Denoising Algorithm with Adaptive Distance Hard-threshold," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 6, no. 6, pp. 41–50, 2013.

-
- [30] O. Pakdelazar and G. Rezai-rad, "Improvement of BM3D algorithm and Employment to satellite and CFA images Denoising," *International Journal of Information Sciences and Techniques (IJIST)*, vol. 1, no. 3, pp. 23–33, 2011.
- [31] R. K. Rai, J. Asnani, and T. Sontakke, "Review of shrinkage techniques for image denoising," *International Journal of Computer Applications*, vol. 42, pp. 13–6, 2012.