Improvement of Image Denoising with Interpolation and RAISR

May 2021

Theingi Zin

A Thesis for the Degree of Ph.D. in Engineering

Improvement of Image Denoising with Interpolation and RAISR

May 2021

Graduate School of Science and Technology

Keio University

Theingi Zin

Contents

1 Introduction **1**

1.1 Background 1

1.2 Types of Image Noise 5

1.2.1 Gaussian Noise 6

1.2.2 Impulse Noise 6

1.2.3 Shot Noise or Poisson Noise 7

1.2.4 Speckle Noise 8

1.3 Causes of Noise in Digital Images 9

1.4 Problem Statement 10

1.5 Research Objectives 12

1.6 Research Methodology 13

1.7 Organization of the Dissertation 14

2 Related Works **15**

2.1 Nonlocal-based Image Denoising Methods 15

2.1.1 Nonlocal Means 16

2.1.2 BM3D Filtering 17

2.1.3 Learned Simultaneous Sparse Coding 19

2.1.4 Nonlocally Centralized Sparse Representation 21

2.1.5 WNNM Filtering 23

2.2 CNN-based Image Denoising Methods 25

2.2.1 Residual Learning of Deep CNN for Image Denoising 26

2.2.2 Fast and Flexible Denoising Convolutional Neural Network: FFDNet 27

3 Image Denoising Using Interpolation **29**

3.1 Introduction 29

3.2 Mixed Noise Model 30
3.3 DWM Filter 31
3.3.1 Detection Process 31
3.3.2 Removal Process 33
3.4 Interpolation 34
3.4.1 Local Functions Estimation 34
3.4.2 Weighted Mean Calculation 37
3.5 Proposed Method 38
3.5.1 Interpolation in Mixed Noise Removal 39
3.5.2 Impulse Noise Removal 42
3.5.3 Down-sampling 43
3.5.4 Gaussian Noise Removal 43
3.5.5 Re-detect Process and BM3D 44
3.6 Experimental Results 44
3.6.1 Parameters Setting 45
3.6.2 Quantitative and Visual Evaluation 45
3.7 Summary 53

4 Nonlocal Based Image Denoising with RAISR 63
4.1 Introduction 63
4.2 RAISR 66
4.2.1 Calculation of Hash-table Keys 67
4.2.2 Global Filter Learning 68
4.2.3 Detection of Structure Deformations: CT Transform 69
4.3 Proposed Method 70
4.3.1 Noise Removal 71
4.3.2 Improvement of RAISR in Denoising 72
4.3.3 Geometric Conversion 74
4.3.4 Reduction of the Classes for the Gradient Strength 74
4.4 Experimental Results 75

4.4.1 Parameters Setting 75
4.4.2 Quantitative and Visual Evaluation 76
4.4.3 Experiments on Various Datasets 79
4.4.4 Comparison of Performance Evaluation on Different Training Sets 81
4.4.5 Effect of CT in Image Denoising 82
4.4.6 Performance Comparison between RAISR and IRAISR in Image Denoising 83
4.5 Summary 84

5 Conclusion **100**

5.1 Overall Conclusion of the Dissertation 100
5.2 Limitations 102
5.3 Further Extensions 103

List of Figures

1.1 General block diagram of Gaussian noise removal 2
1.2 General block diagram of impulse noise removal 3
1.3 General block diagram of mixed noise removal 5
1.4 Generation of Gaussian noise in *Lena* image with $\sigma = 30$ 6
1.5 Generation of Salt-and-Pepper noise with $p = 5\%$ and RVIN with $p = 25\%$ in *Airplane* image 8
1.6 Generation of Speckle noise in *Pepper* image with $\sigma = 15$ 9
1.7 Causes of noise in digital images at different conditions 10
1.8 Denoised result of *Barbara* corrupted by mixed Gaussian and random-valued impulse noise ($\sigma = 25$ and $p = 25\%$) 11
1.9 Denoised result of *Butterfly* corrupted by AWGN ($\sigma = 60$) 12

2.1 Block diagram of BM3D 17
2.2 Denoising model of WNNM 24

3.1 Four directions for impulse detection 32
3.2 Interpolation scheme to calculate the local function Γ_i centered at p_i. 36
3.3 Visual comparison on the effect of second term in the strong edge image 37
3.4 Block diagram of the proposed method 39
3.5 Reduction of impulse noise rate in 100×100-sized *Test* image corrupted by $p = 20\%$ and $p = 40\%$ 41
3.6 Reduction of impulse noise rate in 512×512-sized *Lena* image corrupted by $p = 20\%$ and $p = 40\%$ 41
3.7 Restoration results of *Airplane* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 15$ and $p = 20\%$ 52
3.8 Restoration results of *Barbara* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 25$ and $p = 25\%$ 53

3.9 Restoration results of *Boat* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 20$ and $p = 15\%$ 54
3.10 Restoration results of *Goldhill* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 20$ and $p = 10\%$ 55
3.11 Restoration results of *Houses* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 10$ and $p = 10\%$ 56
3.12 Restoration results of *Lena* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 25$ and $p = 20\%$ 57
3.13 Restoration results of *Pepper* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 15$ and $p = 15\%$ 58
3.14 Restoration results of *Bridge* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 10$ and $p = 25\%$ 59
3.15 Restoration results of *Mandrill* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 5$ and $p = 20\%$ 60
3.16 Restoration results of *Milkdrop* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 25$ and $p = 5\%$ 61
3.17 Restoration results of *Sailboat* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 20$ and $p = 25\%$ 62

4.1 Census transform(CT) 69
4.2 Learning phase of the proposed method 70
4.3 Testing phase of the proposed method 71
4.4 Improved RAISR design for the proposed method based on the pairs of the patches from the denoised image and the pixels from the clean image 73
4.5 Geometric conversion based on each gradient type 74
4.6 Visual comparison of *Butterfly* image corrupted by $\sigma = 30$ 85
4.7 Visual comparison of *Peppers* image corrupted by $\sigma = 50$ 86
4.8 Visual comparison of *Airplane* image corrupted by $\sigma = 40$ 87
4.9 Visual comparison of *Cameraman* image corrupted by $\sigma = 60$ 88

4.10 Visual comparison of *Couple* image corrupted by $\sigma = 30$ 89
4.11 Visual comparison of *Boat* image corrupted by $\sigma = 50$ 90
4.12 Visual comparison of *Barbara* image corrupted by $\sigma = 40$ 91
4.13 Visual comparison of *Lena* image corrupted by $\sigma = 60$ 92
4.14 Visual comparison of *Montage* image corrupted by $\sigma = 20$ 93
4.15 Visual comparison of *Man* image corrupted by $\sigma = 20$ 94
4.16 Visual comparison of one image from Kodak dataset corrupted by $\sigma = 30$ 95
4.17 Visual comparison of one image from BSD68 dataset corrupted by $\sigma = 50$ 96
4.18 Visual comparison of one image from BSD68 dataset corrupted by $\sigma = 30$ 97
4.19 Visual comparison of one image from Kodak dataset corrupted by $\sigma = 50$ 98
4.20 Visual comparison of one image from Set12 dataset corrupted by $\sigma = 30$ 99

5.1 Learning phase in local-based denoising method 104
5.2 Testing phase in local-based denoising method 105

List of Tables

3.1 Accuracy of impulse noise detection due to the effect of interpolation 42

3.2 Restoration results for mixed noise removal in terms of PSNR [dB] ($\sigma = 5$) 46

3.3 Restoration results for mixed noise removal in terms of PSNR [dB] ($\sigma = 10$) 47

3.4 Restoration results for mixed noise removal in terms of PSNR [dB] ($\sigma = 15$) 48

3.5 Restoration results for mixed noise removal in terms of PSNR [dB] ($\sigma = 20$) 49

3.6 Restoration results for mixed noise removal in terms of PSNR [dB] ($\sigma = 25$) 50

3.7 The number of un-detected and miss-detected pixels in *Barbara* image corrupted by mixed noise composed of AWGN and RVIN 50

3.8 The number of un-detected and miss-detected pixels in *Lena* image corrupted by mixed noise composed of AWGN and RVIN 51

4.1 Average PSNR comparison for upscaling by a factor of 2 over Set5. . 75

4.2 The quantitative comparison on average PSNR (dB) over 11 widely used test images 77

4.3 Execution time comparison for 256×256-sized images 78

4.4 Execution time comparison for 512×512-sized images 78

4.5 Comparison of PSNR values and runtime on average over various datasets 80

4.6 Performance evaluation on different training datasets 82

4.7 Performance comparison of image denoising based on CT over 11 widely used images 82

4.8 Impact of RAISR and IRAISR in image denoising over 11 widely used images 83

Acknowledgements

First of all, I would like to express my deepest and sincere gratitude to my supervisor, Professor Masaaki Ikehara, Professor from the Department of Electronics and Electrical Engineering, Graduate School of Science and Technology, Keio University, Japan, for his invaluable suggestion, constructive discussion and continuous guidance throughout my doing research and writing this thesis under his supervision in Japan. He is a brilliant and kind-hearted supervisor who I have ever met in my life. I got a great privilege in his lab to conduct the researches relevant to image denoising, image interpolation and super-resolution that are indispensable steps for low level vision applications as well as an opportunity to know how to do research well. Honestly, I could not accomplish my research and dissertation without his constructive suggestion and support.

I would like to deliver my heartfelt thanks to my respected thesis committee members, Professor Yukitoshi Sanada, Professor Toshiyuki Tanaka and Associate Professor Ryogo Kubo for their insightful comments and suggestions in my Ph.D defense exam to improve and strengthen my dissertation.

I am genuinely thankful to Associate Professor Takuro Yamaguchi for his precious advice about my research and supports in writing paper during my Ph.D study in Ikehara Lab.

I am deeply indebted to Japan International Cooperation Agency (JICA) project for ASEAN University Network/Southeast Asia Engineering Education Development Network (AUN/SEED) Net and Keio Leading-edge Laboratory (KLL) grant for financial supports to perform research well during my Ph.D training.

I would like to appreciate my heartfelt thanks to my Minister, Rectors, Head of Department and officials from the Department of Electronic Engineering, Mandalay Technological University, Department of Higher Education, Ministry of Education, Myanmar, for their permission to study my Doctoral Degree Program in Japan for

about three and a half years.

My sincere also thanks to my teacher Professor Dr.Hla Myo Tun, Pro-rector, Faculty of Electrical and Computer Engineering, Yangon Technological University, Myanmar who enthusiastically persuaded me to pursue my doctoral degree abroad during my work under his leadership in Mandalay Technological University.

I would like to offer my deepest thanks to Dr.Aung Soe Khaing, Professor, from the Department of Electronic Engineering, Technological University (Mandalay), Mandalay, Myanmar who checked my first paper and gave me inestimable suggestions to improve my manuscript.

I send my warmest regards to my parents, U Kaung Sein and Daw Khin Mar Kyi for their continuous supports and encouragement with unconditional love to be strong physically and mentally whenever I face many difficulties since birth until now and I feel very depressed during my study abroad. I am specially thankful to my two elder brothers and one younger sister who encouraged and supported me to continue my career happily.

I really appreciate all my coordinators, Nishigaki Tomomi, Keiko Fujino and Misako Fukuda who helped me everything from the day when I arrive in Japan till now, and JICA staffs from JICA Yokohama International Center for everything what they did for me throughout my stay in JICA center. I had a great memory and wonderful event because of their caring and hospitality.

I truly want to say million thanks to all my beloved relatives, teachers, friends and students from Myanmar who always believed in my efforts and delivered me warm wishes via social media to overcome everything during COVID-19 pandemic.

Finally, I would like to express my special thanks to all my seniors including Dr. Takanori Fujisawa and Thai friend, Sawiya Kiatpapan, and juniors from Ikehara Lab for their help about my research within my Ph.D journey. This is dedicated to all of them.

Abstract

Digital images are frequently contaminated by noise due to different sources such as transmission errors, malfunctioning pixel elements in the camera sensors, faulty memory location and timing errors in analog-to-digital conversion. The presence of noise in digital images leads to some undesirable effects such as image degradation and distortion of some important image features. Therefore, image denoising has recently become essential in many subsequent image processing applications as a pre-processing step. The aim of image denoising is to efficiently attenuate the corrupted noise and preserve the image details such as edges and textures in the image. Sometimes, digital images are degraded by single noise as well as more than one type of noise. Denoising methods may be different depending upon the types of noise because the characteristics of noise and filtering approaches are dissimilar to each other. The objectives of this thesis are to enhance the quantitative performance of mixed-noise removal method with interpolation approach while preserving the image details and improve the denoising performance of Gaussian noise removal method via Improved Rapid and Accurate Image Super-Resolution (IRAISR) with the reduced number of filters without sacrificing salient image features.

The suppression of mixed-noise composed of Additive White Gaussian Noise (AWGN) and Random-Valued Impulse Noise (RVIN) is considered in this thesis. There are mainly two steps in the removal of mixed noise. The first step is to achieve the denoised image by integrating Interpolation, Directional Weighted Median (DWM) filter, downsampling and Block Matching and 3D (BM3D) filtering. The second step is to obtain the restored image by combining re-detect process which is thresholding on the absolute difference between the input noisy image and the pre-estimated image from the first step, and BM3D. Even though some conventional mixed-noise removal methods can successfully filter the noise, some image details are lost due to the miss detection of the image details as the impulse noise. In order to overcome this issue, the interpolation technique based on multi-surface fitting for

single image is added before the detection of impulse noise in DWM filter. And then, it is also necessary to down-sample the interpolated DWM output because of the effect of interpolation. As most mixed-noise removal methods are detection-based, the detection of impulse noise in eliminating the mixed-noise plays a vital role to be considered. The addition of interpolation before DWM filter is very efficient in detecting the impulse noise to achieve an excellent denoising performance.

On the other hand, the elimination of Gaussian noise can be improved by employing IRAISR as a post-processing step to prevent from distortion of some image structures because of the deterioration of high frequency components in the existing noise removal methods. In this method, the two steps are basically structured namely: learning phase and testing phase. The filters are learned from the image pairs between the patches extracted from the images denoised by nonlocal-based benchmark methods such as BM3D and Weighted Nuclear Norm Minimization (WNNM), and the pixels from Ground truth by eigen-analysis in the learning phase. The filtered image can be obtained by applying the pre-learned filters which are the reduction to 18 filters in the hash classes by two improvements including geometric conversion for the gradient angle and the minimization of the classes for the gradient strength to the patches extracted from the denoised image in the testing phase. Moreover, the Census transform (CT) is also utilized by blending the image attenuated by Gaussian noise removal techniques and the filtered output to restore the local structures of the image within a wide range of frequencies.

Chapter 1

Introduction

1.1 Background

Nowadays, digital images can be easily obtained according to the advance in camera technology. However, digital images are inevitably corrupted by noise during the process of image acquisition and transmission. In particular, they are degraded by only single noise and the images with a variety of different noise are frequently contributed in the real applications. In order to get high quality noise-free images, image denoising becomes not only a highly demanded area of research in the image processing fields but also an indispensable pre-processing step for many low-level vision applications. Two types of commonly encountered noises in the real-world are additive Gaussian noise and impulse noise. Practically, impulse noise is occurred by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or transmission in a noisy channel and Gaussian noise is usually generated by the thermal motion of electron in camera sensors during the process of image acquisition.

When Gaussian noise is added to an image, a corrupted pixel value is the sum of the original pixel value and the Gaussian distributed random noise. Especially, Additive White Gaussian Noise (AWGN) is the noise which is found to be additive in nature with uniform power in the whole bandwidth with Gaussian probability distribution. AWGN is a good approximation of noise in the real world and many Gaussian noise removal methods have been successfully proposed since a few decades. The general block diagram of Gaussian noise removal is illustrated in Fig. 1.1. Some denoising methods are locally-based and some are nonlocally. Wiener filtering [1, 2], bilateral filtering which is well-known as edge-aware filtering [3], total variation (TV)-based methods [4–10], nonlocal-based image denoising methods [11–30], sparse

representation (SR) based methods [31–39], low-rank minimization based methods [40–45], dual-domain based image denoising methods [46,47] and convolutional neural network (CNN)-based denoising methods [48–51] have been recently presented to efficiently remove Gaussian noise. Many conventional methods utilize the frequency property to suppress AWGN. For instance, BM3D [24] is a classical and effective denoising method by applying a grouping of similar 2-D fragments extracted from the noisy image and collaborative filtering of grouped blocks. In contrast, SAIST [40] and WNNM [30] use singular value decomposition and low rank approximation without utilizing frequency properties. Zhang *et al.* [52] grouped the similar patches into a matrix and applied principal component analysis (PCA) to remove AWGN.

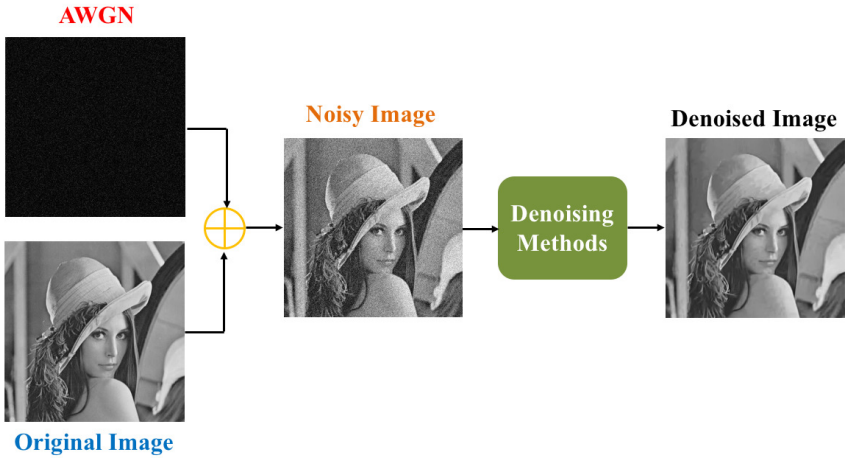


Figure 1.1: General block diagram of Gaussian noise removal

On the other hand, when impulse noise is added to an image, a corrupted pixel is replaced by a fixed-value (salt-and-pepper noise) or random value. Salt-and-pepper noise is the noise which replaces the dark pixel in the bright region and bright pixel in the dark region. Random-Valued Impulse Noise (RVIN) is the impulse noise which replaces pixels by random values within the dynamic range of images without changing the remaining pixels. Figure. 1.2 shows the general block diagram of impulse noise removal. Recently, many impulse noise removal methods [53–69] have been published. The most classical and simplest filter for removal of impulse noise is median filter [53] which is typically nonlinear filter. Although the median filters can attenuate the impulse noise, some local structures of the image are deteriorated when the noise density

level is high. Hence, the modification of median filters such as the weighted median filter [54], the multistate median (MSM) filter [55], the center-weighted median filter [56] and the stack filter [57] are proposed to improve the denoising performance.

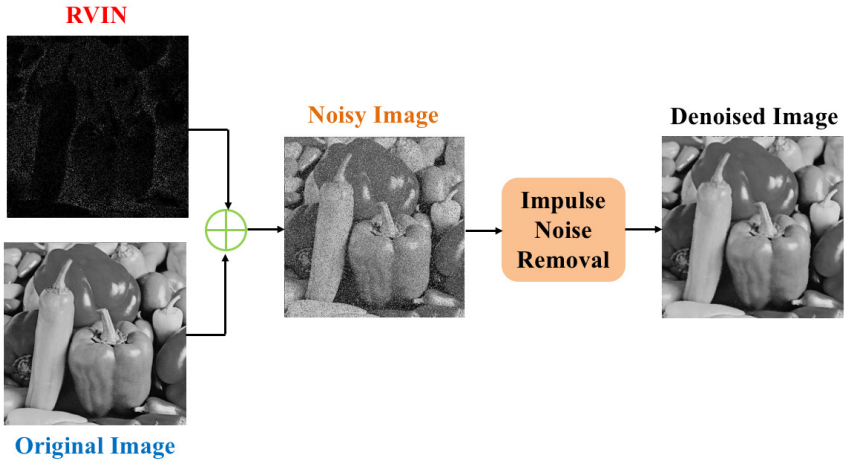


Figure 1.2: General block diagram of impulse noise removal

These aforementioned median filters tend to cause oversmoothing the image local structures because they don't consider the identification of image pixels which are noisy or not. Therefore, many improved denoising techniques for impulse noise which are the integration of impulse noise detection mechanisms and the median filtering such as switching median filter [58], MSM filter [59], tristate median (TRI) filter [60], adaptive center-weighted median (ACWM) filter [61], the pixel-wise median absolute deviation (PWMAD) filter [62], the adaptive switching median (ASWM) filter [63], a directional weighted median (DWM) filter [64], Luo-iterative median filter [65], the conditional signal-adaptive median (CSAM) filter [66], and the rank-ordered logarithmic difference edge-preserving regularization filter (ROLD-EPR) [67] are explored.

In addition, the genetic programming filter [70] is also developed to remove IN by switching between two IN detectors and their associated estimators. Generally speaking, there are typically two processes in detection-based impulse noise removal methods namely: noise detection and noise removal. For instance, the DWM filter [64] first detects noisy pixels by the local difference between a target pixel and its neighboring pixels, and then removes the noisy pixels by the directional weighted

median filter. Hence, the detection of impulse noise plays an important factor in the removal of impulse noise to achieve an enhanced denoising performance.

Sometimes, the images are contaminated by different types of noise. Since the characteristics of noises are significantly different, it is necessary to select an effective algorithm to remove the mixed-noise. The general block diagram of mixed noise removal is shown in Fig. 1.3. A few mixed noise removal methods [71–78] are efficiently developed to suppress the combination of RVIN and AWGN. In [71], the pixels degraded by RVIN are firstly detected and then removed by computing an optimization problem using the pixels unaffected by RVIN. [72] addresses a denoising framework for mixed noise which is a detection mechanism based on Robust Outlyingness Ratio (ROR) incorporating with nonlocal similarity models to achieve a more accurate and powerful denoising performance. The combination of impulse detection and sparse coding-dictionary learning is proposed in [73] to restore the image and textures corrupted by mixed noise. The mixed noise is removed by adopting a variational encoding framework which is an integration of image sparsity prior and nonlocal self-similarity prior [74]. In [75], Weighted Joint Sparse Representation (WJSR) model is applied to mixed noise removal by jointly coding the grouped nonlocal similar image patches to obtain more robust denoising performance.

An effective mixed IN and AWGN removal method is proposed by Huang *et al.* [76] based on Laplacian Scale Mixture (LSM) modeling and nonlocal low-rank regularization. The impulse noise is characterized by LSM model where both the hidden scale parameters and the impulse noise are jointly estimated from the observed noisy image. Zhang *et al.* [77] proposed an algorithm to remove the mixture of Gaussian and impulse noise by exploiting the image local consistency and nonlocal consistency concurrently. Moreover, a Split-Bregman based iterative numerical algorithm is also developed to solve the optimization problem efficiently [77]. The integration of impulse noise detector with improved NL-means is presented in [78] to estimate the parameter for NL-means depending on the noise value.

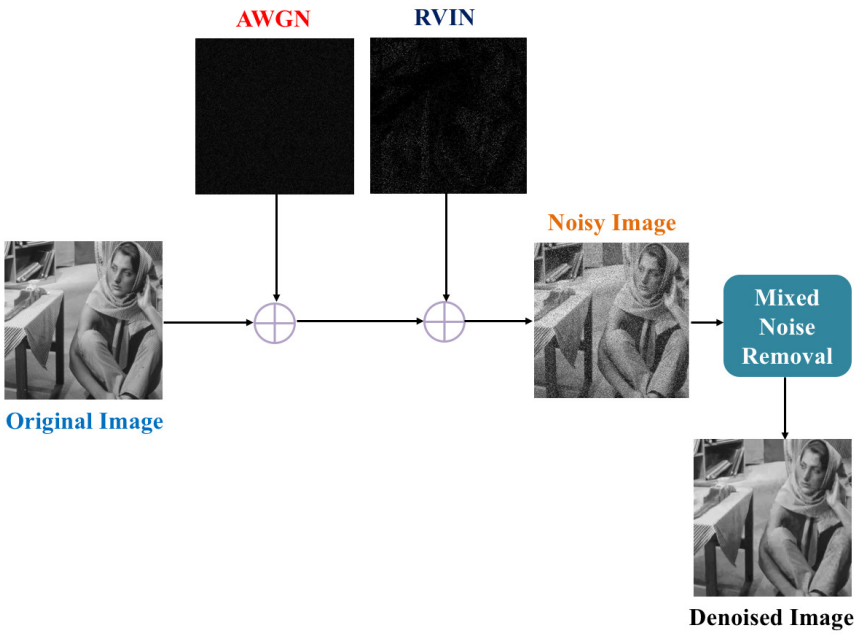


Figure 1.3: General block diagram of mixed noise removal

1.2 Types of Image Noise

Image noise is random variation of brightness or color information in the image captured by digital cameras. Noise generates undesirable effects such as artifacts, unrealistic edges, unseen lines, corners, blurred objects and degrades background scenes. There are various types of noise that are affected to the digital images in the real world. Some noise are multiplicative and some are additive. Among them, some well-known image noise mostly corrupted in the digital images are described in the following.

- Gaussian Noise
- Impulse Noise
- Shot Noise or Poisson Noise
- Speckle Noise

The characteristics of each noise will be briefly explained in the following subsection.

1.2.1 Gaussian Noise

Gaussian Noise is a statistical noise having a probability density function equal to normal distribution, also known as Gaussian distribution. It is generated by adding random Gaussian function with zero mean and standard deviation to image function. GN is also known as electronic noise because it arises in amplifiers or detectors. It is caused by thermal vibration of atoms and discrete nature of radiation of warm objects. The magnitude of Gaussian noise depends on the standard deviation (σ). When the value of standard deviation is higher, the noisy image may be stronger. The Gaussian noisy image of *Lena* is generated as shown in Fig. 1.4.

(a) Original (b) Noisy Image

Figure 1.4: Generation of Gaussian noise in *Lena* image with $\sigma = 30$

1.2.2 Impulse Noise

Digital images are frequently contaminated by impulse noise during the process of image acquisition or transmission through communication channels. Consequently, some pixel intensities are inevitably altered while others remain noise-free. The image model comprehending impulse noise with probability of occurrence p can be expressed as follows:

$$X_{ij} = \begin{cases} N_{ij} & \text{with probability } p; \\ S_{ij} & \text{with probability } 1-p \end{cases} \tag{1.1}$$

where S_{ij} denotes the noiseless image pixel and N_{ij} means the noise substituting for the original pixel. Two commonly encountered impulse noises in digital images are

- Salt-and-Pepper Noise (SPN) and
- Random-Valued Impulse Noise (RVIN).

The image corrupted by salt-and-pepper noise is produced by replacing the dark pixels (with 0 pixel value) in the bright regions and the bright pixels (with 255 pixel value) in the dark regions all over the image if the number of bits are 8 for transmission. This type of noise can be caused by analog-to-digital converter errors, bit errors in transmission and malfunctioning of cameras sensor cell.

RVIN is characterized by replacing a portion of image pixel with noise values normally distributed within the dynamic range $[0, 255]$ without altering the remaining pixel in the original image. To be clearly seen the generation of noisy images, Fig. 1.5 illustrate *Airplane* images degraded by salt-and-pepper noise $p = 5\%$ and random-valued impulse noise $p = 25\%$. In case of random-valued impulse noise, the detection of impulse noise is relatively more difficult in comparison with salt-and-pepper impulse noise because the intensity values of noisy pixels are randomly changed. Among the suppression of impulse noise, median filtering also known as nonlinear filtering is the most fundamental and effective one because of its effectiveness and computational efficiency. It can successfully optimize the tradeoff between noise reduction and detail preservation.

1.2.3 Shot Noise or Poisson Noise

Poisson noise is characteristic of many image acquisition modalities. It is signal dependent because the noise variance is equal to the expected value of the underlying reference signal. Therefore, it is more difficult to remove than Additive White Gaussian Noise. The manifestation of this noise is introduced due to the statistical nature of electromagnetic waves such as x-rays, visible lights and gamma rays. The x-ray and gamma ray sources emit number of photons per unit time. These rays are injected in patient's body from its source in medical x-rays and gamma rays imaging

(a) Original (b) Salt and pepper Noisy Image (c) RVIN Noisy Image

Figure 1.5: Generation of Salt-and-Pepper noise with $p = 5\%$ and RVIN with $p = 25\%$ in *Airplane* image

systems. The removal of poisson noise becomes a fundamental importance for many applications and particularly in astronomy and medical imaging.

1.2.4 Speckle Noise

Speckle noise is a form of multiplicative and locally correlated noise. It can be found in coherent imaging systems, such as laser, SAR (Synthetic Aperture Radar) and medical ultrasonic images. Speckle noise is generated in several ways in various type of images. In case of ultrasound images, speckle noise arises when a sound wave beat arbitrarily interferes with little particles or on a scale equivalent to sound wavelength. The presence of these noises significantly degrade the image quality and consequently the diagnostic decisions. In case of conventional radar images, it arises due to random variation in return signal. The speckle noise can be modeled as the following equation:

$$SN = I + n * I \tag{1.2}$$

where SN is the noisy image contaminated by speckle noise, I means the clean image and n is uniformly distributed random noise with mean and variance. The *Pepper* image disturbed by speckle noise with $\sigma = 15$ is produced as shown in Fig. 1.6. Many denoising methods such as spatial filtering approach, frequency domain approach and spatio-frequency approach have been widely used to suppress the speckle noise.

(a) Original (b) Noisy Image

Figure 1.6: Generation of Speckle noise in *Pepper* image with $\sigma = 15$

1.3 Causes of Noise in Digital Images

Digital images in the real applications are often corrupted by one or more than one type of noise due to many kinds of reasons as expressed in the following:

- Environmental conditions
- Temperature of the sensor
- Confliction in the transmission channel
- Insufficient light
- Dust on the screen during scanning the image

The presence of noise in the digital images leads to generation of poor quality image and distortion of important image features. Thus, image denoising plays a critical role in the subsequent image processing applications such as object recognition, surveillance, remote sensing and image compression etc, as a preprocessing step. Figure. 1.7 depicts the pictorial representations of the corruption of noise in digital images by different causes at different conditions.

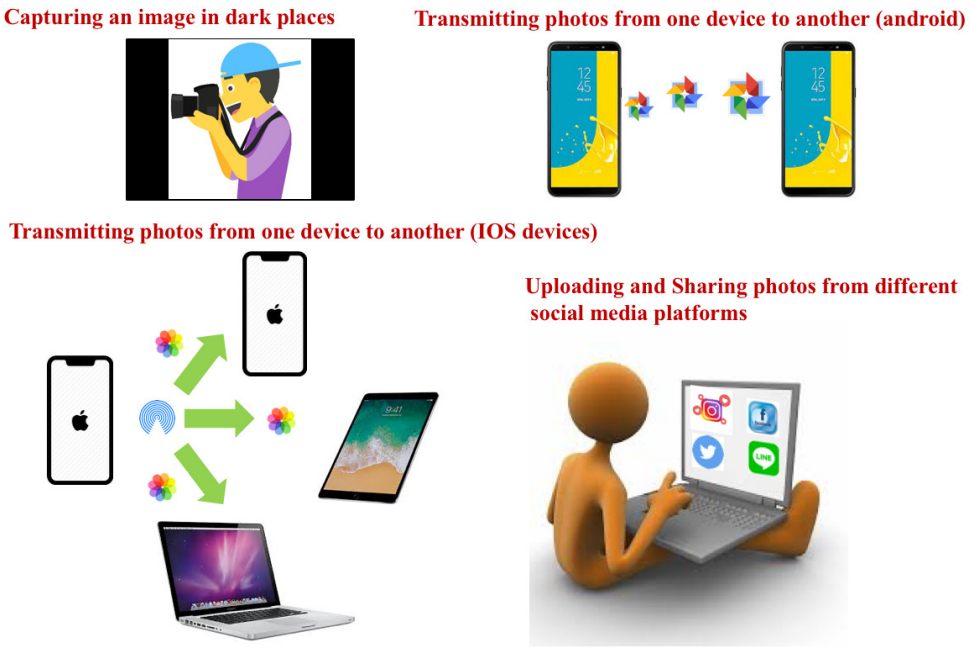


Figure 1.7: Causes of noise in digital images at different conditions

1.4 Problem Statement

Natural images in the real world can be degraded by more than one type of noise during the process of image acquisition and transmission. Many researchers have been trying to remove the mixed noise for a few decades because it is a fundamental and essential step to be considered in sequential image processing applications. Therefore, they utilized many denoising methods for removal of mixed noise because the characteristics of noise are different from each other. In this thesis, how to remove mixed noise composed of Additive White Gaussian Noise (AWGN) and Random-Valued Impulse Noise (RVIN) , and Gaussian noise only are mainly considered. Mixed noise image can be obtained by adding two commonly encountered noise such as AWGN and RVIN to clean image. Gaussian noisy image is produced by concatenating the Gaussian noise with zero mean and standard deviation to a noise-free image.

The removal of mixed-noise is more difficult than the removal of Gaussian noise only because different noises have different characteristics. Most mixed-noise removal methods are basically detection based methods. The presence of impulse noise in the

noisy image is initially detected and then filtered in the detection-based noise removal methods. Although most mixed-noise removal methods can successfully suppress the noise, some image details are lost in the edge and texture regions because of the miss-detection of the image details as the impulse noise. The cropped region of denoised *Barbara* image degraded by the combination of Gaussian noise $\sigma = 25$ and RVIN $p = 25\%$ is illustrated in Fig. 1.8. The image details in the image are failed to preserve according to the simulation result. Hence, the detection of impulse noise becomes a challenging problem to remove the mixed noise and restore some important features of the image.

(a) Original (b) Denoised Image

Figure 1.8: Denoised result of *Barbara* corrupted by mixed Gaussian and random-valued impulse noise ($\sigma = 25$ and $p = 25\%$)

Moreover, an accurate Gaussian noise removal approach has been recently attracting considerable attentions to apply for mobile processors like smart phone. There are accurate conventional denoising methods that have the potential ability for improving the denoising performance with no more time. Local-based denoising methods as well as nonlocal-based denoising methods have been extensively used to remove the Gaussian noise since a few years. Local-based denoising methods can effectively remove the noise. However, they tend to blur the edges and over-smooth the flat regions of the image when the noise level is strong. Similarly, even though some state-of-the-art nonlocal-based denoising methods such as Block Matching and 3D filtering (BM3D) , and Weighted Nuclear Norm Minimization (WNNM) produce high quantitative per-

formance, some fine details of the image are less visualized due to the loss of high frequency information. The visual quality of the *Butterfly* image corrupted by Gaussian noise $\sigma = 60$ by using BM3D filtering is shown in Fig. 1.9. Not only the edge regions of the image are poor but also the flat regions lead to a little blur in the situation of high noise level.

(a) Original (b) Denoised Image

Figure 1.9: Denoised result of *Butterfly* corrupted by AWGN ($\sigma = 60$)

1.5 Research Objectives

As explained in Section. 1.4, it is difficult to successfully remove the mixed noise composed of AWGN and RVIN without sacrificing the image features because of the detection problem of the image details in the image. Moreover, some image structures in the nonlocal-based image denoising techniques are failed to efficiently restore due to the disappearance of high frequency components. Therefore, this dissertation particularly aims at the following factors.

- To preserve the image details lost in the mixed noise removal method by using Interpolation approach based on multi-surface fitting for single image
- To compensate the high frequency information in the nonlocal-based Gaussian noise removal methods with the help of Improved RAISR technique

1.6 Research Methodology

In this dissertation, a mixed-noise removal method is proposed to preserve the image details with the aid of image interpolation technique that will be mentioned in Chapter.3. There are typically two steps in the proposed method. The first step is to estimate the denoised image by integrating interpolation, DWM filter which is used to eliminate the impulse noise, down-sampling and BM3D which is utilized for Gaussian noise reduction. The second step is to preserve the image details lost in the first step by calculating the absolute difference between the input noisy image and the pre-estimated image obtained from the first step. In addition, BM3D is also used to suppress the remaining Gaussian noise after removing the accurately detected impulse noise. The main contribution of this research is that the input noisy image is initially interpolated by multi-surface fitting for single frame before impulse noise detection of DWM filter in the first step to keep the image details in the edge and texture regions.

Furthermore, an improvement of Gaussian noise removal method using learning-based super-resolution (SR) approach as a rapid post-processing is also proposed in this dissertation. Block Matching and 3D filtering (BM3D) , and Weighted Nuclear Norm Minimization (WNNM) are utilized to suppress the noise. Although these nonlocal-based image denoising methods can efficiently remove GN, some important details of the image are deteriorated due to the loss of high frequency information. In order to tackle this problem, an Improvement of Rapid and Accurate Image Super-Resolution, so-called IRAISR is applied as a rapid post-processing to the denoised image because it leads to high performance comparable to some state-of-the-art super-resolution (SR) techniques with low cost computational complexity as well as preserves the important image structures. The modification of this approach is that the hash classes for the patches extracted from the denoised image and the pixels from Ground truth are reduced to 18 filters by two improvements including the minimization of the classes for the angle by geometric conversion and reduction of the classes for the strength. The filters are learned from the image pairs between

the patches extracted from the denoised image and the pixels from the clean image by using eigen-analysis. Similarly to the pioneer RAISR, the Census Transform (CT) is exploited by blending the image processed by noise suppression methods with the filtered output to achieve the artifact-free results.

1.7 Organization of the Dissertation

Chapter 1 provides a brief introduction to background theory of image denoising, a variety of noise commonly degraded in the digital images in the real-world, the causes of noise in the digital images, the main issues in removing both mixed noise and Gaussian noise, the principal objectives to tackle this problem, needed methods in this study and organization of the dissertation.

Chapter 2 presents the characteristics and methodology of some state-of-the-art nonlocal-based Gaussian noise removal methods and convolutional neural network (CNN)-based denoising methods.

Chapter 3 expresses the removal of mixed-noise comprised of Additive White Gaussian Noise (AWGN) and Random-Valued Impulse Noise (RVIN) by using interpolation technique based on multi-surface fitting for single frame.

Chapter 4 describes the improvement of denoising performance in an accurate Gaussian noise removal method by employing Improved Rapid and Accurate Image Super-Resolution (IRAISR) technique with less number of filters.

Chapter 5 explains an overall conclusion of this thesis, some limitations and further extensions of image denoising methods proposed in this thesis.

Chapter 2

Related Works

2.1 Nonlocal-based Image Denoising Methods

Image denoising is a fundamental problem in image processing to be considered for many researchers in order to estimate the original image from noise-contaminated observation while preserving the image features as much as possible. Traditionally, the classical linear filtering known as Gaussian filtering is utilized to remove Additive White Gaussian Noise (AWGN). Even though the flat regions of the image are optimally restored in this filtering, it tends to blurry effect in the edge and texture regions because the intensity values are drastically changed in that regions. To overcome this problem, the edge preserving filter so-called bilateral filter [3] is widely used. The bilateral filter is the simplest and most intuitive filter based on the weighted-averaging. The filtered output at each pixel is computed as the average of its neighboring pixels and the weights are evaluated between the spatial and intensity similarity. The best PSNR performance can be achieved depending upon the selection of spatial and range kernels. The image can be smoothed and the edges are well restored in bilateral filter.

As these above image denoising methods are locally based, the denoising performance is significantly decreased when the noise level is gradually increased. Therefore, nonlocal-based image denoising methods have been recently popular for the removal of AWGN. Among them, many denoising methods based on the nonlocal self-similarity (NSS) models such as nonlocal means (NLM) filter [11, 12], Block Matching and 3D filtering (BM3D) [24], Learned Simultaneous Sparse Coding (LSSC) [34], Nonlocally Centralized Sparse Representation (NCSR) [39] and Weighted Nuclear Norm Minimization (WNNM) [30] have been extensively addressed for the removal of Gaussian noise to obtain an excellent performance while preserving the image details.

2.1.1 Nonlocal Means

Nonlocal means (NLM) is actually an extension of bilateral filtering (BF) based on the fact that similar pixels in an image can be spatially far from each other. It is also a pioneer of denoising method based on nonlocal self-similarity (NSS) models. In NLM, the estimated value at each pixel is computed as the weighted average of all its similar pixels in the image, and the weights are determined by the similarity between them. The main idea of NLM is the calculation of similarity between a target pixel processed and its neighborhoods across the image by using Euclidean distance as a similarity metric.

The denoised result of the image by using NLM is computed by the following:

$$\hat{y}(i) = \sum_{j \in I} \omega(i,j) y(j) \tag{2.1}$$

where $\boldsymbol{y}$ is a noisy image which is an addition of noise-free image $\boldsymbol{x}$ and Gaussian noise n with zero mean and variance σ^2. The weights dependent on the similarity between the pixels i and j can be calculated by

$$\omega(i,j) = \frac{1}{Z(i)} e^{-\frac{\left\| y(N_i) - y(N_j) \right\|_{2,\sigma}^2}{h^2}} \tag{2.2}$$

where N_i and N_j denote square neighborhood of fixed size and centered at pixels i and j, respectively. $\sigma > 0$ is the standard deviation of Gaussian kernel. $Z(i)$ is the normalizing constant and can be evaluated by

$$Z(i) = \sum_{j} e^{-\frac{\left\| y(N_i) - y(N_j) \right\|_{2,\sigma}^2}{h^2}} \tag{2.3}$$

where the parameter h is a degree of filtering.

The decay of the weights can be controlled by this parameter as a function of the Euclidean distances. The NL-means allow a more robust to high noise comparison with the other locally-based denoising methods because not only the intensity level at

a single point but also the geometrical configuration in a whole neighborhood can be considered in NLM. To the best of our knowledge, similar pixel neighborhoods give a large weight while the small weights are obtained in much different pixel neighborhoods. Hence, the similarity of the target pixel with its nearest neighborhoods plays an essential role in NLM to achieve higher performance over local-based Gaussian noise removal methods.

2.1.2 BM3D Filtering

Block Matching and 3D filtering (BM3D) [24] has become a state-of-the-art Gaussian noise removal method since a few years due to its high quantitative performance and subjective visual evaluation. It is also based on nonlocal self-similarity (NSS) models and uses transform-based shrinkage in the frequency domain. There are two steps in BM3D as illustrated in Fig. 2.1. The first step is to obtain the pre-estimated image by using block matching which is a grouping of nonlocal similar patches into a 3-D cube and applying hard-thresholding operation. The second step is to achieve the restored image by utilizing block matching and employing wiener filtering.

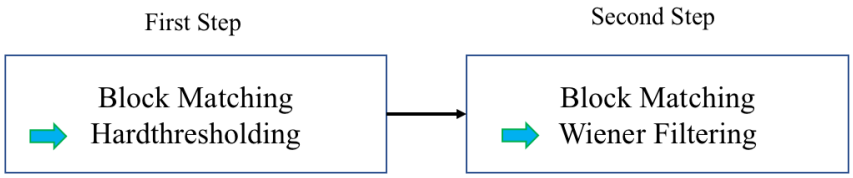


Figure 2.1: Block diagram of BM3D

In hard thresholding operation, BM3D defines the difference d to measure the similarity between the target patch and the candidate patches at different locations of the image as

$$d(\boldsymbol{p_i}, \boldsymbol{p_j}) = \frac{1}{N_1} \| \Upsilon \left(F_{2D}(\boldsymbol{p_i}), \lambda_{2D}\sigma\sqrt{2\log(N_1^2)} \right) - \Upsilon \left(F_{2D}(\boldsymbol{p_j}), \lambda_{2D}\sigma\sqrt{2\log(N_1^2)} \right) \|_2 \tag{2.4}$$

where $\boldsymbol{p_i}$ and $\boldsymbol{p_j}$ are a target patch processed and a neighbor patch sized $N_1 \times N_1$ in the input noisy image $\boldsymbol{y}$ corrupted by AWGN, F_{2D} is a 2D linear unitary transform operator, λ_{2D} is a fixed threshold parameter, σ is the variance of AWGN, and $\|.\|_2$ denotes the L_2-norm. Υ is the hard-threshold operator and it is defined as

$$\Upsilon(\lambda, \lambda_{thr}) = \begin{cases} \lambda, & |\lambda| > \lambda_{thr} \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

where λ_{thr} is the threshold parameter. A set $\boldsymbol{S}_i$ of the coordinates of the blocks similar to $\boldsymbol{p_i}$ can be obtained as follows.

$$\boldsymbol{S}_i = \{(x_j, y_j) \quad | \quad d(\boldsymbol{p_i}, \boldsymbol{p_j}) < \tau\} \tag{2.6}$$

where (x_j, y_j) is the coordinates of the nonlocal similar patch $\boldsymbol{p_j}$ and τ is the maximum parameter of d. Then, the similar patches extracted from the noisy image are stacked together to form a 3-D array sized $N_1 \times N_1 \times |\boldsymbol{S}_i|$, where $\boldsymbol{S}_i$ describes the cardinality of $\boldsymbol{S}_i$ and hard-thresholded to attenuate the noise.

The estimates of all grouped blocks can be produced by inverting the 3-D transform and returned back to their original positions. The denoised array can be depicted as follows.

$$\tilde{\boldsymbol{Y}}_{\boldsymbol{i}} = F_{3D}^{-1}\left(\Upsilon\left(F_{3D}(\boldsymbol{P_i}), \lambda_{3D}\sigma\sqrt{2\log(N_1^2)}\right)\right) \tag{2.7}$$

where $\boldsymbol{P_i}$ is the 3-D array for target patch $\boldsymbol{p_i}$, F_{3D} and F_{3D}^{-1} are 3-D linear unitary transform operator and the inverse operator, and λ_{3D} is a fixed threshold parameter. Moreover, the weight ω_i for each array $\tilde{\boldsymbol{Y}}_{\boldsymbol{i}}$ can be calculated as

$$\omega_i = \begin{cases} \frac{1}{N_{iHT}} & N_{iHT} \geq 1 \\ 1 & \text{otherwise} \end{cases} \tag{2.8}$$

where N_{iHT} is the number of non-zero coefficients of $\boldsymbol{P}_i$.

In wiener filtering, two groups which are one from the input noisy image and the another from the estimates evaluated by using hard-thresholding are formed by block

matching. For the block matching, the difference $d^{'}$ is computed as follows.

$$d^{'}(\boldsymbol{q_i}, \boldsymbol{q_j}) = \frac{1}{N_1} \left\| (\boldsymbol{q_i} - \bar{q}_i) - (\boldsymbol{q_j} - \bar{q}_j) \right\|_2 \tag{2.9}$$

where $\boldsymbol{q_i}$ and $\boldsymbol{q_j}$ are a target patch and its neighbor patch in $\tilde{\boldsymbol{y}}$, and $\bar{q}_i$ and $\bar{q}_j$ are the mean value of $\boldsymbol{q_i}$ and $\boldsymbol{q_j}$, respectively. Analogously to the first step, a set $\acute{\boldsymbol{S}}_i$ of the coordinates of the blocks similar to $\boldsymbol{q_i}$ can be obtained as follows.

$$\acute{\boldsymbol{S}}_i = \{(x_j, y_j) \quad | \quad \acute{d}(\boldsymbol{q_i}, \boldsymbol{q_j}) < \tau\} \tag{2.10}$$

where (x_j, y_j) is the coordinates of the patch $\boldsymbol{q_j}$. In the second step, BM3D processes the linear wiener filter instead of hard-thresholding to improve the grouping of blocks because hard-threshold removes not only AWGN but also the important image details. The estimates of all grouped blocks are analogously produced by applying the inverse 3-D transform on the wiener-filtered coefficients and put them back to their original positions. BM3D produces the filtered array $\hat{\boldsymbol{Y}}_{\boldsymbol{i}}$ as follows.

$$\hat{\boldsymbol{Y}}_{\boldsymbol{i}} = F_{3D}^{-1} \left(\boldsymbol{W}_{\boldsymbol{Q_i}} F_{3D}(\boldsymbol{P_i}) \right) \tag{2.11}$$

where $\boldsymbol{P_i}$ is the noisy observation of 3-D array $\boldsymbol{Q_i}$ in $\boldsymbol{y}$. The Wiener filter coefficients in a 3-D transform can be calculated from the estimates of the first step

$$\boldsymbol{W}_{\boldsymbol{Q_i}}(i,j,t) = \frac{|F_{3D}(\boldsymbol{Q})(i,j,t)|^2}{|F_{3D}(\boldsymbol{Q})(i,j,t)|^2 + \sigma^2}. \tag{2.12}$$

After processing these two operations for all target patches, the restored image is finally estimated by aggregating all denoised arrays of each patch with their relevant weights in both processes.

2.1.3 Learned Simultaneous Sparse Coding

Mairal *et al.* [34,35] proposed a Learned Simultaneous Sparse Coding (LSSC) which is a nonlocal-based framework by combining nonlocal means (NLM) and sparse coding to restore the image. In LSSC, the dictionaries D are learned on the set of overlapping

noisy patches. Unlike NLM, LSSC can solve the denoising process in the patches that are not similar to the other patches in the image by forcing the similar patches to similar sparse decompositions. The convex $l_{1,2}$ norm is fundamentally used for learning the dictionary and the $l_{0,\infty}$ pseudo-norm is utilized for the final reconstruction in this approach. The set S_i of similar patches for each patch $\boldsymbol{y}_i$ from the noisy image $\boldsymbol{y}$ is defined as

$$S_i \triangleq \{j = 1, ..., n \quad \text{s.t} \quad \|y_i - y_j\|_2^2 \leq \xi\} \tag{2.13}$$

where y_i is the target patch processed and y_j means the similar patches related to the target patch across the image. ξ is the threshold value.

The sparse optimization problem can be solved by decomposing the patch y_i with a grouped-sparsity regularizer on the set S_i, expressed as

$$\min_{A_i} \|A_i\|_{p,q} \quad \text{s.t} \quad \sum_{j \in S_i} \|y_j - D\alpha_{ij}\|_2^2 \leq \epsilon_i \tag{2.14}$$

where $A_i = [\alpha_{ij}]_{j \in S_i} \in R^{k \times |S_i|}$. $D\alpha_{ij}$ is the estimate of denoised image and α_{ij} is the corresponding code for the learned dictionary. k is the number of elements.

The maximum window size ω for semi-local grouping is selected to 64 in LSSC. The key contribution is the computation of only n vector α_{ij} because each pixel in the image belongs to exactly one cluster for grouping. In contrast to BM3D, the patches are initially denoised from the noisy image by sparse coding approach before extracting the patches from the noisy image to improve matching. In addition, the mean intensity (or RGB color) value of a patch is often subtracted from all its pixel values in order to enhance the numerical stability of sparse coding. Hence, some image restoration tasks such as image denoising and color image demosaicking can be addressed successfully by applying the integration of nonlocal means and sparse coding approach.

2.1.4 Nonlocally Centralized Sparse Representation

Nonlocally Centralized Sparse Representation (NCSR) [38, 39] is one of the most popular nonlocal-based denoising methods for image restoration. The main key of this method is to recover the original image from the noisy measurements by reducing the sparse coding noise (SCN) which is the difference between the sparse code $\boldsymbol{\alpha}_y$ of the observed image and the sparse code $\boldsymbol{\alpha}_x$ of the reference image. The minimization problem can be solved by the following equation to reconstruct $\boldsymbol{x}$ from the observed image degradation model $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + v$:

$$\boldsymbol{\alpha}_y = \arg\min_{\alpha}\{\|\boldsymbol{y} - \boldsymbol{H}\Phi \circ \boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1\} \tag{2.15}$$

where $\boldsymbol{H}$ is an identity matrix for image denoising, λ means the regularization parameter to adjust the sparse approximation error $\boldsymbol{y}$ and the sparsity of $\boldsymbol{\alpha}$, and $\Phi \in R^{n\times M}(n < M)$ is an over-complete dictionary. The reconstructed image $\hat{\boldsymbol{x}}$ can be calculated by the following equation

$$\hat{\boldsymbol{x}} = \Phi \circ \boldsymbol{\alpha}_{\boldsymbol{y}}. \tag{2.16}$$

One major challenging problem about sparsity-based image restoration is the selection of dictionaries Φ. In NCSR, the adaptive sparse domain selection strategy is adopted and the sub-dictionaries are learned from the given image instead of the example images, leading to a more stable and sparser representation. The extracted image patches from the image $\boldsymbol{x}$ are clusterd into K clusters by using the K-means clustering method. The over-complete dictionary for each cluster is not needed to learn because the patches in a cluster are similar to each other. Therefore, the learned dictionary-based PCA is employed to code the patches in this cluster. These K PCA sub-dictionaries construct a large over-complete dictionary to characterize all the possible local structures of natural images.

The nonlocally centralized sparse representation (NCSR) model can be expressed

as follows.

$$\boldsymbol{\alpha}_y = \arg\min_{\alpha}\{\|\boldsymbol{y} - \boldsymbol{H}\Phi \circ \boldsymbol{\alpha}\|_2^2 + \lambda \sum_i \|\boldsymbol{\alpha}_i - \boldsymbol{\beta}_i\|_p\} \tag{2.17}$$

where $\|\boldsymbol{\alpha}_i - \boldsymbol{\beta}_i\|$ is a nonlocally centralized sparsity term and p is assumed to be 1. The estimate $\boldsymbol{\beta}_i$ is obtained by using the nonlocal redundancy of natural images. Then, a good estimation of $\boldsymbol{\alpha}_i$, i.e., $\boldsymbol{\beta}_i$ can be calculated as the weighted average of those sparse codes associated with the nonlocal similar patches to the target patch i. The sparse codes of patch $\boldsymbol{x}_{i,q}$ within a set of similar patches $\boldsymbol{\Omega}_i$ are denoted by $\boldsymbol{\alpha}_{i,q}$. Then, $\boldsymbol{\beta}_i$ can be computed as the weighted average of $\boldsymbol{\alpha}_{i,q}$

$$\boldsymbol{\beta}_i = \sum_{q \in \boldsymbol{\Omega}_i} \omega_{i,q} \boldsymbol{\alpha}_{i,q} \tag{2.18}$$

where $\omega_{i,q}$ is the weight. The weights should be inversely proportional to the distance between the target patch processed $\boldsymbol{x}_i$ and its similar patches $\boldsymbol{x}_{i,q}$

$$\omega_{i,q} = \frac{1}{W} \exp(-\|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_{i,q}\|_2^2 / h) \tag{2.19}$$

where $\hat{\boldsymbol{x}}_i = \Phi\hat{\boldsymbol{\alpha}}_i$ and $\hat{\boldsymbol{x}}_{i,q} = \Phi\hat{\boldsymbol{\alpha}}_{i,q}$ are the estimates of the patches $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i,q}$, h is a pre-determined scalar and W is the normalization factor.

In order to improve the accuracy of the sparse codes and the image restoration quality, the estimation of sparse vector can be iteratively solved by the following minimization problem

$$\boldsymbol{\alpha}_y^{(l)} = \arg\min_{\alpha}\{\|\boldsymbol{y} - \boldsymbol{H}\Phi \circ \boldsymbol{\alpha}\|_2^2 + \lambda \sum_i \|\boldsymbol{\alpha}_i - \boldsymbol{\beta}_i^{(l)}\|_p\} \tag{2.20}$$

The restored image is then updated as $\hat{\boldsymbol{x}}^{(l)} = \Phi \circ \boldsymbol{\alpha}_y^{(l)}$. When the accuracy of sparse coding coefficient $\boldsymbol{\alpha}_y^{(l)}$ in the iteration process is gradually improved, the accuracy of $\boldsymbol{\beta}_i$ will be significantly increased. The improved $\boldsymbol{\beta}_i$ is used to improve the accuracy of $\boldsymbol{\alpha}_y$.

In addition to reduce the SCN, the tremendous amount of nonlocal redundancies are exploited to the sparse coding coefficients of $\boldsymbol{\alpha}_x$ and then the sparse codes $\boldsymbol{\alpha}_y$

of the observed image are centralized to these estimations of the original image. Accordingly, the denoising performance of NCSR is improved by using this approach and competitive with benchmark BM3D for Gaussain noise removal.

2.1.5 WNNM Filtering

Low rank matrix approximation is an important approach to compensate the underlying low rank matrix from its degraded observation. There are two categories in low rank matrix approximation methods namely: the low rank matrix factorization (LRMF) methods and the nuclear norm minimization (NNM) methods. LRMF is able to be factorized into the product of two low rank matrices to approximate a given data matrix.

The LRMF problem is fundamentally nonconvex optimization problem. Different from LRMF, NNM is a convex optimization problem with certain data fidelity term. The lowest rank approximation $\mathbf{X}$ from the observed matrix $\mathbf{Y}$ can be estimated based on NNM problem:

$$\hat{\boldsymbol{X}} = \arg\min_{X} \|\boldsymbol{Y} - \boldsymbol{X}\|_F^2 + \lambda \|\boldsymbol{X}\|_* \tag{2.21}$$

where $\|.\|_F^2$ denotes the Frobenius norm to measure the difference between the observed data matrix $\boldsymbol{Y}$ and the latent data matrix $\boldsymbol{X}$, $\|\boldsymbol{X}\|_*$ assigns the nuclear norm of a matrix $\boldsymbol{X}$, is defined as the sum of its singular values, i.e., $\|\boldsymbol{X}\|_* = \sum_i \|\sigma_i(\boldsymbol{X})\|_1$, where $\sigma_i(\boldsymbol{X})$ is the i-th singular value of $\mathbf{X}$, and λ is a positive constant. The solution to this problem can be obtained by

$$\hat{\boldsymbol{X}} = \boldsymbol{U} S_\lambda(\boldsymbol{\Sigma}) \boldsymbol{V}^T, \tag{2.22}$$

where $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the SVD of $\boldsymbol{Y}$ and $S_\lambda(\boldsymbol{\Sigma}) = \max(\boldsymbol{\Sigma} - \lambda \boldsymbol{I}, 0)$ is the singular value soft-thresholding operator. In NNM, the same soft-threshold will be applied to all the singular values because the weights of each singular value are equal. Therefore, it is not reasonable and flexible to estimate the low-rank matrix approximation.

In order to improve the flexibility of NNM, Gu *et al.* proposed Weighted Nuclear Norm Minimization (WNNM) [30] which assigns different weights to the singular values in the restoration of the original image from its degradation model. From the noisy image $\boldsymbol{y}$, the non-local similar patches relevant to the target patch across the image are searched for a local patch y_j by block matching as mentioned in [24]. The patch matrix of noisy image, defined by $\boldsymbol{Y}_j$ is obtained by stacking the non-local similar patches from the noisy image. Then, the low rank matrix approximation method based on WNNM can be used to estimate $\boldsymbol{X}_j$ from $\boldsymbol{Y}_j$. The denoising structure of WNNM is illustrated in Fig. 2.2 to remove the Gaussian noise. The WNNM problem is proposed as described in

$$\hat{\boldsymbol{X}}_j = \arg\min_{X_j} \frac{1}{\sigma_n^2} \|\boldsymbol{Y}_j - \boldsymbol{X}_j\|_F^2 + \|\boldsymbol{X}_j\|_{\boldsymbol{w},*} \tag{2.23}$$

where σ_n^2 is the noise variance to normalize the F-norm data fidelity term $\|\boldsymbol{Y}_j - \boldsymbol{X}_j\|_F^2$, $\|\boldsymbol{X}_j\|_{\boldsymbol{w},*} = \sum_i \|\omega_i \sigma_i(\boldsymbol{X}_j)\|_1$ is the weighted nuclear norm of $\boldsymbol{X}$,$\boldsymbol{w}$ is the weight vector and ω_i is a non-negative weight assigned to $\sigma_i(\boldsymbol{X}_j)$.

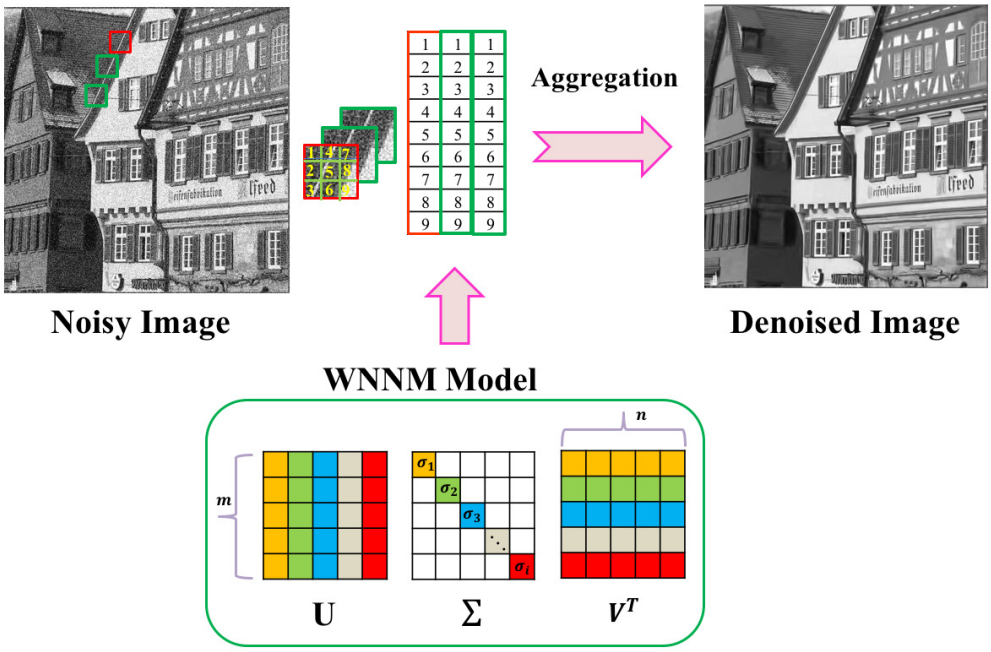


Figure 2.2: Denoising model of WNNM

In the application of denoising, the singular values should be large because they

represent the energy of the major components of the data matrix $\boldsymbol{X}_j$ to be preserved well. Hence, the weight assigned to singular values of $\boldsymbol{X}_j$ might be inversely proportional to $\sigma_i(\boldsymbol{X}_j)$ as described in the following:

$$\omega_i = c\sqrt{n}/\left(\sigma_i(\boldsymbol{X}_j) + \varepsilon\right) \tag{2.24}$$

where $c > 0$ is a constant, n is the number of similar patches in $\boldsymbol{Y}_j$, $\sigma_i(\boldsymbol{X}_j)$ is the i-th singular value of $\boldsymbol{X}_j$,and $\varepsilon = 10^{-16}$ is a constant parameter to avoid division by zero. The unknown singular values $\sigma_i(\boldsymbol{X}_j)$ can be estimated as

$$\hat{\sigma}_i(\boldsymbol{X}_j) = \sqrt{\max(\sigma_i^2(\boldsymbol{Y}_j) - n\sigma_n^2, 0)}, \tag{2.25}$$

where $\sigma_i(\boldsymbol{Y}_j)$ is the i-th singular values of $\boldsymbol{Y}_j$. The number of iterations and the patch size are selected depending on the noise levels in the application of image denoising by WNNM. Note that the weights are sorted in a non-descending order since the singular values have been arranged in non-ascending order in this consideration. Finally, the reconstructed image $\boldsymbol{x}$ can be achieved by applying the above procedures to each patch in the noisy image $\boldsymbol{y}$ and aggregating all of the denoised patches together to return to their original positions.

2.2 CNN-based Image Denoising Methods

Image denoising methods are generally divided into two categories: image priors model based methods and discriminative learning based methods. Although most of the image priors model based methods produce high quantitative measurements and superior visual quality, some limitations are found in these methods. The two major drawbacks are time consuming due to the consideration of optimization problems in the testing stage and the lack of restoration of image structures because of the selection of hand-crafted parameters to boost the denoising performance.

2.2.1 Residual Learning of Deep CNN for Image Denoising

In contrast to the image priors model based methods, the main idea behind the image denoising methods based on discriminative learning is to learn the underlying image priors and fast inference from a training set of degraded and ground-truth image priors. A cascade of shrinkage fields (CSF) method that combines the random field-based model and the unrolled half-quadratic optimization algorithm to form a single learning frame-work is proposed in [79]. The aim of trainable nonlinear reaction diffusion (TNRD) model [80, 81] is to learn stage-wise image priors in the context of truncated inference procedure. CSF and TNRD give high potential to image denoising while bridging a gap between computational efficiency and denoising performance. However, they lead to some drawbacks such as limitation in capturing the characteristics of image structures, robustness of denoising performance due to the manually selective parameters and consideration of training a specific model at certain noise level.

In order to solve this problem, the suppression of noise from the noisy image by feed-forward convolutional neural networks (CNN), namely DnCNN is proposed in [48, 49] instead of learning a discriminative model with an explicit image prior. The DnCNN model is to learn a mapping function $\hat{\boldsymbol{x}} = F(\boldsymbol{y}; \Theta_\sigma)$ between the input noisy version $\boldsymbol{y}$ and the desired clean image $\hat{\boldsymbol{x}}$. The parameters Θ_σ are trained for noisy images contaminated by AWGN at fixed noise variances. An important issue in network architecture design of DnCNN is the selection of proper network depth. When the noise level is high, the effective patch size should be large. The receptive field size of DnCNN is set to 35×35 with the corresponding depth of 17 for Gaussian denoising with a certain noise level.

There are typically three types of layers in DnCNN. In the first layer, 64 features maps are generated by using 64 filters of size $3 \times 3 \times c$ for convolution and rectified linear units (ReLU, max(0,.)) [82] are then utilized for nonlinearity. c represents the number of image channels, i.e., $c = 1$ for gray scale image and $c = 3$ for color image. In the second layer, 64 filters of size $3 \times 3 \times c$ are used and batch normalization [83] is added

between convolution and ReLU. In the last layer, c filters of size $3 \times 3 \times 64$ are used to reconstruct the output. In addition, simple zero padding strategy is utilized before convolution to reduce the boundary artifacts in DnCNN. In this method, not only the training process is accelerated but also the denoising performance is enhanced by integrating the residual learning [84] and batch normalization [83]. Residual learning tends to solve the performance degradation problem and the use of BN is to alleviate the internal covariate shift while producing the excellent denoised results. Therefore, Gaussian denoising method using feed-forward convolution neural network (DnCNN) is competitive with some state-of-the-art nonlocal based denoising methods in both gray scale and color images.

2.2.2 Fast and Flexible Denoising Convolutional Neural Network: FFDNet

The above mentioned DnCNN [48, 49] model is not flexibility to deal with spatially variant noises and has some limited applications in practical denoising because it requires multiple models for denoising images with different noise levels. To circumvent this issue, the fast and flexible denoising convolutional neural network (FFDNet) [50] is appeared to adaptively perform a trade-off between noise elimination and detail restoration. The three main objectives of FFDNet are to be highly efficient without disturbing denoising performance, to handle images with different noise levels and even spatially variant noise , and to achieve artifacts-free images in controlling the trade-off between noise reduction and detail preservation.

Unlike the DnCNN model, the FFDNet model can be formulated as $\hat{\boldsymbol{x}} = F(\boldsymbol{y}, \boldsymbol{M}; \Theta)$ by addressing the tunable noise level map $\boldsymbol{M}$ as a main contribution. The architecture of FFDNet is mainly classified into three layers. In the first layer, the noisy image $\boldsymbol{y}$ is reshaped into four downsampled sub-images by using down-scaling factor 2. Then, a tunable noise level map $\boldsymbol{M}$ which is a stretching of noise level σ is concatenated with the down-sampled images to form a tensor $\tilde{\boldsymbol{y}}$ of size $\frac{W}{2} \times \frac{H}{2} \times (4C+1)$ as the inputs to CNN where W means the width of the image, H is the height of

the image and C defines the image channel i.e., $C = 1$ for gray scale and $C = 3$ for color image. With the tensor $\tilde{\boldsymbol{y}}$ as input, the CNN is composed of a series of 3×3 convolution layers. The combination of convolution (Conv) and Rectified Linear Unit (ReLU) [82] for the first layer, the integration of Conv, batch normalization (BN) [83] and ReLU for the middle layer and Conv for the last layer are adopted respectively. Similarly to DnCNN, zero-padding is utilized to maintain the size of feature maps after convolution. After that, the estimated clean image $\hat{\boldsymbol{x}}$ of size $W \times H \times C$ is produced by applying upscaling operation as the reverse operator of the downsampling operator applied in the input stage.

The efficiency of image denoising for practical CNN-based denoising can be improved by applying the downsampled images as input. The role of noise level map $\boldsymbol{M}$ becomes important to control the trade-off between noise reduction and detail preservation and to avoid the possible visual artifacts caused by noise level mismatch. Moreover, residual learning is not necessary for network design in FFDNet because the final performance after fine-tuning are almost exactly the same. During training and testing of FFDNet-Clip, the noisy images are quantized into 8-bit format. Thus, it is more flexible than the other discriminative learning based methods at different noise levels with a single network while accelerating both the training and testing speed.

Chapter 3

Image Denoising Using Interpolation

3.1 Introduction

Image denoising for mixed-noise is an essential issue to be taken into account for many researchers in the image processing applications because most of the digital images in the real world may be corrupted by more than one type of noise due to different conditions. Therefore, how to remove the mixed-noise and which technique will be used to enhance the image quality without losing the image details such as edges and textures are addressed in this chapter. Most mixed-noise removal methods are fundamentally based on the detection because of the presence of impulse noise in the mixed-noise. Many denoising methods for mixed-noise have effectively suppressed the noise for a few years as mentioned in Chapter 1.

Moreover, a two-step mixed noise removal method is reported in [85] to eliminate RVIN and AWGN mixed noise. In the first step, DWM filter [64] is utilized to remove RVIN and BM3D [24] is applied for removing AWGN. In the second step, a straightforward and effective noise detection method which is thresholding on the absolute difference between the input noisy image and the pre-estimated image obtained from the first step is employed to accurately detect the impulse noise. Besides, BM3D is again utilized to remove the remaining noise. Although the salient RVIN which cannot be removed by BM3D can be eliminated in this method, some image details are still lost because of the miss-detection of the image details as the impulse noise.

In order to overcome this problem, a mixed-noise removal method is proposed by adding the interpolation process before the impulse noise detection to the conven-

tional mixed-noise removal method described in [85] leading to preserve the distorted image details such as edges and textures. Although DWM filter is very effective for the removal of RVIN, it tends to miss-detect the image details and over-smooth them in removing the mixed noise. Therefore, the input noisy image is initially interpolated based on multi-surface fitting for single frame [86]. Interpolation technique can significantly reduce the miss-detection of the impulse noise as well as accurately detect the presence of impulse noise in the noisy image. Moreover, the output of DWM filter is down-sampled to reconstruct the original image. The remaining parts of the proposed mixed-noise removal method are the same as the conventional method. The mixed noise model used in this dissertation, characteristics and mathematical approach of DWM filter, methodology of interpolation scheme, the block diagram of the proposed method, and the quantitative and qualitative comparison of the proposed method with the conventional denoising methods will be specifically explained in the following sections.

3.2 Mixed Noise Model

In this chapter, the removal of mixed noise which is a combination of Additive White Gaussian Noise (AWGN) and Random-Valued Impulse Noise (RVIN) is considered. The mixed noise model will be presented in this portion. Firstly, $\boldsymbol{x}$ is represented as a noise free image and the noisy inspection of $\boldsymbol{x}$ corrupted by AWGN as $\boldsymbol{y_G}$ and RVIN as $\boldsymbol{y_I}$, respectively. $x(i,j)$ is the pixel value at the location (i,j) in $\boldsymbol{x}$. In this mixed noise model, all the scalars are represented with small case normal font letters (e.g., x, y_G, and y_I) to denote the intensity values in the image. The lower case and bold face letters (e.g., $\boldsymbol{x}$, $\boldsymbol{y_G}$ and $\boldsymbol{y_I}$) are reserved for matrices. A corrupted pixel's value by AWGN is the sum of the original pixel value and the noise which follows Gaussian distribution. The Gaussian noisy image with corrupted pixel $y_G(i,j)$ in $\boldsymbol{y_G}$ can be created by using the following formula

$$y_G(i,j) = x(i,j) + z(i,j) \tag{3.1}$$

where $z(i,j)$ is the white Gaussian noise whose variance is σ^2. Alternatively, a corrupted pixel by RVIN is replaced by a random value. Namely, a corrupted pixel $y_I(i,j)$ in $\boldsymbol{y_I}$ is modeled as

$$y_I(i,j) = \begin{cases} n(i,j) & \text{with probability } p \\ x(i,j) & \text{with probability } 1-p \end{cases} \tag{3.2}$$

where $n(i,j)$ represents an impulse noise which has a random value in the range of $[0, 255]$, and p is the impulse noise density. Therefore, a noisy pixel $y(i,j)$ corrupted by the mixed noise composed of AWGN and RVIN is modeled as

$$y(i,j) = \begin{cases} n(i,j) & \text{with probability } p \\ x(i,j) + z(i,j) & \text{with probability } 1-p. \end{cases} \tag{3.3}$$

3.3 DWM Filter

Directional Weighted Median (DWM) filter [64] is utilized to remove the impulse noise because it is one of the denoising methods for RVIN. Similarly to the conventional impulse noise removal methods, there are two processes in DWM filter namely: detection process and removal process.

3.3.1 Detection Process

In this process, DWM detects the pixels corrupted by impulse noise. The edges aligned with four main directions are mainly concentrated on as shown in Fig. 3.1. A set of coordinates aligned with the k^{th} direction centered at $(0,0)$ is represented by $S_k (k = 1 \text{ to } 4)$, i.e.,

$$\begin{aligned} S_1 &= \{(-2,-2), (-1,-1), (1,1), (2,2)\}, \\ S_2 &= \{(0,-2), (0,-1), (0,1), (0,2)\}, \\ S_3 &= \{(2,-2), (1,-1), (-1,1), (-2,2)\}, \\ S_4 &= \{(-2,0), (-1,0), (1,0), (2,0)\}. \end{aligned} \tag{3.4}$$

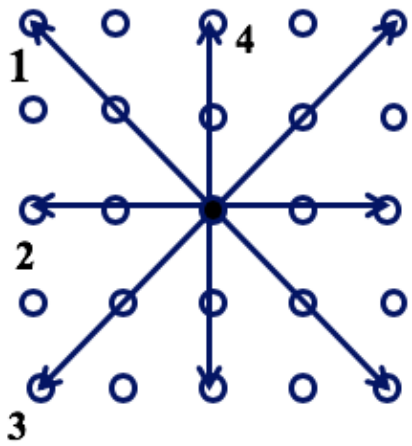


Figure 3.1: Four directions for impulse detection

Firstly, the sum of the absolute difference between $\boldsymbol{y_I}(i+s, j+t)$ and $\boldsymbol{y_I}(i,j)$ in the noisy image $\boldsymbol{y_I}$ corrupted by RVIN with $(s,t) \in S_k$ which is defined by $d_{i,j}^{(k)}$ is calculated for each direction in a 5x5 window centered at (i,j). The weighted difference between the center pixel and the nearest neighboring pixels for the four direction indexes are then evaluated. Namely,

$$d_{i,j}^{(k)} = \sum_{(s,t)\in S_k} \omega_{s,t} |\boldsymbol{y_I}(i+s, j+t) - \boldsymbol{y_I}(i,j)|, 1 \leq k \leq 4 \tag{3.5}$$

where

$$\omega_{s,t} = \begin{cases} 2 & (s,t) \in \Omega^3 \\ 1 & \text{otherwise} \end{cases} \tag{3.6}$$

and $\Omega^3 = \{(s,t) : -1 \leq s,t \leq 1\}$. After that, the smallest value for the four directions $r_{i,j}$ is utilized to detect the impulse noise, which can be denoted as

$$r_{i,j} = \min\{d_{i,j}^{(k)} : 1 \leq k \leq 4\}. \tag{3.7}$$

Moreover, whether the target pixel is a noisy pixel or not can be identified by em-

ploying a threshold T as

$$y(i,j) \text{ is a } \begin{cases} \text{noisy pixel}, & r_{i,j} \geq T \\ \text{noise-free pixel}, & \text{otherwise} \end{cases}. \tag{3.8}$$

3.3.2 Removal Process

After detecting the impulse noise, the DWM filter employs the weighted median filter to restore the noisy pixels. The weighted median filter can be formulated as

$$m_{i,j} = \text{median}\{\omega'_{s,t} \diamond \boldsymbol{y_I}(i+s, j+t) : (s,t) \in W_m\} \tag{3.9}$$

where the operator $\diamond$ represents repetition operation, and

$$\omega'_{s,t} = \begin{cases} 2, & (s,t) \in S_{l_{i,j}} \\ 1, & \text{otherwise} \end{cases} \tag{3.10}$$

$$l_{i,j} = \arg\min_k \{\sigma_{i,j}^{(k)} : k = 1 \text{ to } 4\} \tag{3.11}$$

where $\sigma_{i,j}^{(k)}$ is the standard deviation of gray-level values for all $\boldsymbol{y_I}(i+s, j+t)$ with $(s,t) \in S_k (k = 1 \text{ to } 4)$. Finally, the output of the DWM filter can be calculated as

$$u_{i,j} = \alpha_{i,j} y(i,j) + (1 - \alpha_{i,j}) m_{i,j} \tag{3.12}$$

where

$$\alpha_{i,j} = \begin{cases} 0, & r_{i,j} > \text{T} \\ 1, & r_{i,j} \leq \text{T} \end{cases} \tag{3.13}$$

In order to achieve the high accuracy of detection, the above method is applied recursively and iteratively depending on the threshold value. The threshold is defined

as

$$\begin{aligned} T_{n+1} &= 0.8 \cdot T_n, 0 \leq n \leq N_{max}, \\ T_0 &= 510 \end{aligned} \tag{3.14}$$

where T_0 is the initial threshold, and T_n is the threshold in the n^{th} step.

3.4 Interpolation

Image interpolation aims to achieve the high quality image with fast computation time from single frame or multi-frame low resolution (LR) images based on the image upsampling technologies. In this section, how to interpolate the image for single frame using multi-surface fitting [86] will be mentioned. The main objectives are to improve the reliability of each local function and calculate the weight λ_i of each local function because the first one reduces the blurring effects and the second one minimizes the fitting errors. The details of this process will be explained in the following subsections.

3.4.1 Local Functions Estimation

In order to obtain a target high resolution (HR) pixel p_H , every local function Γ_i can be estimated at each known low resolution (LR) pixel in the image. The function $\Gamma_i(x, y)$ centered at the LR pixel p_i can be evaluated by using the following equation

$$\begin{aligned} \Gamma_i(x, y) = f(x_i, y_i) + \alpha_x(x - x_i) + \alpha_y(y - y_i) \\ + \alpha_{xx}(x - x_i)^2 + \alpha_{yy}(y - y_i)^2. \end{aligned} \tag{3.15}$$

where (x_i, y_i) is the coordinates of p_i , $f(x_i, y_i)$ is the known low resolution pixel values at the coordinates x_i and y_i, and $\alpha_x, \alpha_y, \alpha_{xx}, \alpha_{yy}$ are the parameters of the local function Γ_i. These parameters $\{\alpha_x, \alpha_y, \alpha_{xx}, \alpha_{yy}\}$ are computed by assigning the values and the coordinates of the neighboring pixels to Eq. 3.15 and the following

simultaneous equations are obtained.

$$\begin{cases} f(x_i, y_i) + 2\alpha_x + 4\alpha_{xx} = f(x_i + 2, y_i) \\ f(x_i, y_i) - 2\alpha_x + 4\alpha_{xx} = f(x_i - 2, y_i) \\ f(x_i, y_i) + 2\alpha_y + 4\alpha_{yy} = f(x_i, y_i + 2) \\ f(x_i, y_i) - 2\alpha_y + 4\alpha_{yy} = f(x_i, y_i - 2) \end{cases} \tag{3.16}$$

They are obtained by solving Eq. 3.16 as follows.

$$\begin{cases} \alpha_x = \frac{f(x_i+2,y_i)-f(x_i-2,y_i)}{4} \\ \alpha_y = \frac{f(x_i,y_i+2)-f(x_i,y_i-2)}{4} \\ \alpha_{xx} = \frac{f(x_i+2,y_i)+f(x_i-2,y_i)-2f(x_i,y_i)}{8} \\ \alpha_{yy} = \frac{f(x_i,y_i+2)+f(x_i,y_i-2)-2f(x_i,y_i)}{8} \end{cases} \tag{3.17}$$

The directional weights for horizontal and vertical factors are applied in each local region to prevent the cause of fitting errors in the edge regions of the image by modifying Eq. 3.15.

$$\begin{aligned} \Gamma_i(x, y) = f(x_i, y_i) &+ 2\omega_{h,i}\{\alpha_x(x - x_i) + \alpha_{xx}(x - x_i)^2\} \\ &+ 2\omega_{v,i}\{\alpha_y(y - y_i) + \alpha_{yy}(y - y_i)^2\}. \end{aligned} \tag{3.18}$$

where $\omega_{h,i}$ and $\omega_{v,i}$ are the directional weights for horizontal and vertical factors in the local function Γ_i. The target HR pixels with diamond shape can be estimated by interpolating the known four neighboring LR pixels which consists of one black circle-shaped and three blue circle-shaped pixels. The remaining square-shaped HR pixels are interpolated by the two neighboring LR pixels. The interpolation scheme for calculating the local functions at every known LR pixel in the image is illustrated in Fig. 3.2.

The interpolation quality may be degraded due to the large differences of directional weights in the edge and texture regions. Therefore, the directional weights for

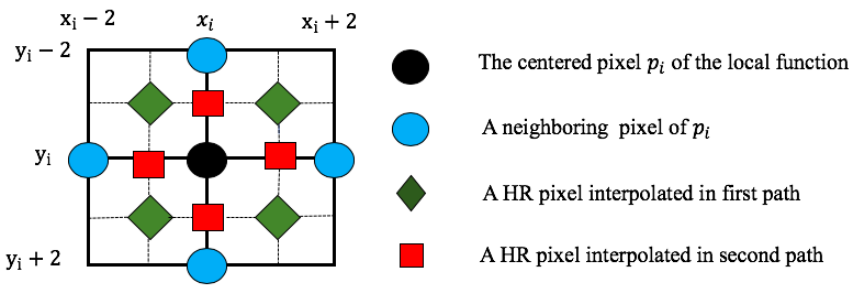


Figure 3.2: Interpolation scheme to calculate the local function Γ_i centered at p_i.

vertical and horizontal factors can be respectively computed by using the following equations.

$$\begin{aligned} d_{v,i} = & |f(x_i, y_i + 2) + f(x_i, y_i - 2) - 2f(x_i, y_i)| \\ & + \left| \begin{array}{l} \{f(x_i, y_i + 4) - f(x_i, y_i - 4)\} \\ *\{f(x_i + 2, y_i + 4) - f(x_i - 2, y_i - 4)\} \\ *\{f(x_i - 2, y_i + 4) - f(x_i + 2, y_i - 4)\} \end{array} \right|^{\frac{1}{3}} \end{aligned} \tag{3.19}$$

$$\begin{aligned} d_{h,i} = & |f(x_i + 2, y_i) + f(x_i - 2, y_i) - 2f(x_i, y_i)| \\ & + \left| \begin{array}{l} \{f(x_i + 4, y_i) - f(x_i - 4, y_i)\} \\ *\{f(x_i + 4, y_i + 2) - f(x_i - 4, y_i - 2)\} \\ *\{f(x_i + 4, y_i - 2) - f(x_i - 4, y_i + 2)\} \end{array} \right|^{\frac{1}{3}} \end{aligned} \tag{3.20}$$

$$\omega_{v,i} = \frac{1}{d_{v,i} + \varepsilon_1}, \omega_{h,i} = \frac{1}{d_{h,i} + \varepsilon_1} \tag{3.21}$$

where ε_1 is a constant parameter to control when both $d_{v,i}$ and $d_{h,i}$ are small values. The fitting error between the center pixel and its nearest neighboring pixels depends on the first terms of Eq. 3.19 and Eq. 3.20 because the 1-D second derivative at the coordinate of the local functions (x_i, y_i) is represented as these terms. When these terms are larger, the fitting error may be bigger. To distinguish the regions between edges and textures, the second terms of Eq. 3.19 and Eq. 3.20 is utilized because they represent the geometric means of the differences for three directions. The three differences are almost the same in the edge regions and random in the

texture regions. Hence, the difference of the geometric mean of these directions in vertical and horizontal components is large at edge regions and almost zero at texture regions. How much important the consideration of second term in each local region of the image is visualized as shown in Fig. 3.3. The interpolated result without the second term produces more smoothness in the edge region of the image. Eventually, the reliable local functions Γ_i can be achieved depending upon the directional weights.

(a) LR image (b) With the second term (c) Without second term

Figure 3.3: Visual comparison on the effect of second term in the strong edge image

3.4.2 Weighted Mean Calculation

It is also necessary to calculate the weighted mean for obtaining the final estimation of the target pixel p_H as follows

$$\hat{f}(p_H) = \frac{\sum_{i=1}^{K} \lambda_i \Gamma_i(x_H, y_H)}{\sum_{i=1}^{K} \lambda_i} \tag{3.22}$$

where $\hat{f}(p_H)$ is the final estimation of HR target pixel p_H at (x_H, y_H) , K is the number of local functions and λ_i is the weight parameter for the estimation in a local function $\Gamma(x_H, y_H)$. The weight λ_i can be evaluated depending on the weighted differences

$$
\begin{aligned}
d_i = \omega_{h,i}\{|f(x_i+2,y_i)-f(x_i,y_i)|+|f(x_i-2,y_i)-f(x_i,y_i)|\} \\
+\omega_{v,i}\{|f(x_i,y_i+2)-f(x_i,y_i)|+|f(x_i,y_i-2)-f(x_i,y_i)|\}
\end{aligned}
\tag{3.23}
$$

$$
\lambda_i = \frac{1}{d_i+\varepsilon_2} \tag{3.24}
$$

where d_i is the sum of the weighted differences between the center pixel p_i and the four neighbor known pixels, and ε_2 is a constant parameter to avoid division by zero. If the difference d_i is smaller, the fitting error of the local function $\Gamma_i(x,y)$ will be smaller.

3.5 Proposed Method

Many mixed-noise removal methods are typically detection-based methods that is detecting the impulse noise pixels from the noisy image followed by removing the noise. Although the mixed-noise removal method which is a combination of DWM and BM3D filtering sufficiently suppresses the noise after conducting the whole process, some image details are deteriorated because of the miss-detection of the image details as RVIN in DWM filter. Therefore, the detection of the impulse noise plays an important role in a mixed-noise removal method to restore the image details.

In order to handle this problem, a mixed-noise removal method is proposed in this section. There are two steps in the proposed method. The first step is to estimate the denoised image from the input noisy image. The second step is to obtain the restored image by utilizing the pre-estimated image from the first step while preserving the important image structures. The core of this proposed method is that the input noisy image is firstly interpolated before the impulse noise detection in DWM filter to preserve the image details. Due to the effect of interpolation, the upscaled DWM output is also needed to down-sample. However, the remaining parts of the proposed noise suppression method for mixed noise are the same as the conventional two-step mixed noise removal method. The block diagram of the proposed method for removal

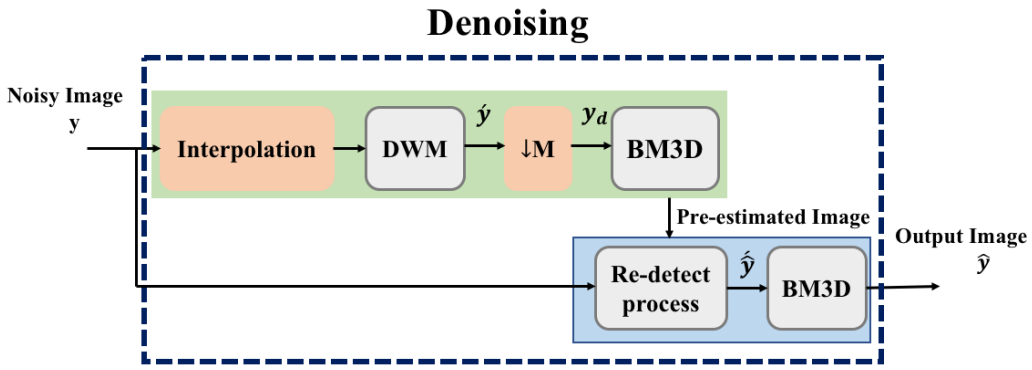


Figure 3.4: Block diagram of the proposed method

of mixed-noise is shown in Fig. 3.4. The first step is highlighted in green block and the second step is described in blue block to be clearly seen.

3.5.1 Interpolation in Mixed Noise Removal

The first step is composed of interpolation, DWM filter, downsampling and BM3D. In this section, the effectiveness of interpolation for the removal of impulse noise and the accuracy of impulse noise detection by using interpolation are considered. In order to verify the first consideration, RVIN (impulse noise density levels 20 and 40 are used in this approach) is added to some natural images and the test images in which the edges can be clearly seen to obtain the noisy image. Then, the reference image and the noisy image are simultaneously up-sampled by using some interpolation methods such as bicubic interpolation [87] and Interpolation for Single Frame using Multi-surface Fitting (ISFMF) [86]. The impulse noise rate without interpolation is the ratio of thresholding on the absolute difference between the reference image and the noisy image to the total number of pixels in the reference image. Similarly, the influence of impulse noise in the interpolated image can be formulated by the ratio of thresholding on the absolute difference between the interpolated reference image and the interpolated noisy image to the total number of pixels in the interpolated reference image.

To be clearly seen, the comparisons of impulse noise rate between without interpolation and using interpolation techniques including bicubic [87] and ISFMF [86]

are shown in Fig. 3.5 and Fig. 3.6, respectively for the test image with strong edge and some natural images (for instance *Lena* image in this case). Experiments are conducted on these images with different impulse noise density levels and different threshold values to estimate the impulse noise rate. The solid line represents the results for the impulse noise density level 20% and the dashed-line demonstrates the impulse noise level 40%.

As can be observed, the impulse noise rate using ISFMF is the lowest among them for both noise levels except for the high threshold values. This is because bicubic interpolation technique considers the same filter in each local region and does not consider the noise and local structure. In contrast, ISFMF considers the different directional weights for vertical and horizontal factors in the local region. The reliability of the local function can be improved by utilizing the different weights in each local region. In the vertical direction, the large difference of the three pixel values gives a small weight. In the horizontal direction, the small difference of the three pixels generates a large weight. If the impulse noise is located in the vertical direction of each local function, the weight of noise can be reduced. Therefore, interpolation using ISFMF is more effective in removing the impulse noise than bicubic interpolation because the rate of impulse noise can be significantly attenuated. When the threshold value is gradually increased, the number of impulse noise is slightly decreased. By inspiration to this concept, the mixed-noise image should be initially interpolated before the detection of impulse noise to reduce the impulse noise rate. Hence, the impulse noise can be efficiently removed as well as the image details can be well preserved.

For the second consideration, some experiments are conducted to detect the accuracy of impulse noise in some natural images (512×512-sized) with different noise levels. Table 3.1 presents the comparison of true-detected, miss-detected and undetected pixels in the noisy image between using DWM filter only and DWM filter with interpolation. True-detected pixels are the correct estimation of impulse noise pixel values that locate at the correct position. Miss-detected pixels define the estimation of noisy pixels although they are actually the original pixel values. The

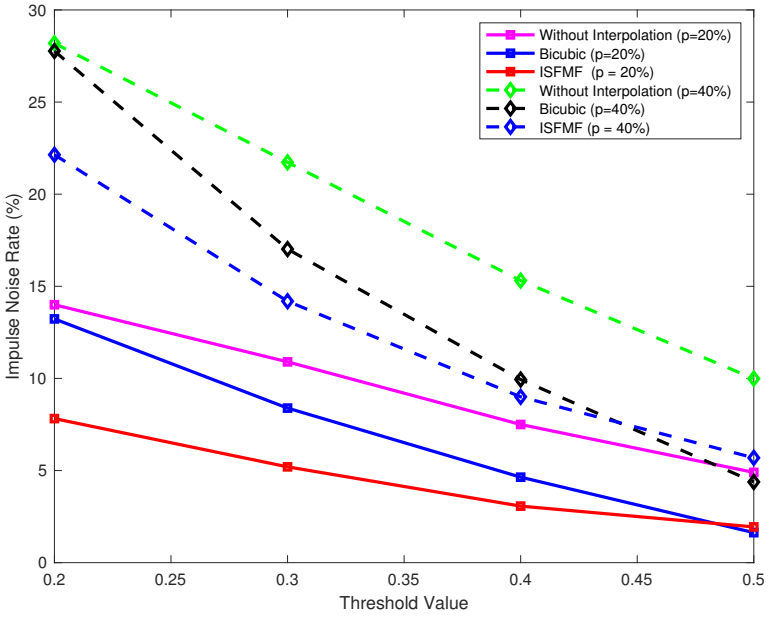


Figure 3.5: Reduction of impulse noise rate in 100×100-sized *Test* image corrupted by $p = 20\%$ and $p = 40\%$

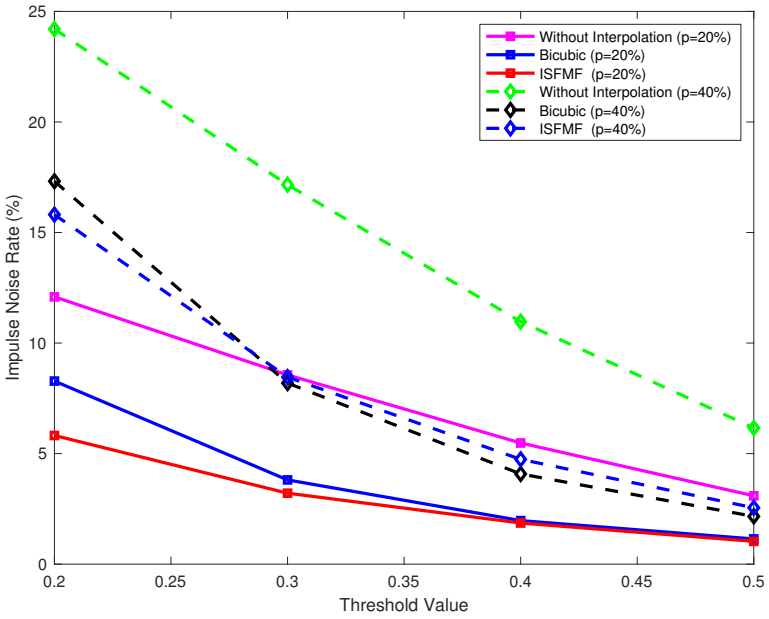


Figure 3.6: Reduction of impulse noise rate in 512×512-sized *Lena* image corrupted by $p = 20\%$ and $p = 40\%$

Table 3.1: Accuracy of impulse noise detection due to the effect of interpolation

Image(512×512)	Impulse Noise level (p)	DWM			Interpolation+DWM		
		True-detect	Miss-detect	Un-detect	True-detect	Miss-detect	Un-detect
Lena	10	8060	0	18327	17035	283	9352
	20	14093	0	38433	33240	536	19286
	30	18681	1	60062	48925	1139	29818
Barbara	10	8093	14	18294	16625	3284	9762
	20	14278	26	38248	32557	4287	19969
	30	19134	40	59609	48229	5511	30514
Hill	10	8219	1	18168	16933	232	9454
	20	14316	1	38210	33186	498	19340
	30	18951	3	59792	48891	1129	29852

un-detected pixels are the estimated original pixel values that are indeed the noisy pixels. As can be seen, the addition of interpolation to the DWM filter increases the number of true-detected and miss-detected pixels in the detection of impulse noise although the un-detected pixels are decreased. The accuracy of impulse noise detection is significantly improved due to the increase in true-detected pixels. The increment in miss-detection of impulse noise pixels tends to cause the low PSNR value in the interpolated DWM output. But, the miss-detected pixels can be removed by BM3D because they are actually Gaussian noisy pixels that are miss-detected as the impulse noise. Besides, the number of miss-detected pixels are increased based on upscaling the noisy image and the impulse noise density level because the random-valued impulse noise is not so different from its neighborhood pixels. However, BM3D is the Gaussian noise removal method and is affected by un-detected pixels. Therefore, the higher PSNR values can be obtained in the first step of the proposed mixed noise reduction method with interpolation than the conventional two-step mixed noise removal method [85].

3.5.2 Impulse Noise Removal

After interpolation, the DWM filter mentioned in Sec. 3.3 is applied to the upsampled noisy image to detect and remove RVIN because it enhances the suppression of impulse noise with high noise level. For DWM filter, the initial value T_0 of the threshold parameter T is selected to 500 and the number of iterations is 6 to avoid over-smoothing image details based on PSNR value. The greater the threshold value,

the more corrupted pixels and the larger the number of iterations. Moreover, the neighboring window size of each pixel is set as 5×5 to prevent from decreasing the quality of the reconstructed image in the proposed mixed noise removal method.

3.5.3 Down-sampling

Down-sampling is the reduction of image size by using an arbitrary factor in both horizontal and vertical directions. Normally, the signal is needed to pre-filter before down-sampling to avoid aliasing. In the proposed method, the output of DWM filter should be down-sampled by using scaling factor to reconstruct the original image due to the effect of interpolation process. The down-sampled image can be calculated by using the following equation.

$$y_d(m,n) = \acute{y}(K(m-1)+1, K(n-1)+1) \tag{3.25}$$

where K is the downsampling factor.

3.5.4 Gaussian Noise Removal

BM3D filtering is used to remove the remaining noise because it is an effective Gaussian noise removal method. Even though BM3D cannot successfully remove the mixed noise, the remaining noise can be eliminated because of its robustness.

Although almost all noises can be removed in the first step, some important image details are still lost. Despite setting the significant parameters to reduce the miss-detection of the impulse noise pixels in the noisy image by observing the best PSNR values, it is unable to preserve some image details because of the detection process of DWM filter. However, it is very difficult to detect most corrupted pixels without miss-detection of image details in the removal of RVIN. The precision of detection strategy may be decreased because the DWM filter cannot remarkably detect the mixed noise corrupted by AWGN. Therefore, this problem will be solved in the second step.

3.5.5 Re-detect Process and BM3D

The second step is a combination of re-detect process and BM3D. After conducting some experiments on several test images with different Gaussian and impulse noise levels in the first step, some important image structures are still eliminated although the mixed-noise can be efficiently suppressed. Hence, an effective detection method for RVIN is addressed to preserve the image details degraded in the first step. To do this, the evaluation error $e(i,j)$ which is the absolute difference between the input noisy image and the output of the first step is firstly defined as

$$e(i,j) = |\tilde{y}(i,j) - y(i,j)|. \tag{3.26}$$

Whether the current pixel is a noisy or noise-free pixel can then be identified by thresholding the error $e(i,j)$. Furthermore, the noisy pixel is replaced with the pixels in $\tilde{\boldsymbol{y}}$. Therefore, a pixel value $\hat{y}^{'}(i,j)$ in the output $\hat{\boldsymbol{y}}^{'}$ of this process is indicated as follows.

$$\hat{y}^{'}(i,j) = \begin{cases} \tilde{y}(i,j) & e(i,j) > \tau \\ y(i,j) & \text{otherwise} \end{cases} \tag{3.27}$$

where τ is the threshold parameter. The threshold parameter τ is set to detect the significant noisy pixels which BM3D cannot remove through many experimental tests because it plays an important role in the detection process. The parameter τ is chosen three times of standard deviation of Gaussian noise according to the conducted experimental results. After removing the impulse noise accurately, BM3D is again exploited for the removal of remaining noise from $\hat{\boldsymbol{y}}^{'}$ in the same manner as the first step and the reconstructed output image is obtained eventually.

3.6 Experimental Results

In order to evaluate the performance of mixed-noise removal method with interpolation, the experiments are conducted on 512×512-sized eleven natural images. The peak signal-to-noise ratio (PSNR) values are used in this implementation as a quanti-

tative metric. All the experiments are conducted in MATLAB (R2018b) environment on 2.2 GHz Intel Core i7 processor with 8GB 1600 MHz DDR3 memory.

3.6.1 Parameters Setting

The basic parameters of the proposed method are set as follows: the upscaling factor $M = 2$ to upsample the input noisy image for image enlargement; the specified window size is 5×5 to calculate the four direction indexes in DWM filter; the patch size is set as 9×9; the patch step size is fixed as 3; the neighborhood window size is set as 39×39 in both steps of BM3D filtering. The maximum number of patches are 16 in hard-thresholding and 32 in wiener-filtering, respectively. The input noisy image is symmetrically padded depending on the patch size to handle the image border in this consideration. Simulations are carried out on 11 widely used images corrupted by Gaussian coupled with random-valued impulse noise with $\sigma = 5$, $\sigma = 10$, $\sigma = 15$ $\sigma = 20$, $\sigma = 25$ and $p = 5\%, p = 10\%, p = 15\%$, $p = 20\%$, $p = 25\%$, respectively.

3.6.2 Quantitative and Visual Evaluation

Table 3.2, Table 3.3 ,Table 3.4, Table 3.5 and Table 3.6 describe the comparison of quantitative measurements of the mixed noise removal method using ISFMF [86] in the interpolation process with some denoising methods including BM3D [24], DWM filter [64], two-step mixed noise removal method (TSMNR) [85], and the replacement of bicubic interpolation [87] in the proposed method on 11 extensively used natural images corrupted by Gaussian noise with $\sigma = 5, 10, 15, 20, 25$ and RVIN with noise level $p = 5\%, 10\%, 15\%, 20\%, 25\%$ in terms of PSNR, respectively. For more convenient comparison, the best values are highlighted in bold. According to the experimental results, the PSNR values of DWM filter [64] are gradually decreased when the impulse noise level is increased. Although BM3D [24] has a weakness in removing the impulse noise, the presence of Gaussian noise in the mixed noise image can be efficiently attenuated. The PSNR values of BM3D will be efficiently improved if Gaussian noise level is maximized.

Table 3.2: Restoration results for mixed noise removal in terms of PSNR [dB] ($\sigma = 5$)

Images(512×512)	Airplane	Barbara	Boats	Bridge	Goldhill	Houses	Lena	Mandrill	Milkdrop	Pepper	Sailboat	Avg.
Noise Level	$p = 5\%$											
BM3D [24]	21.17	22.16	22.59	21.90	22.39	21.37	22.77	22.58	22.21	22.22	21.38	25.92
DWM [64]	30.56	24.07	28.02	24.92	29.55	21.93	31.93	22.23	36.91	32.77	27.88	28.25
TSMNR [85]	31.19	24.29	28.47	25.10	30.27	22.18	32.75	22.21	38.05	33.48	28.31	28.75
Proposed(Bicubic)	34.28	26.68	31.40	27.79	32.39	25.76	34.96	24.54	37.93	34.53	31.03	30.98
Proposed(ISFMF)	**35.71**	**28.59**	**32.79**	**28.22**	**33.31**	**27.54**	**36.19**	**25.70**	**38.52**	**35.29**	**32.04**	**32.17**
Noise Level	$p = 10\%$											
BM3D [24]	18.13	18.96	19.42	18.83	19.18	18.37	19.39	19.59	18.85	19.01	18.32	18.91
DWM [64]	30.24	23.94	27.79	24.75	29.37	21.74	31.61	22.12	36.18	32.39	27.70	27.99
TSMNR [85]	30.91	24.19	28.26	24.95	30.09	22.01	32.47	22.12	**37.57**	33.16	28.15	28.53
Proposed(Bicubic)	32.48	26.10	30.33	26.48	31.38	24.74	33.55	24.09	35.64	33.17	29.78	29.79
Proposed(ISFMF)	**34.43**	**27.74**	**31.87**	**27.53**	**32.60**	**26.37**	**35.14**	**25.14**	37.42	**34.49**	**31.16**	**31.88**
Noise Level	$p = 15\%$											
BM3D [24]	16.36	17.16	17.61	17.07	17.41	16.64	17.59	17.85	17.05	17.23	16.57	17.14
DWM [64]	29.86	23.79	27.50	24.54	29.16	21.48	31.26	22.00	35.34	31.88	27.48	27.66
TSMNR [85]	30.57	24.05	27.98	24.76	29.89	21.77	32.18	22.03	**36.85**	32.71	27.96	28.25
Proposed(Bicubic)	29.82	25.27	28.87	25.49	29.82	23.62	31.59	23.57	32.36	30.94	28.05	28.13
Proposed(ISFMF)	**32.54**	**26.79**	**30.68**	**26.67**	**31.53**	**25.15**	**33.72**	**24.54**	35.52	**32.97**	**29.86**	**29.99**
Noise Level	$p = 20\%$											
BM3D [24]	15.08	15.91	16.35	15.81	16.13	15.39	16.32	16.59	15.79	15.94	15.30	15.87
DWM [64]	29.39	23.63	27.19	24.27	28.88	21.15	30.79	21.87	34.29	31.28	27.18	27.27
TSMNR [85]	27.02	23.91	27.69	24.52	29.64	21.44	**31.78**	21.93	**35.83**	**32.19**	27.69	27.89
Proposed(Bicubic)	**30.14**	24.28	27.24	24.37	27.89	23.62	29.34	22.99	20.09	28.46	26.03	26.29
Proposed(ISFMF)	29.91	**25.74**	**29.12**	**25.61**	**29.86**	**23.86**	31.68	**23.88**	32.27	30.86	**28.05**	**28.26**
Noise Level	$p = 25\%$											
BM3D [24]	14.08	14.91	15.37	14.82	15.14	14.39	15.32	15.61	14.81	14.95	14.32	14.89
DWM [64]	28.88	23.45	26.84	23.96	28.54	20.79	32.28	21.72	33.26	30.58	26.81	26.83
TSMNR [85]	**29.69**	23.76	**27.36**	24.23	**29.33**	21.08	**31.30**	21.81	**34.78**	**31.56**	**27.36**	**27.48**
Proposed(Bicubic)	24.46	23.14	25.63	23.13	25.93	21.31	27.10	22.34	26.46	26.09	24.09	24.52
Proposed(ISFMF)	27.01	**24.51**	27.33	**24.31**	27.79	**22.55**	29.25	**23.13**	29.09	28.23	25.99	26.29

The use of bicubic interpolation in the proposed method gives larger PSNR values than the other conventional mixed noise reduction methods , especially in the images corrupted by the impulse noise levels less than 25%. However, the PSNR values of the mixed noise elimination method using ISFMF interpolation technique are the highest for almost all noise levels. In particular, the proposed mixed noise removal method is quite effective for the images which are rich in regular and repetitive structures such as *Barbara*, *Bridge*, *Houses* and *Mandrill* images in the case of Gaussian plus random-valued impulse noise with $\sigma = 25$ and $p = 25\%$. The average PSNR values of the proposed method using ISFMF is lower than the two-step conventional mixed noise removal method at low Gaussian noise levels $\sigma = 5$ and $\sigma = 10$ with strong impulse noise level $p = 25\%$ because the number of miss-detected pixels are highly increased. When the Gaussian noise level is slightly increased, the number of miss-detected pixels are decreased and the number of un-detected pixels are increased. The un-detected pixels can be successfully removed by BM3D. Moreover, DWM fails

Table 3.3: Restoration results for mixed noise removal in terms of PSNR [dB] ($\sigma = 10$)

Images(512×512)	Airplane	Barbara	Boats	Bridge	Goldhill	Houses	Lena	Mandrill	Milkdrop	Pepper	Sailboat	Avg.
Noise Level	$p = 5\%$											
BM3D [24]	22.63	23.99	24.55	22.79	24.41	22.23	25.27	23.43	24.47	24.37	22.59	23.70
DWM [64]	29.92	23.92	27.65	24.71	29.03	21.86	31.09	22.14	34.97	31.82	27.53	27.69
TSMNR [85]	31.26	24.98	28.69	25.42	30.45	22.71	32.87	22.45	36.89	33.03	28.45	28.84
Proposed(Bicubic)	33.64	27.55	30.99	27.06	31.77	26.26	34.38	24.57	36.92	33.75	30.55	30.68
Proposed(ISFMF)	**34.24**	**29.42**	**31.74**	**27.55**	**32.09**	**27.67**	**34.82**	**25.49**	**37.07**	**34.02**	**30.99**	**31.37**
Noise Level	$p = 10\%$											
BM3D [24]	18.63	19.62	20.24	19.21	19.91	18.69	20.39	20.04	19.79	19.81	18.76	19.55
DWM [64]	29.59	23.79	27.42	24.53	28.84	21.65	30.77	22.03	34.34	31.44	27.32	27.43
TSMNR [85]	30.95	24.81	28.46	25.22	30.22	22.50	32.57	22.34	**36.63**	32.77	28.27	28.61
Proposed(Bicubic)	32.81	26.99	30.37	26.54	31.31	25.42	33.69	24.18	36.11	33.18	29.94	30.05
Proposed(ISFMF)	**33.61**	**28.59**	**31.19**	**27.11**	**31.73**	**26.79**	**34.29**	**25.03**	36.61	**33.64**	**30.52**	**30.83**
Noise Level	$p = 15\%$											
BM3D [24]	16.59	17.46	17.96	17.28	17.73	16.82	17.99	18.09	17.43	17.57	16.79	17.43
DWM [64]	29.18	23.63	27.13	24.32	28.61	21.38	30.41	21.90	33.60	30.94	27.07	27.11
TSMNR [85]	30.62	24.61	28.14	24.99	29.97	22.21	32.26	22.22	**36.22**	32.41	28.08	28.34
Proposed(Bicubic)	31.48	26.31	29.54	25.88	30.60	24.46	32.72	23.76	34.55	32.07	29.03	29.13
Proposed(ISFMF)	**32.67**	**27.70**	**30.56**	**26.57**	**31.25**	**25.78**	**33.60**	**24.55**	35.86	**32.94**	**29.89**	**30.12**
Noise Level	$p = 20\%$											
BM3D [24]	15.24	16.09	16.57	15.97	16.34	15.52	16.56	16.77	16.02	16.16	15.45	16.06
DWM [64]	28.69	23.46	26.81	24.05	28.31	21.05	29.95	21.77	32.71	30.35	26.75	26.72
TSMNR [85]	30.21	24.42	27.84	24.72	29.70	21.85	31.87	22.08	**35.52**	**31.99**	27.82	28.00
Proposed(Bicubic)	29.37	25.46	28.35	25.02	29.39	23.39	31.39	23.28	31.79	30.34	27.59	27.74
Proposed(ISFMF)	**31.13**	**26.77**	**29.59**	**25.85**	**30.39**	**24.62**	**32.49**	**24.01**	34.03	31.78	**28.81**	**29.04**
Noise Level	$p = 25\%$											
BM3D [24]	14.20	15.05	15.54	14.95	15.30	14.51	15.50	15.75	14.98	15.11	14.45	15.03
DWM [64]	28.13	23.27	26.45	23.73	27.94	20.69	29.43	21.61	31.76	29.67	26.34	26.28
TSMNR [85]	**29.78**	24.20	27.52	24.42	**29.38**	21.45	**31.42**	21.92	**34.64**	**31.45**	**27.47**	**27.61**
Proposed(Bicubic)	26.73	24.35	26.95	23.95	27.66	22.18	29.23	22.73	28.90	28.10	25.78	26.05
Proposed(ISFMF)	28.79	**25.65**	**28.28**	**24.86**	29.03	**23.34**	30.75	**23.39**	31.23	29.78	27.25	27.49

to suppress the miss-detected pixels affected by Gaussian noise level. Therefore, the performance evaluation of the proposed method is higher at strong noise levels $\sigma = 15$, $\sigma = 20$ and $\sigma = 25$ than the low noise levels. However, the average values for all test images in the proposed method are superior to the other competitive denoising methods.

Table 3.7 and Table 3.8 compare the number of un-detected and miss-detected pixels in *Barbara* image and *Lena* image degraded by different noise levels using the proposed method with the conventional two-step mixed noise removal method (TSMNR) [85] to detect the presence of impulse noise in the noisy image. The number of miss-detected and un-detected pixels are estimated between the input noisy image and the re-detect process. Un-detect means the estimation of the original pixel values but they are actually noisy. Miss-detected pixels are the estimated noisy pixels although they are indeed the original pixel values. As can be seen in Table 3.7 and Table 3.8, the number of miss-detected pixels in both images are significantly

Table 3.4: Restoration results for mixed noise removal in terms of PSNR [dB] ($\sigma = 15$)

Images(512×512)	Airplane	Barbara	Boats	Bridge	Goldhill	Houses	Lena	Mandrill	Milkdrop	Pepper	Sailboat	Avg.
Noise Level	$p = 5\%$											
BM3D [24]	25.33	26.72	26.75	24.19	27.15	23.77	28.33	24.36	27.85	27.40	24.83	23.81
DWM [64]	29.14	23.71	27.16	24.42	28.39	21.70	30.11	21.99	33.05	30.72	27.06	27.04
TSMNR [85]	31.09	26.47	28.74	25.50	30.16	23.31	32.51	22.89	35.92	32.43	28.38	28.85
Proposed(Bicubic)	32.56	28.57	30.28	26.48	30.86	26.38	33.32	24.59	35.96	32.88	29.71	30.15
Proposed(ISFMF)	**32.84**	**29.88**	**30.66**	**26.72**	**30.98**	**27.26**	**33.49**	**25.17**	**36.01**	**33.04**	**29.91**	**30.54**
Noise Level	$p = 10\%$											
BM3D [24]	20.16	21.26	22.08	20.24	21.81	19.58	22.65	21.12	21.85	21.88	20.06	21.15
DWM [64]	28.78	23.57	26.92	24.24	28.18	21.47	29.77	21.88	32.48	30.33	26.81	26.77
TSMNR [85]	30.75	26.17	28.47	25.28	29.91	23.04	32.21	22.72	35.64	32.19	28.16	28.59
Proposed(Bicubic)	32.02	28.01	29.82	26.11	30.53	25.68	32.86	24.24	35.48	32.49	29.30	29.68
Proposed(ISFMF)	**32.38**	**29.13**	**30.25**	**26.38**	**30.69**	**26.58**	**33.09**	**24.77**	**35.65**	**32.73**	**29.57**	**30.11**
Noise Level	$p = 15\%$											
BM3D [24]	17.23	18.28	19.02	17.86	18.69	17.29	19.22	18.78	18.57	18.61	17.39	18.27
DWM [64]	28.34	23.41	26.62	24.01	27.92	21.20	29.39	21.75	31.82	29.83	26.53	26.44
TSMNR [85]	30.36	25.84	28.16	25.04	29.66	22.69	31.89	22.53	**35.26**	31.85	27.94	28.29
Proposed(Bicubic)	31.22	27.31	29.27	25.65	30.09	24.91	32.24	23.86	34.61	31.82	28.76	29.07
Proposed(ISFMF)	**31.74**	**28.29**	**29.80**	**25.99**	**30.35**	**25.80**	**32.62**	**24.36**	35.11	**32.21**	**29.15**	**29.58**
Noise Level	$p = 20\%$											
BM3D [24]	15.59	16.57	17.18	16.34	16.89	15.83	17.21	17.21	16.62	16.72	15.82	16.54
DWM [64]	27.82	23.24	26.29	23.73	27.59	20.87	28.93	21.61	31.02	29.24	26.17	26.05
TSMNR [85]	29.99	25.51	27.80	24.74	29.38	22.29	31.50	22.35	**34.78**	**31.48**	27.64	27.95
Proposed(Bicubic)	30.03	26.52	28.49	25.05	29.37	23.98	31.34	23.42	32.99	30.75	27.88	28.17
Proposed(ISFMF)	**30.85**	**27.44**	**29.16**	**25.49**	**29.78**	**24.84**	**31.91**	**23.89**	34.04	31.47	**28.45**	**28.85**
Noise Level	$p = 25\%$											
BM3D [24]	14.46	15.39	15.95	15.23	15.68	14.75	15.93	16.08	15.37	15.48	14.72	15.37
DWM [64]	27.21	23.02	25.91	23.40	27.19	20.51	28.41	21.45	30.15	28.56	25.71	25.59
TSMNR [85]	**29.53**	25.14	27.46	24.42	**29.05**	21.84	**31.08**	22.14	**34.07**	**30.99**	27.29	27.55
Proposed(Bicubic)	28.17	25.49	27.49	24.27	28.31	22.93	30.01	22.93	30.67	29.12	26.63	26.91
Proposed(ISFMF)	29.36	**26.45**	**28.29**	**24.81**	28.96	**23.79**	30.79	**23.38**	32.15	30.13	**27.43**	**27.78**

decreased by 50% in the proposed mixed noise reduction method in comparison with TSMNR method. When the number of miss-detected pixels are smaller, the image details are clearer. Therefore, the proposed method with interpolation approach is very effective to preserve some fine details in the image. According to the increase in Gaussian noise level, the number of miss-detected pixels are decreased and the number of un-detected pixels are efficiently increased. Although the number of un-detected pixels are increased, BM3D can remove the remaining noise.

Figure. 3.7 shows the enlarged parts of the denoised *Airplane* image corrupted by mixed Gaussian-impulse noise with $\sigma = 15$ and $p = 20\%$ to compare the visual quality. The conventional two-step mixed noise removal method blurs the image edges and some image details are lost due to over-smoothing. In contrast, the proposed method can remove the noise drastically as well as preserve most of the image details; especially the numerical value written on the plane. The restoration results of *Barbara* image affected by the mixture of Gaussian and impulse noise with $\sigma = 25$ and $p = 25\%$

Table 3.5: Restoration results for mixed noise removal in terms of PSNR [dB] ($\sigma = 20$)

Images(512×512)	Airplane	Barbara	Boats	Bridge	Goldhill	Houses	Lena	Mandrill	Milkdrop	Pepper	Sailboat	Avg.
Noise Level	$p = 5\%$											
BM3D [24]	27.70	28.31	27.82	24.86	28.39	24.80	29.90	24.52	30.38	29.19	26.38	27.48
DWM [64]	28.29	23.45	26.59	24.08	27.69	21.48	29.12	21.81	31.36	29.62	26.51	26.36
TSMNR [85]	30.69	27.69	28.55	25.28	29.68	23.75	31.86	23.20	35.06	31.84	28.12	28.70
Proposed(Bicubic)	31.55	29.07	29.49	25.79	29.99	26.08	32.32	24.36	35.09	32.09	28.89	29.52
Proposed(ISFMF)	**31.68**	**29.75**	**29.70**	**25.92**	**30.06**	**26.63**	**32.40**	**24.65**	**35.11**	**32.18**	**28.99**	**29.74**
Noise Level	$p = 10\%$											
BM3D [24]	22.37	23.17	23.72	21.41	23.90	20.74	24.96	22.11	23.94	23.98	21.78	22.92
DWM [64]	27.92	23.31	26.35	23.89	27.46	21.25	28.77	21.69	30.83	29.21	26.23	26.08
TSMNR [85]	30.37	27.32	28.26	25.04	29.41	23.43	31.56	22.99	34.77	31.56	27.86	28.42
Proposed(Bicubic)	31.07	28.50	29.12	25.51	29.69	25.50	31.91	24.01	34.66	31.73	28.53	29.11
Proposed(ISFMF)	**31.27**	**29.11**	**29.35**	**25.64**	**29.78**	**26.07**	**32.03**	**24.31**	**34.75**	**31.87**	**28.67**	**29.35**
Noise Level	$p = 15\%$											
BM3D [24]	18.37	19.54	20.54	18.79	20.18	18.09	20.97	19.80	20.14	20.24	18.47	19.56
DWM [64]	27.45	23.15	26.04	23.65	27.19	20.97	28.38	21.56	30.21	28.71	25.92	25.75
TSMNR [85]	29.95	26.86	27.92	24.79	29.13	23.05	31.20	22.76	**34.39**	31.19	27.61	28.08
Proposed(Bicubic)	30.45	27.81	28.65	25.14	29.33	24.85	31.40	23.67	33.98	31.17	28.08	28.59
Proposed(ISFMF)	**30.72**	**28.38**	**28.94**	**25.33**	**29.46**	**25.43**	**31.60**	**23.94**	34.23	**31.40**	**28.29**	**28.88**
Noise Level	$p = 20\%$											
BM3D [24]	16.27	17.42	18.27	17.03	17.87	16.40	18.42	18.00	17.71	17.74	16.49	17.42
DWM [64]	26.90	22.96	25.69	23.36	26.84	20.64	27.90	21.41	29.49	28.13	25.51	25.35
TSMNR [85]	29.54	26.43	27.54	24.51	28.85	22.60	30.82	22.52	**33.87**	**30.79**	27.27	27.70
Proposed(Bicubic)	29.59	27.05	28.04	24.69	28.78	24.07	30.68	23.28	32.82	30.38	27.42	27.89
Proposed(ISFMF)	**30.02**	**27.64**	**28.39**	**24.93**	**28.99**	**24.63**	**30.99**	**23.55**	33.36	30.77	**27.74**	**28.27**
Noise Level	$p = 25\%$											
BM3D [24]	14.95	16.00	16.70	15.76	16.38	15.19	16.73	16.69	16.09	16.17	15.22	15.99
DWM [64]	26.26	22.72	25.29	23.02	26.40	20.28	27.38	21.24	28.66	27.46	25.02	24.89
TSMNR [85]	**29.07**	25.93	27.18	24.19	**28.49**	22.12	**30.40**	22.27	**33.16**	**30.31**	26.87	27.27
Proposed(Bicubic)	28.28	26.11	27.29	24.11	28.02	23.17	29.73	22.85	31.10	29.17	26.52	26.94
Proposed(ISFMF)	28.92	**26.71**	**27.74**	**24.42**	28.36	**23.74**	30.18	**23.11**	31.91	29.75	**26.95**	**27.44**

are illustrated in Fig. 3.8. The image details such as the stripes and thin lines on the scarf are failed to restore in TSMNR method because of over-filtering. But, not only the mixed noises are successfully suppressed but also more accurate textures are reconstructed in the proposed method. The denoised fragments of *Boat* image contaminated by AWNG and RVIN mixed noise ($\sigma = 20$ and $p = 15\%$) are depicted in Fig. 3.9. Some important image features such as text written on the front part of the boat and tiny ropes can be well preserved in the proposed method.

The zoom version of denoised results in *Goldhill* image degraded by $\sigma = 20$ and $p = 10\%$, and *Houses* image corrupted by AWGN and RVIN mixed noise with $\sigma = 10$ and $p = 10\%$ are shown in Fig. 3.10 and Fig. 3.11, respectively. The more accurate brick textures in both images and text written on the wall of house in *Houses* image can be clearly seen in the proposed mixed noise removal method with ISFMF interpolation approach. The restored results of *Lena* image disturbed by AWGN + RVIN ($\sigma = 25$ and $p = 20\%$) and *Pepper* image contaminated by mixed GN and RVIN

Table 3.6: Restoration results for mixed noise removal in terms of PSNR [dB] ($\sigma = 25$)

Images(512×512)	Airplane	Barbara	Boats	Bridge	Goldhill	Houses	Lena	Mandrill	Milkdrop	Pepper	Sailboat	Avg.
Noise Level	$p = 5\%$											
BM3D [24]	28.71	28.55	27.89	24.75	28.44	25.01	30.08	24.13	31.27	29.59	26.77	27.75
DWM [64]	27.45	23.18	25.99	23.72	27.02	21.22	28.16	21.59	29.89	28.56	25.93	25.70
TSMNR [85]	30.19	28.26	28.19	24.89	29.09	23.99	31.18	23.24	34.27	31.23	27.68	28.38
Proposed(Bicubic)	30.68	28.97	28.79	25.19	29.28	25.68	31.43	23.95	34.29	31.37	28.14	28.89
Proposed(ISFMF)	**30.75**	**29.38**	**28.89**	**25.25**	**29.31**	**26.01**	**31.47**	**24.09**	**34.32**	**31.43**	**28.21**	**29.01**
Noise Level	$p = 10\%$											
BM3D [24]	23.94	24.37	24.69	22.16	25.07	21.63	26.31	22.49	25.44	25.29	23.01	24.04
DWM [64]	27.07	23.03	25.75	23.51	26.75	20.98	27.81	21.47	29.41	28.15	25.63	25.41
TSMNR [85]	29.82	27.76	27.88	24.67	28.82	23.64	30.84	22.99	**33.91**	30.91	27.39	28.06
Proposed(Bicubic)	30.21	28.39	28.42	24.92	28.94	25.16	31.01	23.64	33.82	30.99	27.79	28.48
Proposed(ISFMF)	**30.22**	**28.77**	**28.55**	**25.00**	**29.00**	**25.50**	**31.09**	**23.78**	33.88	**31.08**	**27.88**	**28.62**
Noise Level	$p = 15\%$											
BM3D [24]	19.68	20.76	21.74	19.74	21.53	18.98	22.48	20.64	21.51	21.64	19.64	20.76
DWM [64]	26.58	22.85	25.44	23.26	26.45	20.70	27.41	21.34	28.82	27.66	25.29	25.07
TSMNR [85]	29.37	27.21	27.55	24.42	28.52	23.24	30.43	22.76	**33.49**	30.53	27.11	27.69
Proposed(Bicubic)	29.59	27.68	27.98	24.61	28.58	24.56	30.52	23.31	33.14	30.43	27.36	27.98
Proposed(ISFMF)	**29.77**	**28.03**	**28.17**	**24.71**	**28.66**	**24.92**	**30.65**	**23.47**	33.29	**30.58**	**27.50**	**28.16**
Noise Level	$p = 20\%$											
BM3D [24]	17.23	18.47	19.48	17.91	19.05	17.17	19.74	18.92	18.92	18.95	17.43	18.48
DWM [64]	26.01	22.64	25.07	22.95	26.07	20.38	26.94	21.18	28.14	27.09	24.85	24.67
TSMNR [85]	28.91	26.69	27.15	24.13	28.21	22.76	30.03	22.49	**32.87**	**30.08**	26.72	27.28
Proposed(Bicubic)	28.84	26.96	27.42	24.21	28.08	23.83	29.87	22.96	32.05	29.69	26.77	27.34
Proposed(ISFMF)	**29.11**	**27.32**	**27.66**	**24.36**	**28.22**	**24.21**	**30.06**	**23.10**	32.40	29.96	**26.97**	**27.58**
Noise Level	$p = 25\%$											
BM3D [24]	15.71	16.87	17.73	16.54	17.37	15.85	17.84	17.55	17.08	17.16	15.99	16.88
DWM [64]	25.34	22.38	24.67	22.60	25.61	20.02	26.41	21.01	27.36	26.42	24.33	24.19
TSMNR [85]	**28.42**	26.08	26.74	23.83	**27.83**	22.24	**29.56**	22.21	**32.11**	**29.53**	**26.31**	26.80
Proposed(Bicubic)	27.79	26.05	26.75	23.72	27.42	23.03	29.05	22.59	30.63	28.67	25.98	26.52
Proposed(ISFMF)	28.18	**26.42**	**27.03**	**23.91**	27.64	**23.41**	29.32	**22.72**	31.10	29.02	26.27	**26.82**

with $\sigma = 15$ and $p = 15\%$ are illustrated in Fig. 3.12 and Fig. 3.13, respectively. Not only almost all mixed noises in both images are efficiently eliminated but also the features in the smooth regions of *Lena*, and the image details in both edge and smooth regions of *Pepper* image are well restored in accordance with the conducted visual assessments.

Table 3.7: The number of un-detected and miss-detected pixels in *Barbara* image corrupted by mixed noise composed of AWGN and RVIN

[pix]	TSMNR [85]		Proposed Method	
noise	Un-detect	Miss-detect	Un-detect	Miss-detect
$\sigma = 5, p = 20\%$	6819	43029	8076	29983
$\sigma = 5, p = 25\%$	8545	41060	10468	31847
$\sigma = 15, p = 20\%$	18808	11095	19868	6056
$\sigma = 15, p = 25\%$	23626	10547	25354	6606
$\sigma = 25, p = 20\%$	29697	3583	30650	2276
$\sigma = 25, p = 25\%$	37457	3469	38941	2407

The enlarged fragments of *Bridge* image degraded by mixed Gaussian and impulse noise ($\sigma = 10$ and $p = 25\%$) and *Mandrill* image contaminated by a mixture of GN

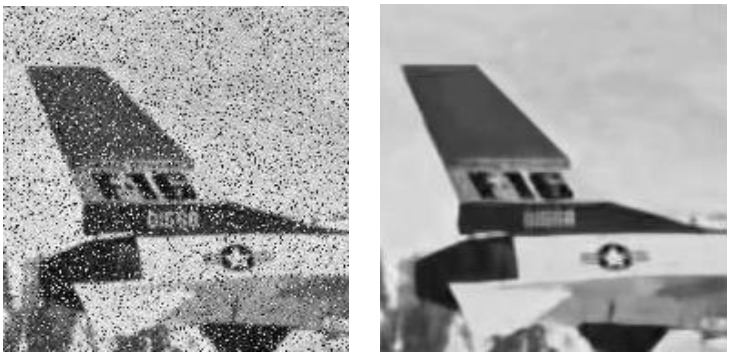
Table 3.8: The number of un-detected and miss-detected pixels in *Lena* image corrupted by mixed noise composed of AWGN and RVIN

[pix]	TSMNR [85]		Proposed Method	
noise	Un-detect	Miss-detect	Un-detect	Miss-detect
$\sigma = 5, p = 20\%$	6319	13300	7164	9862
$\sigma = 5, p = 25\%$	7945	13222	9380	12593
$\sigma = 15, p = 20\%$	18649	2260	19488	1437
$\sigma = 15, p = 25\%$	23427	2271	24806	1777
$\sigma = 25, p = 20\%$	29967	1139	30841	807
$\sigma = 25, p = 25\%$	37809	1104	39042	847

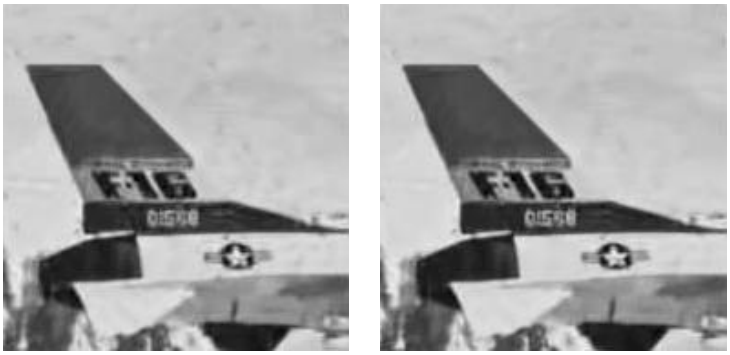
$\sigma = 5$ and RVIN $p = 20\%$ are shown in Fig. 3.14 and Fig. 3.15, respectively. It can be observed that some impulse noise cannot be sufficiently reduced in these figures when the noise density level is strong. However, BM3D can remove Gaussian noise efficiently with high noise intensity level. In addition, the highlighted regions of *Milkdrop* image contaminated by $\sigma = 25$ and $p = 5\%$,and *Sailboat* image corrupted by Gaussian noise $\sigma = 20$ and RVIN $p = 25\%$ are demonstrated in Fig. 3.16 and Fig. 3.17. It is obvious that some image details in both images can be accurately preserved except for poor in removing impulse noise in *Sailboat* image.

Fig. 3.7(c), Fig. 3.8(c), Fig. 3.9(c), Fig. 3.10(c), Fig. 3.11(c), Fig. 3.12(c), Fig. 3.13(c), Fig. 3.14(c), Fig. 3.15(c), Fig. 3.16(c) and Fig. 3.17(c) illustrate the reconstructed results by using DWM filter. DWM filter cannot remove Gaussian noise although RVIN can be effectively eliminated in this filter. Similarly, Fig. 3.7(d), Fig. 3.8(d), Fig. 3.9(d), Fig. 3.10(d), Fig. 3.11(d), Fig. 3.12(d), Fig. 3.13(d), Fig. 3.14(d), Fig. 3.15(d), Fig. 3.16(d) and Fig. 3.17(d) show the denoising results by using BM3D filtering. BM3D also cannot remove the mixed noise due to the effect of RVIN. Moreover, the visual representations of mixed noise removal method using bicubic interpolation [87] instead of ISFMF [86] are shown in Fig. 3.7(f), Fig. 3.8(f), Fig. 3.9(f), Fig. 3.10(f), Fig. 3.11(f), Fig. 3.12(f), Fig. 3.13(f), Fig. 3.14(f), Fig. 3.15(f), Fig. 3.16(f) and Fig. 3.17(f), respectively. The visual measurements of the proposed method using bicubic are not so different from the method with ISFMF. Consequently, after restoring the mixed noise image corrupted with different values of standard variance and impulse noise level by utilizing the proposed method throughout many experiments, it can be seen that the image details in the edge and

(a) Original (b) Noisy (c) DWM filter [64]



(d) BM3D [24] (e) TSMNR [85]



(f) Proposed(Bicubic) (g) Proposed(ISFMF)

Figure 3.7: Restoration results of *Airplane* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 15$ and $p = 20\%$

texture regions can be remarkably preserved as well as the PSNR performance is significantly increased.

(a) Original

(b) Noisy

(c) DWM filter [64]

(d) BM3D [24]

(e) TSMNR [85]

(f) Proposed(Bicubic)

(g) Proposed(ISFMF)

Figure 3.8: Restoration results of *Barbara* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 25$ and $p = 25\%$

3.7 Summary

In this Chapter, a mixed-noise removal method has been proposed to remove the combination of AWGN and RVIN with interpolation technique. There are two steps in the proposed denoising method for mixed-noise. The first step is the integration of interpolation, DWM filter, down-sampling and BM3D filtering to denoise the noisy

(a) Original (b) Noisy (c) DWM filter [64]

(d) BM3D [24] (e) TSMNR [85]

(f) Proposed(Bicubic) (g) Proposed(ISFMF)

Figure 3.9: Restoration results of *Boat* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 20$ and $p = 15\%$

image. The second step is composed of re-detect process and BM3D to reconstruct the image. The main contribution of the proposed method is that the input noisy image is initially interpolated before the detection of impulse noise in DWM filter by using multi-surface fitting for single frame in order to preserve the distorted image details. Interpolating the input noisy image not only reduces the impulse noise rate but also improves the accuracy of impulse noise detection. Therefore, it is very effective for

(a) Original (b) Noisy (c) DWM filter [64]

(d) BM3D [24] (e) TSMNR [85]

(f) Proposed(Bicubic) (g) Proposed(ISFMF)

Figure 3.10: Restoration results of *Goldhill* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 20$ and $p = 10\%$

the removal of RVIN in the proposed method. The output of interpolated DWM filter is down-sampled due to the effect of interpolation. Although almost all noises can be sufficiently reduced in the first step, some image details are still lost. Hence, the detection and removal method of RVIN based on thresholding on the error value between the input noisy image and the pre-estimated image from the first step is applied in the second step to tackle this issue. After the removal of RVIN, BM3D is

(a) Original (b) Noisy (c) DWM filter [64]

(d) BM3D [24] (e) TSMNR [85]

(f) Proposed(Bicubic) (g) Proposed(ISFMF)

Figure 3.11: Restoration results of *Houses* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 10$ and $p = 10\%$

exploited to eliminate the remaining noise. According to the experimental results, the proposed method gives higher PSNR values and more excellent visual quality than the conventional noise suppression methods.

(a) Original (b) Noisy (c) DWM filter [64]

(d) BM3D [24] (e) TSMNR [85]

(f) Proposed(Bicubic) (g) Proposed(ISFMF)

Figure 3.12: Restoration results of *Lena* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 25$ and $p = 20\%$

(a) Original

(b) Noisy

(c) DWM filter [64]

(d) BM3D [24]

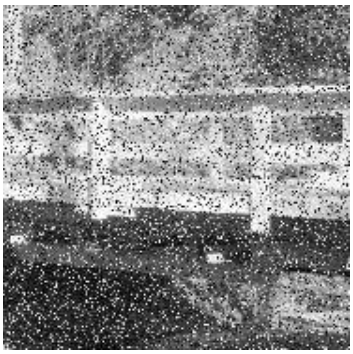
(e) TSMNR [85]

(f) Proposed(Bicubic)

(g) Proposed(ISFMF)

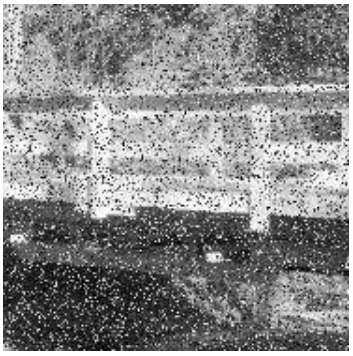
Figure 3.13: Restoration results of *Pepper* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 15$ and $p = 15\%$

(a) Original



(b) Noisy

(c) DWM filter [64]



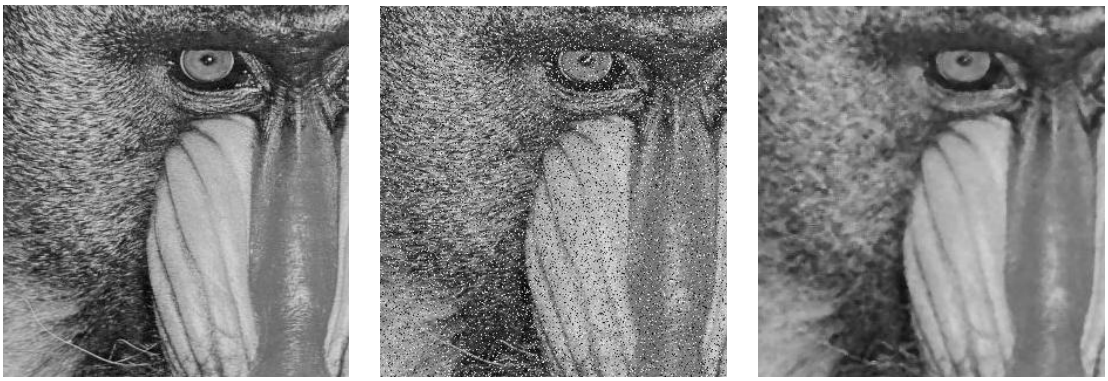
(d) BM3D [24]

(e) TSMNR [85]

(f) Proposed(Bicubic)

(g) Proposed(ISFMF)

Figure 3.14: Restoration results of *Bridge* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 10$ and $p = 25\%$

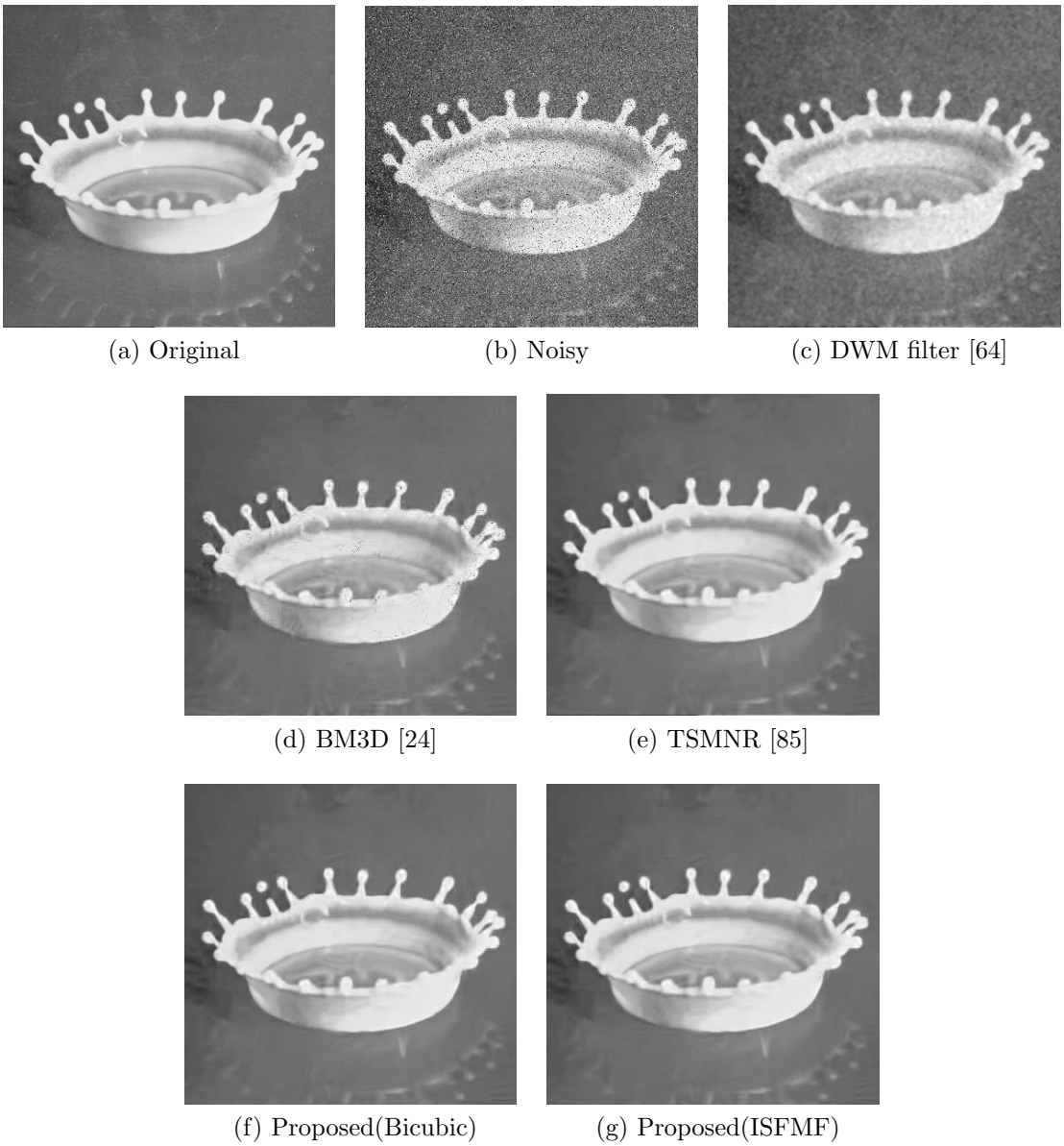


(a) Original (b) Noisy (c) DWM filter [64]

(d) BM3D [24] (e) TSMNR [85]

(f) Proposed(Bicubic) (g) Proposed(ISFMF)

Figure 3.15: Restoration results of *Mandrill* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 5$ and $p = 20\%$



(a) Original (b) Noisy (c) DWM filter [64]

(d) BM3D [24] (e) TSMNR [85]

(f) Proposed(Bicubic) (g) Proposed(ISFMF)

Figure 3.16: Restoration results of *Milkdrop* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 25$ and $p = 5\%$

(a) Original

(b) Noisy

(c) DWM filter [64]

(d) BM3D [24]

(e) TSMNR [85]

(f) Proposed(Bicubic)

(g) Proposed(ISFMF)

Figure 3.17: Restoration results of *Sailboat* corrupted by mixed Gaussian and random-valued impulse noise with $\sigma = 20$ and $p = 25\%$

Chapter 4

Nonlocal Based Image Denoising with RAISR

4.1 Introduction

Image denoising aims to recover the efficient original image $\boldsymbol{x}$ from its noisy measurement $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{y}$ is the observed noisy image, $\boldsymbol{x}$ is the latent clean image and $\boldsymbol{n}$ is defined by Additive White Gaussian Noise (AWGN) with zero mean and variance σ_n^2. Due to the influence of environment, different transmission channels and other inevitable factors, the images are contaminated by noise in many subsequent image processing tasks, such as video processing, image analysis and tracking. Therefore, image denoising has become not only an indispensible step for many vision applications but also an ideal test bed for investigating statistical image modeling techniques since a few decades.

Many researchers have been trying to remove the noise and preserve some important features of the image such as fine details, textures and singularities by applying image denoising methods based on probability theory, statistics, partial differential equations, linear and non-linear filtering, nonlocal self-similarity (NSS), sparsity, and low-rank approximations for a few years. Although the mean filter or averaging filter [88] which is a linear filter has been adopted for the removal of Gaussian noise, the images are over-smoothed with high noise. In addition, Gaussian noise can also be suppressed by exploiting non-linear filters such as median filtering [53, 88], weighted median filtering [89] and the well-known edge preserving filter called bilateral filter [3]. All these filters are not robust to high level noise because they are basically performed locally i.e., the intensity value of each pixel is replaced by a weighted average of in-

tensity values from its neighborhood pixels. Hence, they also tend to blurring effect to the edge regions of the image.

In contrast, non-local means (NLM) [12], a pioneer of nonlocal-based approach significantly enhances the image denoising performance. This is because its basic idea is to build a point-wise estimation of the image, where each pixel is obtained by weighted averaging of the pixel centered at the target patch and the pixels centered at the similar patches at different locations in the image. Block Matching and 3D filtering (BM3D) [24] has recently become a benchmark Gaussian noise removal method as the extension of NLM approach in the transform domain. The principal idea of BM3D is stacking the similar patches to obtain 3-D groups by block matching. Hard-thresholding and wiener-filtering are employed to attenuate the noise from the 3-D transformed blocks in the first and second step, respectively. In Weighted Nuclear Norm Minimization (WNNM) [30], the vectorized similar patches that are typically stacked by block matching are transformed into matrices and the noise can be suppressed based on low-rank approximations. As these two methods are mainly based on searching the patches at different locations similar to the reference patch across the image, the performance is efficiently increased.

Similarly to image denoising, single image Super-Resolution (SR) is a popular branch in image reconstruction in terms of high frequency compensation within a short period of computation time. Recently, many SR algorithms have been successfully advanced to reconstruct the image without losing the quality of the super-resolved output image. The Anchored Neighborhood Regression (ANR) [90] is an example-based super-resolution method in which the nearest neighbors are correlated with dictionary atoms instead of using euclidean distance to keep the quantitative performance in low execution speed. An improved variant of ANR [90], A+ [91] , is the most efficient dictionary-based super-resolution method which builds on the feature and anchored regressors from ANR but the full training material is employed as simple function (SF) [92].

Different from the other external example-based SR methods, the excellent quality in super-resolution performance and speed is obtained with the use of deep convo-

lutional neural network for single image super-resolution (SRCNN) [93] which learns an end-to-end mapping from low resolution (LR) images to their high resolution (HR) counterparts. Instead of learning the dictionaries for modeling the patches, the model is implicitly learned by the hidden convolutional layers in this method. Although these above SR methods are superior to the image restoration, the cost of computational complexity is very high due to the use of huge amount of dictionaries to learn mapping. Hence, the rapid and accurate image super-resolution, RAISR [94] has recently become advanced because it produces a better performance and more than one to two orders of magnitude faster than the state-of-the-art example-based image super-resolution methods. In RAISR, the LR patches extracted from the initial interpolated image are divided into hash classes. The hash parameters are evaluated based on the gradient of each patch. 864 filters including 4 classes for the pixel type, 3 classes for the strength, 24 classes for the angle and 3 classes for the coherence are needed to learn the filters. Therefore, the needed storage capacity of RAISR to install into mobile devices such as smart phones is quite large.

To overcome this problem, an improvement of RAISR is proposed in this chapter. The main idea is that the number of filters are reduced to 18 filters by upgrading the hash mechanisms which consist of minimizing the classes for the gradient angle by geometric conversion and reducing the classes for the gradient strength. Its performance and runtime are almost the same as RAISR. The improvement of RAISR nominated by IRAISR in the following sections is applied to denoising methods for enhancing the denoising performance with less memory storage requirement.

Moreover, an accurate Gaussian noise removal method followed by IRAISR as a post-processing step is proposed. There are two phases in the poposed method namely: learning phase and testing phase. In the learning phase, the noisy image is initially denoised by using nonlocal denoising method. Not only the patches extracted from the image processed by nonlocal-based noise removal method but also the pixels from Ground truth are classified into 18 hash classes with two improvements including geometric conversion and reduction of the gradient strength. The filters are learned by least-square method based on these classes. In the testing phase, Gaussian noise

is firstly suppressed by nonlocal denoising method. The patches extracted from the denoised image are divided into classes. Different from [94], only two improvements which consist of decreasing the classes for the gradient angle by geometric conversion and the reduction of the classes for the gradient strength are considered to compute the hash key parameters in the proposed method without considering the pixel type. The filtered output is obtained by utilizing the pre-learned filters from the learning phase to the patches from the denoised image in each class and aggregating them. In addition, the Census Transform (CT) [94] is also employed between the denoised image and the filtered output to protect the image structure. Eventually, an enhanced performance and pleasant visual quality of the image can be achieved by weighted averaging.

4.2 RAISR

Rapid and Accurate Image Super Resolution (RAISR) [94] is one of the most efficient learning-based single image super-resolution methods which produces high quality restoration with extremely fast evaluation time. The core idea of RAISR is to boost the image quality by employing the pre-learned filters on the image patches extracted from the initial upscaling image. The filters are learned based on the pairs of low resolution (LR) patches and high resolution (HR) pixels. RAISR needs to classify the patches into 864 classes i.e., 4 classes for pixel type which depends on the upsampling factor, 3 classes for strength, 24 classes for angle and 3 classes for coherence. Thus, 864 filters with size 11×11 are needed.

The hashing approach is considered in distinguishing the image patches into clusters without using the expensive clustering methods (e.g., K-means [95], Gaussian Mixture Model (GMM) [96], [97]) for keeping the low complexity of the linear filtering. The hash-table keys are obtained by estimating the local gradients' statistics. However, RAISR causes some artifacts such as noise amplification and halos due to the effect of filtering. In order to tackle this problem, Census Transform (CT) [98] is utilized between the initial bicubic interpolated image and the filtered output based

on the hamming distance. Some important algorithms for implementing RAISR will be specifically explained in the following subsections.

4.2.1 Calculation of Hash-table Keys

The local gradient characteristics of the matrix that can be used as the hash-table keys are evaluated by eigenanalysis [99] . The local gradient of the nearest neighborhoods of the k-th pixel which is typically $\sqrt{n} \times \sqrt{n}$ patch , i.e., all the pixels are located at $k_1, ..., k_n$ is calculated in $n \times 2$ matrix. It is composed from the horizontal gradient g_x and the vertical gradient g_y, expressed by

$$\boldsymbol{G_k} = \begin{bmatrix} g_{x_{k_1}} & g_{y_{k_1}} \\ \vdots & \vdots \\ g_{x_{k_n}} & g_{y_{k_n}} \end{bmatrix}. \tag{4.1}$$

In addition, the matrix $\boldsymbol{G}_k^T \boldsymbol{W}_k \boldsymbol{G}_k$ can be constructed by employing a diagonal weighting matrix $\boldsymbol{W}_k$, a separable normalized Gaussian kernel to incorporate a small neighborhood of gradient samples per pixel.

From an eigen-decomposition of $\boldsymbol{G}_k^T \boldsymbol{W}_k \boldsymbol{G}_k$, the gradients' strength λ_k, angle θ_k and coherence μ_k are evaluated by using the larger eigenvalue λ_1^k and the smaller eigenvalue λ_2^k, and the two eigenvectors ϕ_1^k and ϕ_2^k related to λ_1^k and λ_2^k as

$$\lambda_k = \lambda_1^k, \tag{4.2}$$

$$\theta_k = \arctan\left(\frac{\phi_{1,y}^k}{\phi_{1,x}^k}\right), \tag{4.3}$$

$$\mu_k = \frac{\sqrt{\lambda_1^k} - \sqrt{\lambda_2^k}}{\sqrt{\lambda_1^k} + \sqrt{\lambda_2^k}}. \tag{4.4}$$

The three hash-table keys are quantized for computing hash indices λ, θ and μ, denoted by

$$\lambda = \left\lceil \frac{\lambda_k}{Q_s} \right\rceil, \tag{4.5}$$

$$\theta = \left\lceil \frac{\theta_k}{Q_\theta} \right\rceil, \tag{4.6}$$

$$\mu = \left\lceil \frac{\mu_k}{Q_\mu} \right\rceil, \tag{4.7}$$

where $\lceil . \rceil$ is the ceiling function, Q_s, Q_θ and Q_μ are the quantization factors for strength, angle and coherence, respectively. In this quantization, 3 classes for the strength λ, 24 classes for the angle θ and 3 classes for the coherence μ are considered to learn the filters in RAISR [94]. Hence, 216 classes for the hash keys are needed because of the integration of these three quantized parameters.

4.2.2 Global Filter Learning

In the learning phase of RAISR, a $d \times d$ filter h has to be learned with training database images which consist of the upscaled versions of LR images $\boldsymbol{y}_i \in \mathbf{R}^{M \times N}$ and the HR images $\boldsymbol{x}_i \in \mathbf{R}^{M \times N}$, $i = 1, ..., L$. L is the number of images in the training set. The filter is typically computed by solving a least-squares minimization problem

$$\boldsymbol{h} = \min_{\boldsymbol{h}} \sum_{i=1}^{L} \|\boldsymbol{A}_i \boldsymbol{h} - \boldsymbol{b}_i\|_2^2 \tag{4.8}$$

where $\boldsymbol{h}$ is the filter in vector notation with size $d^2 \times 1$. $\boldsymbol{A}_i$ is the matrix with size $MN \times d^2$ composed of the patches with size $d \times d$ extracted from the image $\boldsymbol{y}_i$. $\boldsymbol{b}_i$ is the vector with size $MN \times 1$ composed of the pixels extracted from the image $\boldsymbol{x}_i$, corresponding to the center coordinates of $\boldsymbol{y}_i$ patches.

In order to control the complexity of estimating the filters for the quite enormous matrix $\boldsymbol{A}$, the minimization of the least-squares problem can be solved by RAISR based on the extension of Eq. (4.8) as

$$\boldsymbol{h} = \min_{\boldsymbol{h}} \|\boldsymbol{Q}\boldsymbol{h} - \boldsymbol{V}\|_2^2, \tag{4.9}$$

where $\boldsymbol{Q} = \boldsymbol{A}^T \boldsymbol{A}$ and $\boldsymbol{V} = \boldsymbol{A}^T \boldsymbol{b}$. The memory requirements and the computational complexity can be reduced depending upon the observation of the matrix-matrix and

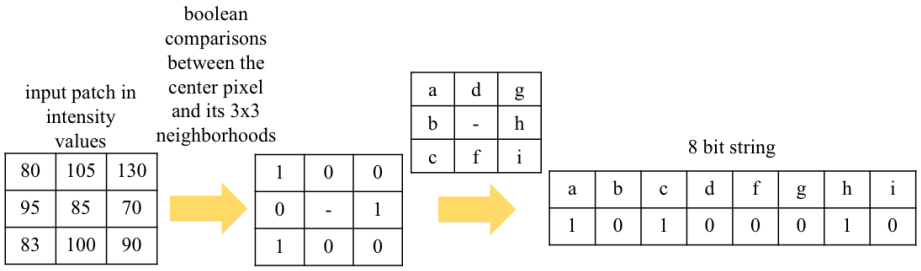


Figure 4.1: Census transform(CT)

matrix-vector multiplications as described in

$$\boldsymbol{Q} = \boldsymbol{A}^T\boldsymbol{A} = \sum_{i=1}^{L} \boldsymbol{A}_i^T\boldsymbol{A}_i \tag{4.10}$$

and

$$\boldsymbol{V} = \boldsymbol{A}^T\boldsymbol{b} = \sum_{i=1}^{L} \boldsymbol{A}_i^T\boldsymbol{b}_i, \tag{4.11}$$

where $\boldsymbol{Q}$ is a small $d^2 \times d^2$ matrix and $\boldsymbol{V}$ is a $d^2 \times 1$ matrix.

4.2.3 Detection of Structure Deformations: CT Transform

Structure deformations may occur when the pre-learned filters applied to the initial interpolated image due to the sharpening property. To preserve the important image structures, the Census Transform (CT) [98] is exploited between the upscaled image and the filtered output. As illustrated in Fig. 4.1, an 8 bit string that measures the local structure is constructed by boolean comparisons between the center pixel and its nearby pixels in 3×3 patch. Then, the hamming distance is evaluated to count the number of bits that were changed for each pixel. The larger weights converted from the number of changed bits can be achieved according to the increase in hamming distance because the change in structure depends on the distance. The output image can be estimated by weighted averaging of the interpolated image and the filtered image. Moreover, the more contrast-enhanced output is sufficiently achieved because of the consideration in a wide range of frequencies using this approach.

4.3 Proposed Method

Even though most of the nonlocal-based image denoising methods accurately suppress the Gaussian noise, some fine image details are degraded due to the losses of high frequency components in the image. Therefore, an accurate Gaussian noise removal method is proposed by applying IRAISR as a rapid post-processing step based on the extension of RAISR [94] to preserve the distorted image details in the denoised image because it can compensate the high frequency information. There are mainly two processes in the proposed method. The first process is to remove the noise from the noisy image for obtaining the denoised image. The second process is to enhance the performance of the denoised image by applying the improved RAISR to the noise-free image processed by nonlocal-based noise removal methods. The learning phase and testing phase of the proposed method are shown in Fig. 4.2 and Fig. 4.3, respectively. How to remove the noise from the noisy images and the improvement of RAISR in denoising will be explained in the following subsections.

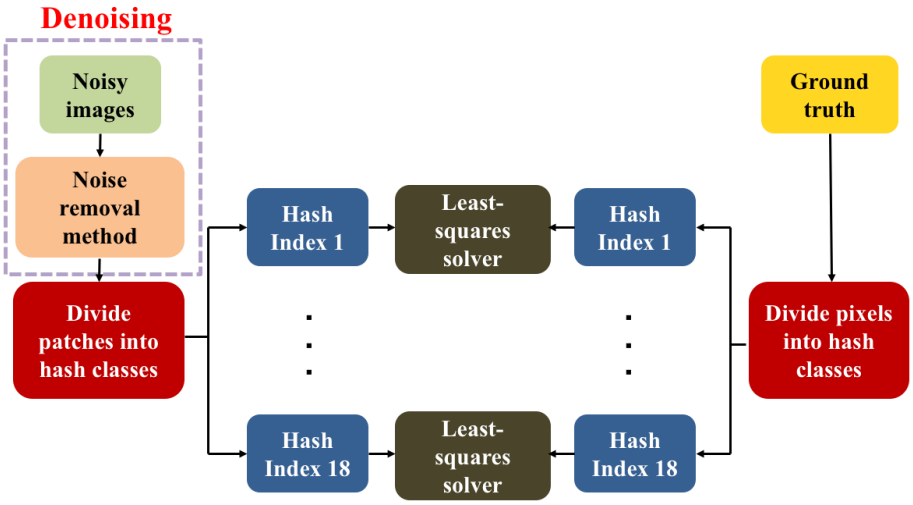


Figure 4.2: Learning phase of the proposed method

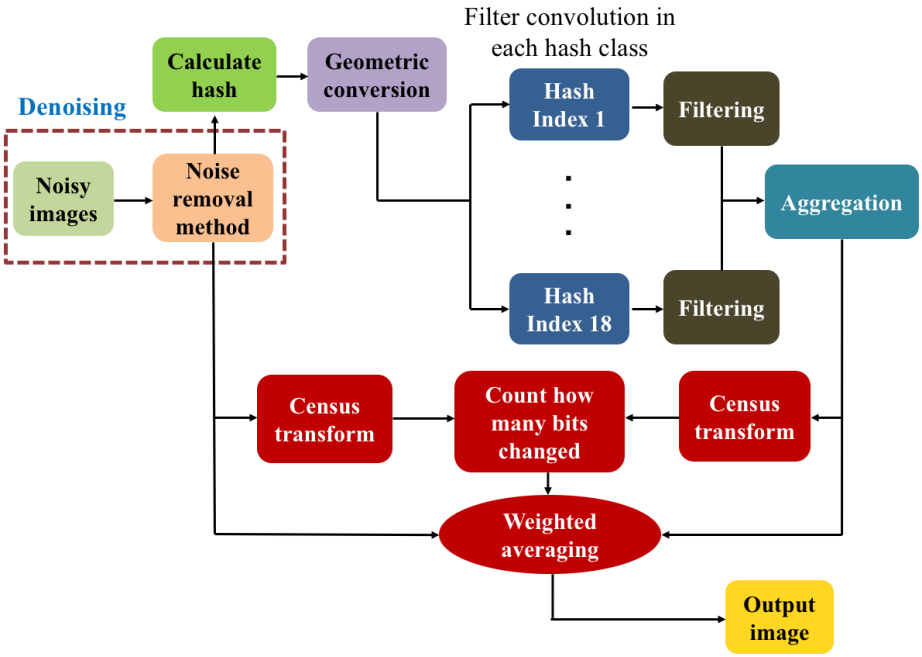


Figure 4.3: Testing phase of the proposed method

4.3.1 Noise Removal

The extensively used Gaussian noise removal methods based on nonlocal self-similarity models such as BM3D [24] and WNNM [30] are firstly utilized to suppress the noise from the noisy image in the learning phase of the proposed method. The patches extracted from the denoised image are classified into 18 hash classes including 6 classes for the angle θ and 3 classes for the coherence μ without considering the classes for the strength. Analogously, the same hash classes are divided from the pixels of Ground truth. The 18 filters can be learned by solving the least squares method between the denoised image and the reference image in 18 hash classes.

In the testing phase, the denoised image can be obtained from the noisy images that are not included in the training sets by using nonlocal-based Gaussian noise removal methods, BM3D [24] and WNNM [30]. The patches extracted from the denoised image are separated into hash classes. The 18 hash classes which is composed of 6 classes for the angle and 3 classes for the coherence are divided from the patches of denoised image with geometric conversion. Then, the pre-learned filters (18 filters for 18 classes) generated from the learning phase are applied on the patches and the

filtered image is produced by aggregating the outputs of each patch.

Similarly to the primary RAISR [94], Census Transform (CT) is used between the denoised image and the filtered output to keep the structure deformations within a wide range of frequencies. Then, Hamming distance is evaluated to know how many bits are changed between the center pixel and its nearest neighborhood pixels of each patch based on CT. The larger the hamming distance, the larger the change in structure and the bigger the weights. Finally, an advanced quality of output image can be achieved by weighted averaging of the denoised image and the filtered output because this CT approach leads to contrast enhancement of the denoised image.

4.3.2 Improvement of RAISR in Denoising

The nonlocal-based image denosing methods assigned to the proposed method (BM3D [24] and WNNM [30]) can sufficiently eliminate the Gaussian noise. However, some important image details from the denoised image are deteriorated due to the damage of high frequency information. In order to overcome this problem, an improved RAISR is exploited in the proposed method as a post-processing approach to the denoised image because it is not only a rapid and accurate image super-resolution method but also it can restore the degraded high frequency components. A modified version of RAISR [94] used for the proposed method is designed as shown in Fig. 4.4. The size of the patches extracted from the denoised image is assigned to 11×11 and the patch size of hash table is defined as 9×9. Unlike the initial RAISR [94], the pixel type is not taken into account in the proposed method because the denoised image is assumed to be the interpolated image. The pre-learned filter generated from the learning phase is also the same patch size as the denoised patches. The output pixel is implemented by convolving the patches from the denoised image with the filters based on the hash-table indices.

In RAISR [94], the LR patches extracted from the initial bicubic upscaled image are classified into 864 hash classes which consist of 4 classes for the pixel type based on the upscaling factor, 3 classes for the strength λ, 24 classes for the angle θ and 3 classes for the coherence μ. The memory storage for the filter coefficients in this

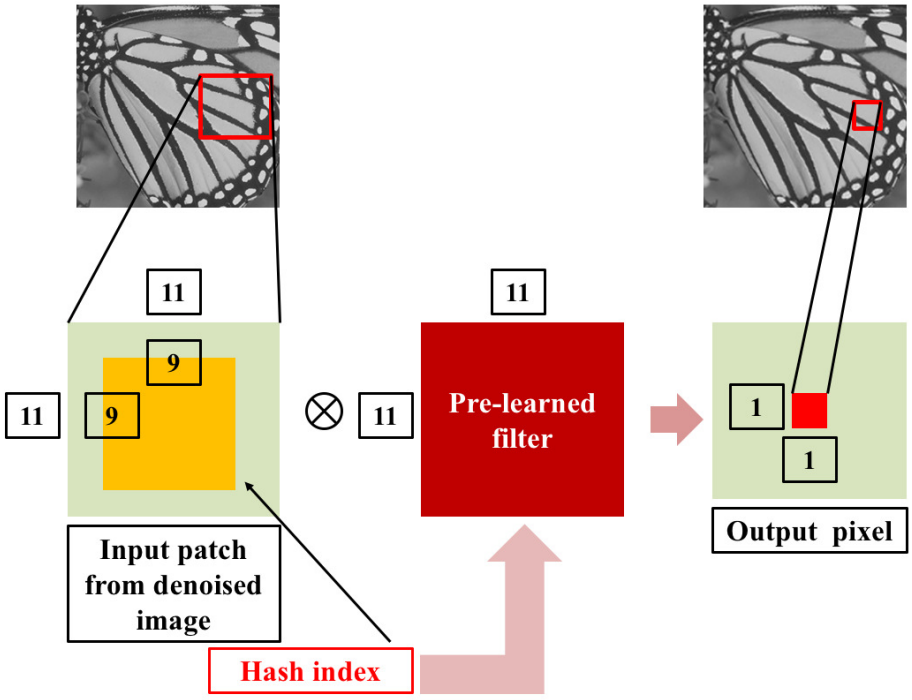


Figure 4.4: Improved RAISR design for the proposed method based on the pairs of the patches from the denoised image and the pixels from the clean image

approach is highly increased due to the use of 864 filters with size 11×11. In order to solve this problem, the number of filters are reduced to 18 filters by addressing IRAISR with simple geometric and reduction of the classes for λ_k without interfering the performance and computational complexity of the original RAISR [94]. For this reason, IRAISR is utilized in the application of denoising to achieve an accurate denoised image with low memory requirements. Although three improvements of RAISR including the reduction of the classes for the pixel type by rotation based on scaling factor, geometric conversion for the gradient angle and the reduction of the classes for the gradient strength are contributed in the super-resolution (SR) approach to obtain the same performance and runtime comparable to RAISR with little computational complexity, only two improvements are considered in image denoising because the pixel type is not contributed in this implementation.

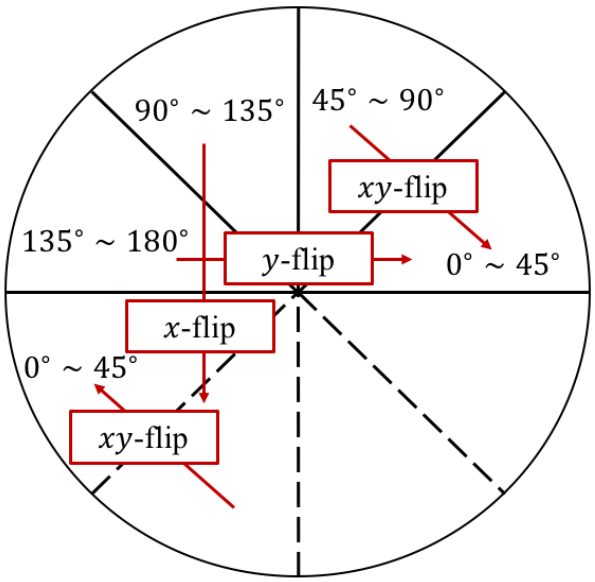


Figure 4.5: Geometric conversion based on each gradient type

4.3.3 Geometric Conversion

The gradient angle θ of the patch with size 11×11 can be changed by geometric conversion to reduce the number of classes for the gradient angle. As illustrated in Fig. 4.5, the patches whose θ_k range from 45° to 90° are applied xy-flip to, and the patches whose θ_k range from 90° to 135° are applied x-flip and xy-flip to, and the patches whose θ_k range from 135° to 180° are applied y-flip to the range from 0° to 45°, respectively. Hence, only 6 angle classes which belong to the range from 0° to 45° are needed in the proposed method instead of 24 classes for the gradient angle in RAISR [94].

4.3.4 Reduction of the Classes for the Gradient Strength

Another improvement of RAISR considered in the proposed method is the reduction of the classes for the gradient strength. The three classes for λ are classified as λ_1, λ_2 and λ_3 in RAISR [94]. The parameter λ_k is ranged from 0 to 0.34 for the class λ_1, from 0.34 to 0.67 for the class λ_2 and from 0.67 to 1 for the class λ_3, respectively to learn the filters. Table. 4.1 shows the performance comparisons of applying the

Table 4.1: Average PSNR comparison for upscaling by a factor of 2 over Set5.

Image name	Only λ_1	Only λ_2	Only λ_3	RAISR
Baby	37.127	38.387	38.450	38.452
Bird	37.101	39.702	40.275	40.264
Butterfly	27.249	29.911	30.498	30.526
Head	34.905	35.649	35.704	35.695
Woman	32.365	34.453	34.851	34.843
Average	33.749	35.620	35.956	35.956

filters learned with only λ_1, only λ_2, and only λ_3 on Set5 [100] by an upsampling factor of 2. According to the results, the performance of applying the filters learned with only λ_3 is quite similar to RAISR. Actually, the strength of gradient λ implies the presence of high-frequency components in the input patch. As the filter learned with high-frequency patches can generate the low-frequency HR patches, the hash classes for the strength are not needed in the testing phase.

4.4 Experimental Results

The comparison of the proposed method which is actually an enhanced denoising algorithm using IRAISR with nonlocal-based state-of-the-art Gaussian noise removal methods such as BM3D [24] and WNNM [30] is presented in this section. The peak signal to noise ratio (PSNR) is used as a quantitative metric for performance evaluation.

4.4.1 Parameters Setting

The parameter settings utilized in the proposed method are as follows: the patch size is set as 9×9; the patch step size is 3; the neighborhood window size is set as 39×39 in both steps of BM3D filtering. The maximum number of similar patches are fixed to 16 in hard-thresholding and 32 in wiener-filtering, respectively. The input noisy images are padded symmetrically depending on the patch size to keep the image borders in this implementation before processing.

In WNNM, the iterative regularization parameter δ and the parameter c are defined as 0.1 and 2.8 for all noise levels. The patch size and the iteration number K are selected based on noise level. The patch size is described as 6×6, 7×7, 8×8 and 9×9 for $\sigma_n \leq 20$, $20 < \sigma_n \leq 40$, $40 < \sigma_n \leq 60$ and $60 < \sigma_n$, respectively. K is set to 8, 12, 14 and 14, respectively based on these noise levels. When the noise level is higher, the chosen patch size should be bigger and the consuming time is longer. 191 images including General 100 and 91 images from Yang *et al.* are used as training image sets to learn the filters in the improvement of RAISR for image denoising. The patch size of the learned filter is 11×11. The hash index is considered on its neighborhood of size 9×9 for the computation of hash key parameters in both learning and testing phase of the proposed method. The source codes of BM3D [24] and WNNM [30] provided on their relevant websites are used for the experiments.

4.4.2 Quantitative and Visual Evaluation

The comparison of PSNR values of nonlocal-based image denoising methods including BM3D [24] and WNNM [30] with the proposed method which is the application of IRAISR as a post-processing stage to the denoising methods is presented in Table. 4.2. The experiments are conducted over 11 extensively used test images corrupted by Gaussian noise with $\sigma = 10$, $\sigma = 20$, $\sigma = 30$, $\sigma = 40$, $\sigma = 50$, and $\sigma = 60$, respectively. The highest PSNR values are marked in bold-faced. As can be observed, WNNM post-processed by IRAISR outperforms efficiently the other competing methods in the *Airplane*, *Butterfly*, *Cameraman* and *Peppers* images which are rich in edge regions for all noise levels because the image details in the edge regions of the image can be well restored in the proposed method.

The integration of BM3D and IRAISR gives the highest PSNR value on average at noise levels $\sigma = 20$ and $\sigma = 30$, respectively. When the noise level is slightly increased, the combination of WNNM and IRAISR is the best in almost all test images excluding *Bird* and *Montage* images. As WNNM can remove the Gaussian noise level (especially $\sigma \geq 40$) more efficiently than BM3D, the performance evaluation of WNNM is higher than BM3D. WNNM is the signal processing-based state-of-the-art noise reduction

method. Hence, the combination of WNNM and IRAISR can be sufficiently applied for the high noise levels while preserving the image details. However, the superiority of quantitative performance on average for all noise levels can be achieved by using the proposed method.

Table 4.2: The quantitative comparison on average PSNR (dB) over 11 widely used test images

Images	Airplane	Barbara	Bird	Boat	Butterfly	C.man	Couple	Lena	Man	Montage	Peppers	Average
Noise Level	$\sigma = 10$											
BM3D [24]	33.37	**35.04**	**37.39**	33.69	33.12	33.73	33.77	35.85	33.63	36.89	34.27	34.62
WNNM [30]	33.39	**35.04**	37.14	33.49	34.31	33.80	33.55	35.61	33.51	**37.59**	34.52	34.72
BM3D+IRAISR	33.44	35.01	**37.39**	**33.72**	33.24	33.73	**33.83**	**35.93**	**33.73**	36.89	34.32	34.66
WNNM+IRAISR	**33.63**	34.93	37.24	33.63	**34.54**	**33.96**	33.73	35.72	33.71	37.11	**34.67**	**34.81**
Noise Level	$\sigma = 20$											
BM3D [24]	29.45	**31.81**	**33.68**	30.59	29.60	29.99	30.53	32.96	30.27	32.99	30.91	31.16
WNNM [30]	29.57	31.42	32.89	30.23	30.49	29.95	30.11	32.53	29.90	**33.68**	30.90	31.06
BM3D+IRAISR	29.58	31.74	**33.68**	**30.67**	29.77	30.03	**30.62**	**33.09**	**30.39**	32.89	30.98	**31.22**
WNNM+IRAISR	**29.88**	31.42	33.08	30.44	**30.79**	**30.13**	30.35	32.69	30.14	32.95	**31.09**	31.18
Noise Level	$\sigma = 30$											
BM3D [24]	27.43	**29.86**	**31.43**	28.84	27.72	28.19	28.69	31.18	28.60	30.71	28.98	29.24
WNNM [30]	27.49	29.48	30.67	28.49	28.39	28.26	28.31	30.86	28.29	**31.05**	28.91	29.11
BM3D+IRAISR	27.53	29.79	31.42	**28.91**	27.89	28.25	**28.78**	**31.31**	**28.71**	30.62	28.99	**29.29**
WNNM+IRAISR	**27.79**	29.54	30.84	28.69	**28.76**	**28.47**	28.54	31.00	28.50	30.68	**29.08**	29.26
Noise Level	$\sigma = 40$											
BM3D [24]	25.96	27.59	29.64	27.38	25.66	26.53	27.15	29.55	27.28	28.52	27.38	27.51
WNNM [30]	26.04	27.88	28.98	27.18	26.96	26.99	26.88	29.51	27.11	**29.16**	27.47	27.65
BM3D+IRAISR	26.04	27.44	**29.68**	**27.47**	25.99	26.78	**27.23**	**29.75**	**27.38**	28.53	27.39	27.61
WNNM+IRAISR	**26.33**	**28.05**	29.15	27.41	**27.37**	**27.24**	27.13	29.66	27.33	28.97	**27.65**	**27.84**
Noise Level	$\sigma = 50$											
BM3D [24]	24.55	26.56	**28.37**	25.98	24.62	24.85	25.74	28.51	26.19	25.97	26.15	26.14
WNNM [30]	25.05	26.97	27.87	26.27	25.77	25.89	25.97	28.70	26.32	**27.52**	26.44	26.62
BM3D+IRAISR	24.73	26.17	28.36	26.16	24.91	25.27	25.92	28.65	26.32	26.04	26.16	26.24
WNNM+IRAISR	**25.29**	**27.11**	28.02	**26.47**	**26.21**	**26.16**	**26.19**	**28.85**	**26.49**	27.36	**26.62**	**26.79**
Noise Level	$\sigma = 60$											
BM3D [24]	23.98	25.66	**27.50**	25.35	23.83	24.21	25.03	27.79	25.63	25.10	25.44	25.41
WNNM [30]	24.25	25.99	26.84	25.49	24.83	25.03	25.11	27.90	25.61	**26.42**	25.59	25.73
BM3D+IRAISR	24.09	25.03	27.48	25.50	24.09	24.60	25.17	27.92	25.72	25.33	25.39	25.48
WNNM+IRAISR	**24.49**	**26.17**	26.99	**25.69**	**25.31**	**25.29**	**25.33**	**28.06**	**25.79**	26.38	**25.78**	**25.94**

The execution time comparisons of the benchmark Gaussian noise removal methods (BM3D [24] and WNNM [30]) with the application of IRAISR to these denoising methods for 256×256-sized and 512×512-sized images corrupted by additive white Gaussian noise with $\sigma = 30$ are reported in Table. 4.3 and Table. 4.4, respectively. According to the experimental results, the proposed method takes approximately 0.18 sec and 0.66 sec on average longer than the denoising methods for 256×256-sized and 512×512-sized images. Therefore, the performance of image denoising can be improved by employing IRAISR to the noise removal methods within a rapid process-

Table 4.3: Execution time comparison for 256×256-sized images

Images(256×256)	Execution time in sec ($\sigma = 30$)			
	BM3D [24]	BM3D+IRAISR	WNNM [30]	WNNM+IRAISR
Airplane	23.29	23.47	139.44	139.68
Butterfly	22.14	22.32	156.14	156.31
C.man	19.92	20.09	146.01	146.17
Montage	17.93	18.11	146.26	146.44
Peppers	21.01	21.19	143.59	143.75
Average	20.86	21.04	146.29	146.47

Table 4.4: Execution time comparison for 512×512-sized images

Images(512×512)	Execution time in sec ($\sigma = 30$)			
	BM3D [24]	BM3D+IRAISR	WNNM [30]	WNNM+IRAISR
Barbara	93.95	94.53	577.21	577.84
Boat	91.72	92.29	617.99	618.69
Couple	100.33	100.98	611.77	612.42
Lena	91.76	92.37	606.81	607.48
Man	90.53	91.09	567.41	568.09
Average	93.66	94.25	596.24	596.90

ing time. The merit of the improved RAISR is the reduction of learned filters with the equivalent performance and run-time as RAISR [94] which utilizes 864 filters with size 11×11. The reduction of the number of filters simplifies the implementation of RAISR as well as provides the denoising performance effectively.

The evaluations of visual quality of the proposed method are compared with the denoising methods including BM3D [24] and WNNM [30] for 256×256-sized *Butterfly* image corrupted by additive white Gaussian noise with $\sigma = 30$ and *Peppers* image degraded by $\sigma = 50$ as illustrated in Fig. 4.6 and Fig. 4.7, respectively. It can be seen that the edge regions of *Butterfly* are sharper than the competing methods due to the effect of Census Transform (CT) [94] although the flat regions are a little over-smoothed in the proposed method. In *Peppers* image, the edges are well preserved in the improvement of RAISR applied to the denoising methods except for the generation of a little blur in the smooth regions. The improvement of image denoising using IRAISR for 256×256-sized *Airplane* image degraded by Gaussian noise $\sigma = 40$ and *Cameraman* image corrupted by $\sigma = 60$ are comparatively visualized with the conventional Gaussian noise removal methods as illustrated in Fig. 4.8 and Fig. 4.9, respectively. The abruptly changed image pixel intensities of both images are

significantly reconstructed in the proposed method in accordance with the simulated results.

Figure. 4.10 and Fig. 4.11 compare the visual assessments of 512×512-sized *Couple* image corrupted by Gaussian noise with standard deviation 30 and *Boat* image corrupted by standard deviation 50. To be clearly seen, the emphasized regions of both images are demonstrated in orange box. It is obvious that the strong edge sharpness can be effectively produced in spite of less maintenance of some image details in the highlighted regions of *Couple* image. Moreover, some image features of *Boat* image (*e.g.,* text written on the front part of the boat) are failed to restore but a tiny rope can be clearly visualized in the proposed method.

The subjective comparisons for 512×512-sized *Barbara* image deteriorated by Gaussian noise with $\sigma = 40$ and *Lena* image affected by GN with $\sigma = 60$ are respectively shown in Fig. 4.12 and Fig. 4.13. As can be observed, the highlighted texture patterns on the table sheet in *Barbara* image can be well reconstructed and some salient image details in *Lena* can be significantly preserved in nonlocal-based image denoising using IRAISR. In addition, the visualization of *Montage* image contaminated by Gaussian noise with $\sigma = 20$ and *Man* image affected by GN with $\sigma = 20$ using IRAISR are compared with existing image denoising methods as illustrated in Fig. 4.14 and Fig. 4.15, respectively. It can be obvious that the proposed method gives strong sharp edge restoration in *Montage* and excellent preservation of hair textures in the enlarged fragment of *Man* image while efficiently removing Gaussian noise.

4.4.3 Experiments on Various Datasets

Besides conducting experiments on 11 widely used natural images as a test set for image denoising, the proposed accurate Gaussian noise removal method is applied to other various image datasets including Kodak images, Berkeley segmentation dataset (BSD68) and Set12 images. These image datasets are also not included in the training set. The average quantitative and runtime comparisons of the proposed method with BM3D and WNNM are reported in Table. 4.5 for those image datasets corrupted

Table 4.5: Comparison of PSNR values and runtime on average over various datasets

Dataset	Noise Level	BM3D		BM3D+IRAISR		WNNM		WNNM+IRAISR	
		PSNR	runtime	PSNR	runtime	PSNR	runtime	PSNR	runtime
Kodak	10	34.16	149.132	**34.25**	150.067	34.03	418.649	34.24	419.552
	30	28.98	149.326	**29.09**	150.261	28.77	872.275	28.99	873.126
	50	26.39	93.647	26.53	94.49	26.82	694.370	**27.00**	695.251
BSD68	10	33.04	56.685	33.12	57.085	32.91	180.075	**33.13**	180.454
	30	27.49	56.690	**27.60**	57.089	27.35	338.575	27.58	338.944
	50	24.84	36.498	25.02	36.883	25.34	254.98	**25.53**	255.360
Set12	10	34.13	51.057	34.19	51.442	34.16	164.556	**34.30**	164.901
	30	28.88	50.871	28.98	51.288	28.81	323.650	**29.01**	324.007
	50	25.96	31.841	26.13	32.193	26.47	235.121	**26.69**	235.470

by different noise levels $\sigma = 10$, $\sigma = 30$ and $\sigma = 50$, respectively. The best PSNR values are highlighted in bold faced. It can be observed that the combination of BM3D and IRAISR outperforms the other competing methods at noise levels $\sigma = 10$ and $\sigma = 30$, and WNNM post-processed by improvement of RAISR is the highest value among them at noise level $\sigma = 50$ in Kodak images. In BSD68 dataset, the application of IRAISR to BM3D produces the highest PSNR value at noise level $\sigma = 30$, and the integration of WNNM and IRAISR generates the best PSNR values at noise levels $\sigma = 10$ and $\sigma = 50$, respectively. Similarly, the greatest PSNR values can be achieved in the combination of WNNM and IRAISR for all noise levels in Set12 images. Therefore, the proposed noise suppression method for Gaussian noise with the aid of IRAISR as a post-processing step is very effective for any dataset to obtain high quality denoising performance. The execution time for denoising and the application of IRAISR depend on the image size. However, it takes a rapid processing time for IRAISR to enhance the denoising performance.

Figure. 4.16 and Fig. 4.17 compare the visual assessments of one image from Kodak dataset corrupted by Gaussian noise with standard deviation 30 and one test image taken from BSD68 contaminated by standard deviation 50. To be clearly seen, the enlarged fragments of both images are demonstrated in green box. It is obvious that not only Gaussian noise can be successfully removed but also the strong edge sharpness (e.g., numerical values written on the sail) can be effectively produced in

the highlighted regions of the image by using the proposed method. Moreover, some image details in the cropped regions of the image from BSD68 can be well recovered in the proposed method according to the performed visual evaluation.

In addition, the restoration results of one image taken from Kodak dataset corrupted by Additive White Gaussian Noise with $\sigma = 50$ and one image chosen from BSD68 dataset affected by GN with $\sigma = 30$ are shown in Fig. 4.18 and Fig. 4.19, respectively. The proposed method can successfully remove Gaussian noise and efficiently preserve the image details such as stripes in the highlighted regions of both images.

Furthermore, the denoised results of one image from Set12 dataset contaminated by $\sigma = 30$ are illustrated in Fig. 4.20 to compare the visual evaluation of the proposed method with the conventional denoising methods including BM3D and WNNM. Gaussian noise can be eliminated and some image structures of *Starfish* image can be well restored in the proposed denoising method with super-resolution technique.

4.4.4 Comparison of Performance Evaluation on Different Training Sets

The performance evaluation and runtime of the proposed denoising method which is the application of IRAISR to BM3D are compared by employing the pre-learned filters from two different training sets: 191 images including General 100 images and 91 images from Yang *et al.*, and General 100 images as described in Table. 4.6. The experiments are evaluated on 11 widely used natural images corrupted by Gaussian noise with $\sigma = 10$, $\sigma = 30$ and $\sigma = 50$, respectively. It takes about 8 hours for 191 training set and 3 hours for 100 training set to learn the filters in the training phase. The best PSNR values are described in bold faced and the fastest runtimes are denoted by bold faced blue color. As can be seen from the table, the average PSNR values are almost the same at noise levels $\sigma = 10$ and $\sigma = 30$ for both training sets. When the noise level is steadily increased to 50, the average PSNR value of the combination of BM3D and IRAISR is higher in 191 training images. Moreover, it

Table 4.6: Performance evaluation on different training datasets

Training Dataset	Noise Level	BM3D+IRAISR	
		PSNR	runtime
191 images	10	**34.66**	**54.758**
	30	**29.29**	**51.885**
	50	**26.24**	**33.928**
100 images	10	**34.66**	58.265
	30	**29.29**	53.326
	50	26.23	34.004

takes faster execution time in the testing phase by applying the filters learned from the training 191 images than 100 training dataset for all noise levels.

4.4.5 Effect of CT in Image Denoising

Census transform (CT) [94] is utilized to restore the local image structures between the denoised image and the filtered output. The experiments are conducted on 11 widely used natural images degraded by Gaussian noise $\sigma = 10$, $\sigma = 30$ and $\sigma = 50$, respectively to know the effect of CT in image denoising. Table. 4.7 presents the comparison of average PSNR values among BM3D, BM3D+IRAISR without CT and BM3D+IRASIR with CT. According to the simulation result, the proposed denoising method with CT highlighted in bold face is the best among them for all noise levels.

Table 4.7: Performance comparison of image denoising based on CT over 11 widely used images

Method	Noise level		
	$\sigma = 10$	$\sigma = 30$	$\sigma = 50$
BM3D	34.62	29.24	26.14
BM3D+IRAISR without CT	34.65	29.27	26.18
BM3D+IRAISR with CT	**34.66**	**29.29**	**26.24**

4.4.6 Performance Comparison between RAISR and IRAISR in Image Denoising

RAISR and IRAISR can be applied to noise reduction methods as a post-processing step to enhance the performance of denoising. 216 filters which consist of 3 classes for the gradient strength, 24 classes for the gradient angle and 3 classes for the coherence in RAISR, and 18 filters including 3 classes for the coherence and 6 classes for the gradient angle by geometric conversion in IRAISR are needed excluding the pixel type in the image denoising application. The reduction of the classes in IRAISR simplifies the implementation of RAISR without sacrificing the performance and runtime of RAISR. The experiments are conducted on 11 widely used images contaminated by Gaussian noise with $\sigma = 10$, $\sigma = 30$ and $\sigma = 50$, respectively to compare the average PSNR value and runtime between RAISR and IRAISR applied to the denoising method (BM3D in this experiment) as presented in Table. 4.8. As can be observed, the PSNR values of IRAISR and RAISR are almost the same for all noise levels although the computation time for RAISR is a little faster than that of IRAISR. Moreover, the computation time for learning the filters in the training phase of IRAISR can be successfully reduced due to the minimization of the classes. Therefore, the proposed method can improve the performance of nonlocal image denoising methods with little computational complexity.

Table 4.8: Impact of RAISR and IRAISR in image denoising over 11 widely used images

Method	Noise level						No. of filters
	$\sigma = 10$		$\sigma = 30$		$\sigma = 50$		
	PSNR	runtime	PSNR	runtime	PSNR	runtime	
BM3D+RAISR	**34.66**	53.585	**29.29**	52.170	**26.24**	33.628	216
BM3D+IRAISR	**34.66**	54.758	**29.29**	51.885	**26.24**	33.928	18

4.5 Summary

In this Chapter, an accurate Gaussian noise removal followed by an Improvement of RAISR (IRAISR) has been presented to successfully remove the noise without regressing some image features such as edges and textures. BM3D and WNNM, the state-of-the-art nonlocal-based noise suppression methods for Gaussian, are used for image denoising. In order to preserve the distored image details from the denoised image, IRAISR is utilized as a post-processing step because it is a rapid and accurate learning-based super-resolution method as well as it can restore the high frequency information. The main idea behind IRAISR is the minimization of the number of filters. The number of filters can be reduced from 216 to 18 by two improvements including the reduction of the classes for the gradient angle by geometric conversion and the reduction of the classes for the gradient strength without considering the pixel type in the image denoising. IRAISR can simplify the implementation of RAISR and provides the denoising performance effectively with less computational complexity.

(a) Ground truth (b) Noisy

(c) BM3D [24] (d) WNNM [30]

(e) BM3D+IRAISR (f) WNNM+IRAISR

Figure 4.6: Visual comparison of *Butterfly* image corrupted by $\sigma = 30$

(a) Ground truth

(b) Noisy

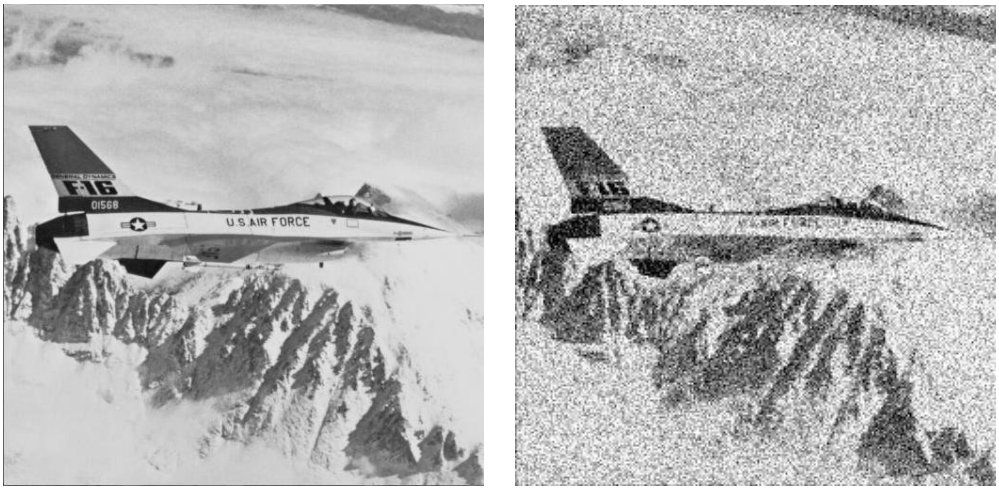
(c) BM3D [24]

(d) WNNM [30]

(e) BM3D+IRAISR

(f) WNNM+IRAISR

Figure 4.7: Visual comparison of *Peppers* image corrupted by $\sigma = 50$



(a) Ground truth (b) Noisy

(c) BM3D [24] (d) WNNM [30]

(e) BM3D+IRAISR (f) WNNM+IRAISR

Figure 4.8: Visual comparison of *Airplane* image corrupted by $\sigma = 40$

(a) Ground truth

(b) Noisy

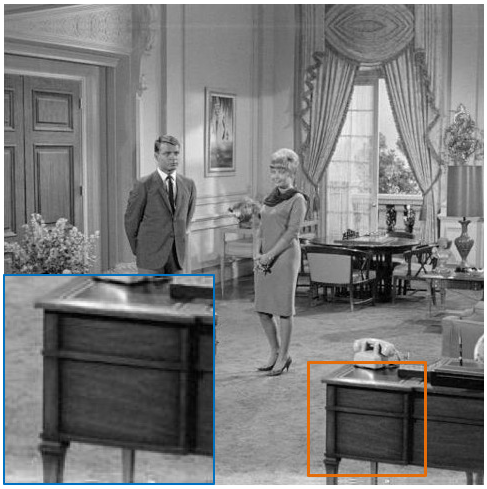
(c) BM3D [24]

(d) WNNM [30]

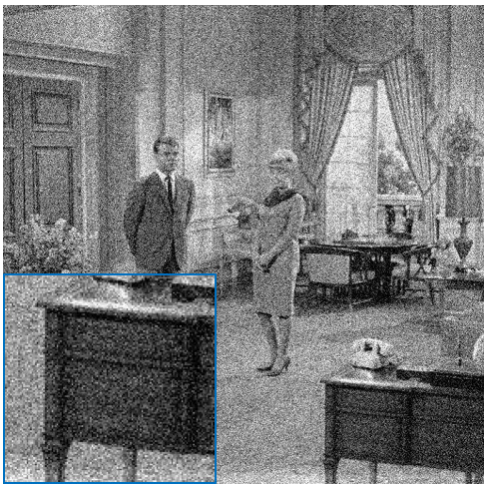
(e) BM3D+IRAISR

(f) WNNM+IRAISR

Figure 4.9: Visual comparison of *Cameraman* image corrupted by $\sigma = 60$



(a) Ground truth

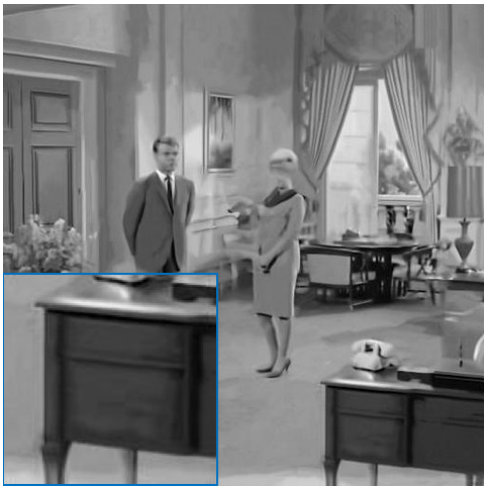


(b) Noisy

(c) BM3D [24]

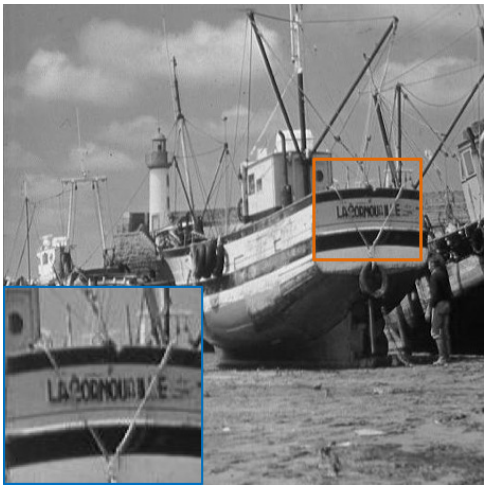
(d) WNNM [30]

(e) BM3D+IRAISR

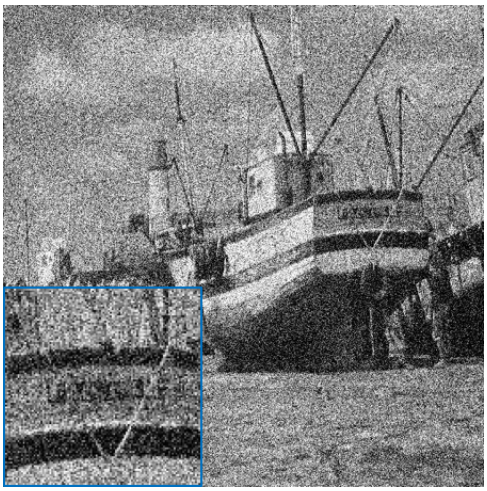


(f) WNNM+IRAISR

Figure 4.10: Visual comparison of *Couple* image corrupted by $\sigma = 30$



(a) Ground truth



(b) Noisy

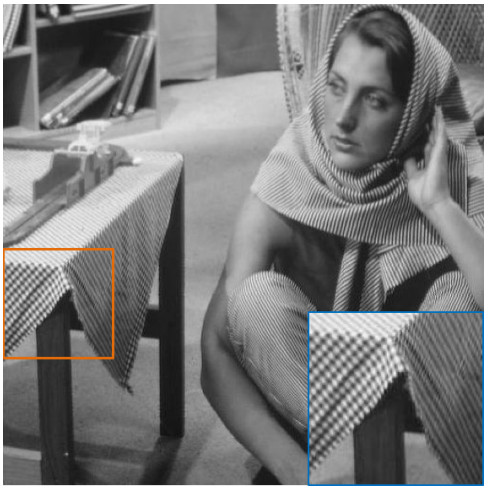
(c) BM3D [24]

(d) WNNM [30]

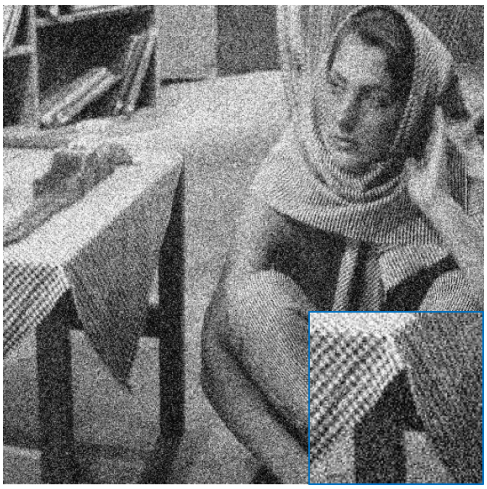
(e) BM3D+IRAISR

(f) WNNM+IRAISR

Figure 4.11: Visual comparison of *Boat* image corrupted by $\sigma = 50$



(a) Ground truth



(b) Noisy

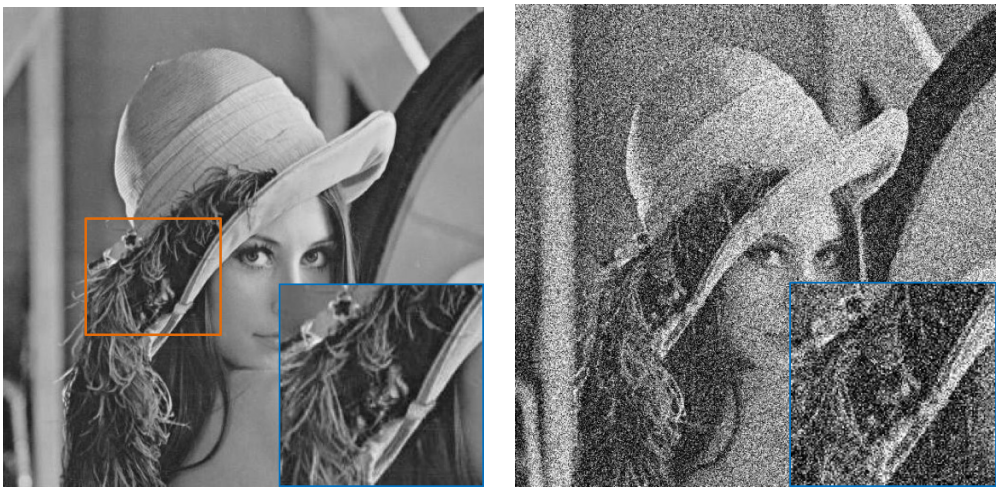
(c) BM3D [24]

(d) WNNM [30]

(e) BM3D+IRAISR

(f) WNNM+IRAISR

Figure 4.12: Visual comparison of *Barbara* image corrupted by $\sigma = 40$



(a) Ground truth (b) Noisy

(c) BM3D [24] (d) WNNM [30]

(e) BM3D+IRAISR (f) WNNM+IRAISR

Figure 4.13: Visual comparison of *Lena* image corrupted by $\sigma = 60$

(a) Ground truth

(b) Noisy

(c) BM3D [24]

(d) WNNM [30]

(e) BM3D+IRAISR

(f) WNNM+IRAISR

Figure 4.14: Visual comparison of *Montage* image corrupted by $\sigma = 20$

(a) Ground truth

(b) Noisy

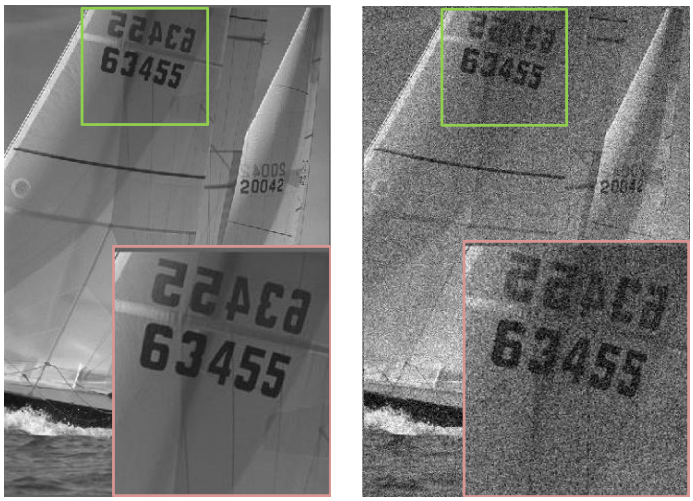
(c) BM3D [24]

(d) WNNM [30]

(e) BM3D+IRAISR

(f) WNNM+IRAISR

Figure 4.15: Visual comparison of *Man* image corrupted by $\sigma = 20$



(a) Ground truth (b) Noisy

(c) BM3D [24] (d) WNNM [30]

(e) BM3D+IRAISR (f) WNNM+IRAISR

Figure 4.16: Visual comparison of one image from Kodak dataset corrupted by $\sigma = 30$

(a) Ground truth (b) Noisy

(c) BM3D [24] (d) WNNM [30]

(e) BM3D+IRAISR (f) WNNM+IRAISR

Figure 4.17: Visual comparison of one image from BSD68 dataset corrupted by $\sigma = 50$

(a) Ground truth

(b) Noisy

(c) BM3D [24]

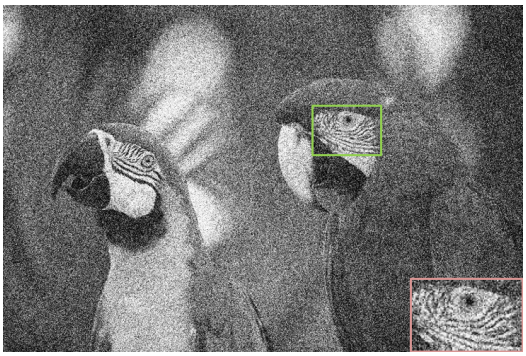
(d) WNNM [30]

(e) BM3D+IRAISR

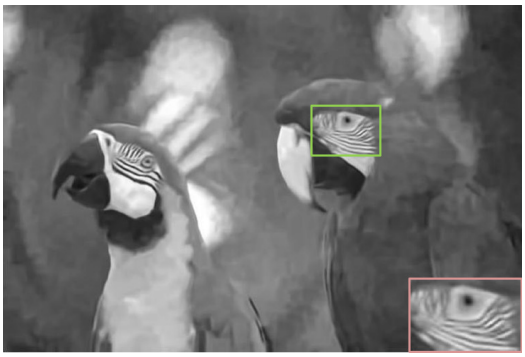
(f) WNNM+IRAISR

Figure 4.18: Visual comparison of one image from BSD68 dataset corrupted by $\sigma = 30$

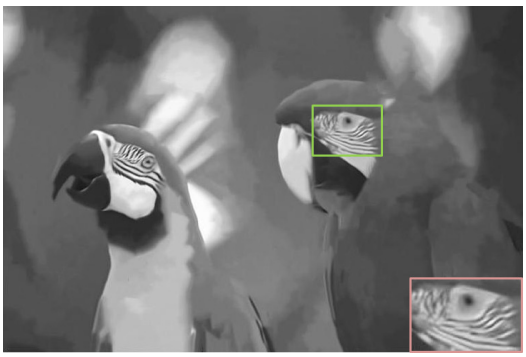
(a) Ground truth



(b) Noisy

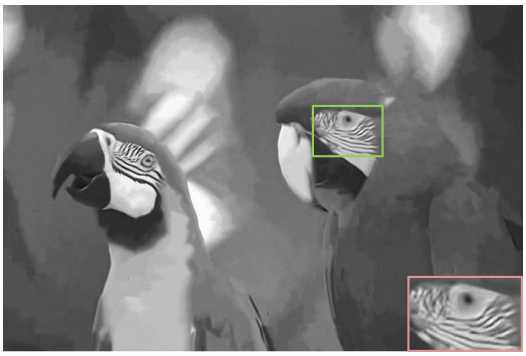


(c) BM3D [24]



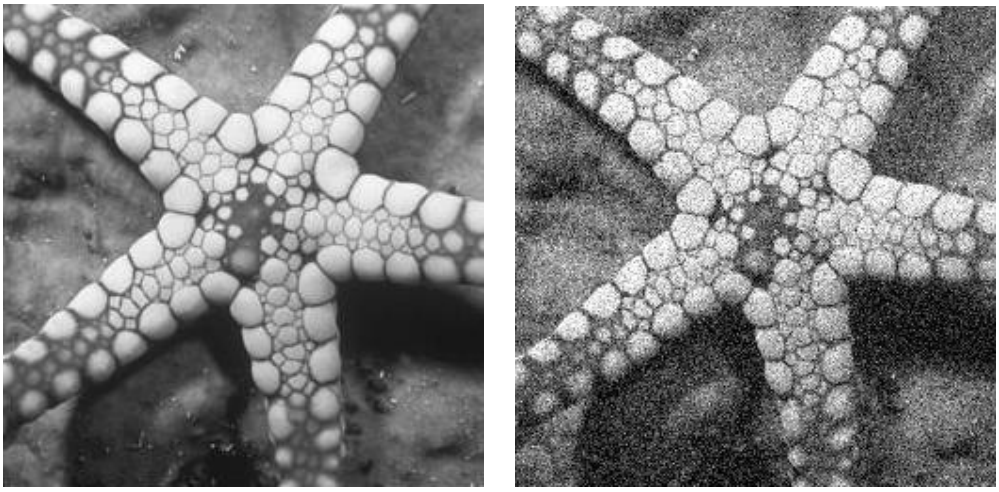
(d) WNNM [30]

(e) BM3D+IRAISR



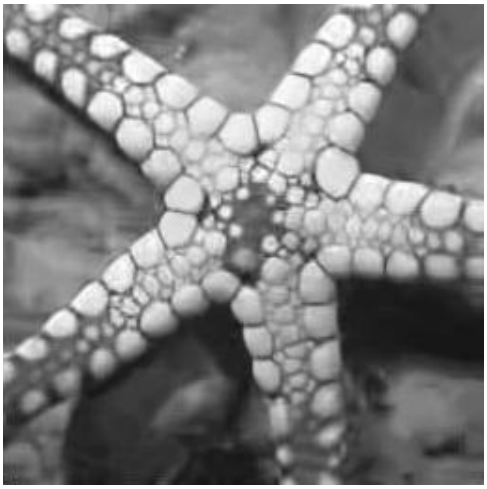
(f) WNNM+IRAISR

Figure 4.19: Visual comparison of one image from Kodak dataset corrupted by $\sigma = 50$

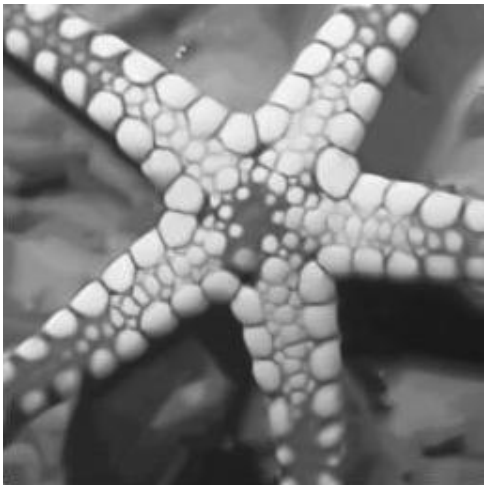


(a) Ground truth

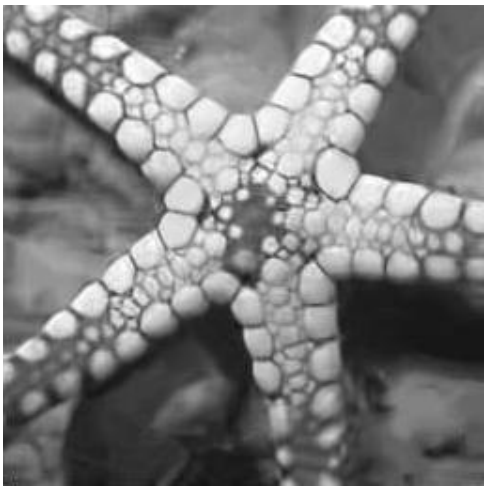
(b) Noisy



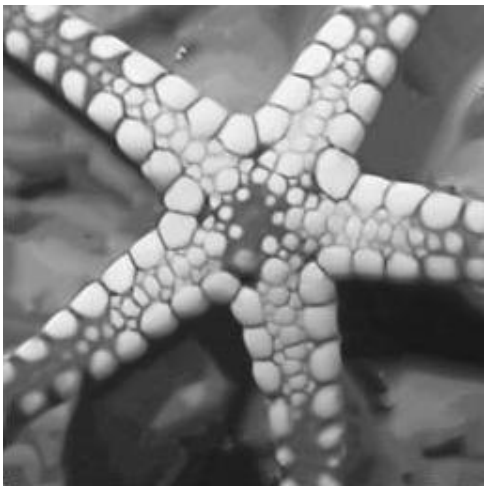
(c) BM3D [24]



(d) WNNM [30]



(e) BM3D+IRAISR



(f) WNNM+IRAISR

Figure 4.20: Visual comparison of one image from Set12 dataset corrupted by $\sigma = 30$

Chapter 5

Conclusion

5.1 Overall Conclusion of the Dissertation

This dissertation mainly focuses on the removal of mixed-noise that consists of AWGN and RVIN with the help of interpolation technique to efficiently preserve the image details and the removal of Gaussian noise using Improved RAISR with less number of filters to maintain the deteriorated high frequency information.

It is more difficult to remove the mixed-noise than single noise because the characteristics of noise are dissimilar to each other. In this dissertation, the removing of mixed-noise composed of AWGN and RVIN is considered. Image denoising methods for AWGN are basically classified into spatial-domain based methods, transform domain-based methods, dictionary learning-based methods and CNN-based methods. Gaussian filtering also known as linear filtering can effectively remove GN but the edge and texture regions of the image tend to blurring effect because of the over-smoothing. Bilateral filtering is then advantageous to attenuate Gaussian noise due to its effectiveness in preserving the edges. Even though linear-based denoising methods can remove the noise, some image details are still disappeared to reconstruct as the original. In order to tackle this problem, nonlocal-based image denoising methods have been recently eminent in suppressing the noise because the basic idea behind these methods is searching the similar patches related to the target patch at different locations across the image.

Analogously, median filtering is the most traditional impulse noise removal method in image processing applications. The main concept of median filtering is that every pixel in the image is replaced by median value arranged in ascending order with its nearest neighborhood across the image. The demerit of this method is the blurring

effect in the desirable image details because it removes not only impulse noise but also the important image details when the noise level is enormous. Therefore, various types of modified median filtering techniques have been successfully utilized in attenuating impulse noise for a few decades. Fundamentally, these methods consist of noise detection followed by noise filtering.

Mixed-noise removal method proposed in this dissertation typically consists of two steps. The first step is the integration of interpolation, DWM filter, down-sampling and BM3D filtering to denoise the noisy image. The second step is composed of re-detect process and BM3D to reconstruct the image. DWM filter is employed to remove RVIN and BM3D is utilized for the removal of Gaussian noise because these methods are state-of-the-art denoising methods. As most mixed-noise removal methods are detection based, the detection of impulse noise present in the mixed-noise may become an important role to upgrade the image quality without losing the image details. Hence, the input noisy image is initially interpolated before the detection of impulse noise in DWM filter by using multi-surface fitting for single frame in order to preserve the image details.

Interpolating the input noisy image not only reduces the impulse noise rate but also improves the accuracy of impulse noise detection. Therefore, it is very effective in the removal of RVIN. The output of DWM filter is also needed to down-sample because of the consideration of interpolation. Although almost all noises can be sufficiently removed in the first step, it can be seen that some image details are still damaged. Hence, the detection and removal method of RVIN is applied in the second step to overcome this problem. After the removal of RVIN, BM3D is exploited to remove the remaining noise. According to the experimental results presented in Chapter 3, the proposed two-step mixed-noise removal method using interpolation gives higher PSNR values and more excellent visual quality than the conventional denoising methods.

In Gaussian noise removal, an accurate Gaussian noise removal strategy followed by an Improvement of RAISR (IRAISR) is presented to successfully remove the noise without regressing some image features such as edges and textures. Not only the

patches extracted from the image processed by BM3D and WNNM , and the pixels from the reference image in the learning phase but also the patches from the denoised image in the testing phase are classified into hash classes. The filtered output is obtained by convolving the pre-learned filter generated from the learning phase with the patches extracted from the denoised image. The main contribution of IRAISR is the minimization of number of filters by two improvements including the reduction of the gradient angle by geometric conversion and the reduction of the classes for the gradient strength. In addition, Census transform (CT) is utilized by weighted averaging of the images processed by nonlocal-based denoising methods and the filtered output to restore the distorted image structure. Eventually, more advanced quantitative performance and stronger sharpness than the conventional denoising methods can be significantly acquired in accordance with the experimental results.

5.2 Limitations

The performance of image denoising in both mixed-noise and Gaussian noise can be determined by peak-signal-to-noise-ratio (PSNR) value as a quantitative metric. The visual evaluation of these denoising methods can be assessed depending upon the preservation of some important features such as artifacts, edges, textures and fine details in the image. In the removal of mixed-noise, it can be observed that some images which are plenty of texture regions such as *Barbara* and rich in smooth regions such as *Lena* images can be well restored by using the proposed mixed-noise removal method excluding *Airplane*, *Goldhill* and *Peppers* images with strong noise levels. DWM filter can only suppress RVIN and BM3D can also eliminate AWGN only. Experiments are conducted on 11 widely used natural images with different values of Gaussian noise and RVIN. Even though GN can be efficiently eliminated in the mixed-noise by BM3D whenever the noise level is increased, DWM filter cannot suppress the degraded impulse noise in the interpolated noisy images such as *Bridge*, *Mandrill* and *Milkdrop* images. This is because the number of miss-detected pixels are highly increased based on interpolating the noisy image and increasing the impulse noise

levels. As the handcrafted parameters such as threshold value τ for the detection of impulse noise in the second step of the proposed method and window size for DWM filter are chosen depending on the standard deviation of noise values, it provides some leeway to boost denoising performance.

There are mainly two portions in the proposed Gaussian noise removal method. The first portion is to remove noise by using nonlocal-based denoising methods namely: BM3D and WNNM. The second portion is to improve the denoising performance with the aid of Improved RAISR as a post-processing step without sacrificing the image details with less number of filters. The use of Improved RAISR is very effective in enhancing the image quality with high computational efficiency. The disadvantage of this approach is the execution time for denoised process in both learning and testing phase. It takes one day for BM3D and three days for WNNM in each noise level to learn the filters for 191 training sets including 100 general images and 90 images from Yang.*et.al*. This is due to searching patches similar to the target patch for every pixel in the noisy image although nonlocal-based denoising methods can significantly produce higher performance and better visual quality than the local-based denoising methods. Moreover, the proposed method is applied in the noisy images with gray-scale where the noise level is already known.

5.3 Further Extensions

According to the advanced camera technology in mobile devices especially smart phones, the production of high quality images without losing some valuable image features within a rapid computation time plays a vital role in the local market. Many mobile phone companies are currently endeavoring to manufacture the products with high technology that are cost effective, durable, high-speed performance and good quality depending on the demands for end users. From a theoretical and research perspective, nonlocal-based denoising methods are not reliable to use in the real applications due to its high execution time. Hence, other noise removal methods should be selected to obtain an enhanced quantitative performance with less computation

time because Gaussian noise removal methods assigned in this dissertation are based on nonlocal self-similarity (NSS) models.

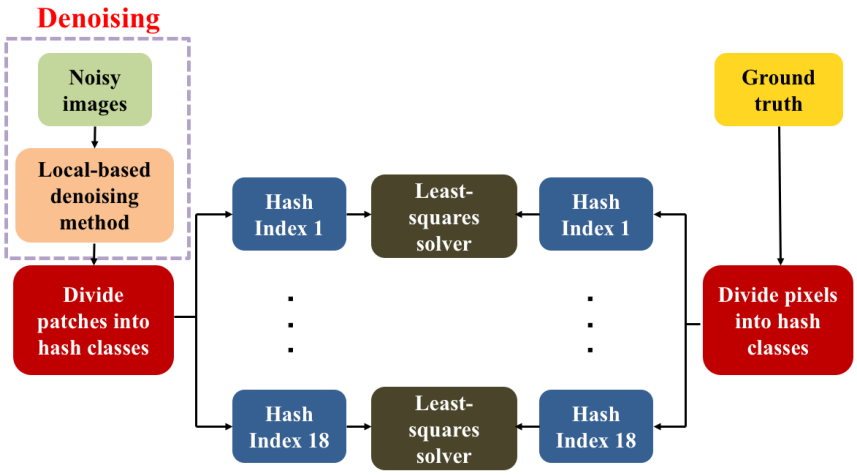


Figure 5.1: Learning phase in local-based denoising method

As a further extension, some local-based denoising methods such as bilateral filtering or wiener filtering will be utilized to remove Gaussian noise in both learning and testing phases as illustrated in Fig. 5.1 and Fig. 5.2, respectively instead of using nonlocal-based denoising methods. IRAISR is applied to the images processed by local-based denoising methods to enhance the performance of image by visually and quantitatively. Similarly to the nonlocal-based denoising methods, the patches extracted from the denoised images processed by local-based noise reduction method and the pixel from Ground truth are divided into hash classes in the learning phase. 18 filters are learned based on these image pairs by least square minimization problem.

In the testing phase, Gaussian noise can be removed by local-based denoising method to achieve noise-free image. The patches extracted from the denoised image are classified into hash classes. The filters are reduced from 216 to 18 filters by modifying the hash mechanism with two improvements including geometric conversion for the angle θ and the reduction of the strength λ without considering the pixel type. The filtered image can be obtained by convolving the patches from the denoised image with the pre-learned filters generated from the learning phase. Then, census transform (CT) is also used by blending the denoised image and the filtered output

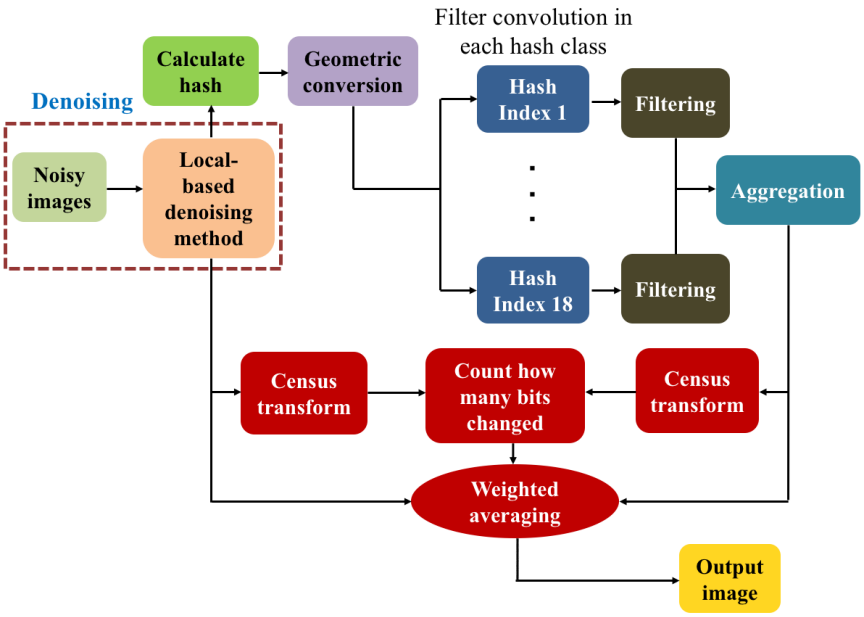


Figure 5.2: Testing phase in local-based denoising method

to get the final denoised image. Generally speaking, local-based denoising methods take faster execution time than nonlocal-based denoising methods. In addition, an excellent image quality with a rapid computational efficiency while efficiently restoring some image structures can be achieved by adding the improvement of RAISR to the denoised image as a post-processing step according to the experimental results as mentioned in Chapter. 4. It will be quite beneficial to us for using these methods in the real applications such as mobile devices if the denoising process can be improved within a low execution time.

Moreover, the proposed Gaussian noise removal method will be applied in real noisy images to be more practical and in RGB color images to distinguish the image features clearly as a future work. Furthermore, the proposed method will be utilized in the applications where the noise values are unknown (e.g., blind Gaussian denoising) by evaluating the noise based on the noise level estimation methods as a preliminary processing.

Bibliography

[1] *Fundamentals of digital image processing.* Prentice-hall, Inc, Upper Saddle River, 1989.

[2] J. Benesty, J. Chen, and Y. Huang, "Study of the widely linear wiener filter for noise reduction," in *Abstrasts of IEEE international conference on acoustics, speech and signal processing*, pp. 205–208, IEEE, Dallas, TX, USA, 2010.

[3] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the Sixth International Conference on Computer Vision*, (ICCV '98, IEEE Computer Society, Washington .DC, USA), pp. 839–846, 1998.

[4] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D, Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[5] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 12, pp. 629–639, Jul. 1990.

[6] L. Rudin and S. Osher, "Total variation based image restoration with free local constraints," in *Abstracts of the 1st international conference on image processing*, pp. 31–35, IEEE, 1994.

[7] C. Vogel and M. Oman, "Iterative methods for total variation denoising," *SIAM J Sci. Comput.*, vol. 17, pp. 227–238, Jan. 1996.

[8] Y. Lou, T. Zeng, S. Osher, and J. Xin, "A weighted difference of anisotropic and isotropic total variation model for image processing," *SIAM J Imaging Sci.*, vol. 8, pp. 1798–1832, Mar. 2015.

[9] Y. Hu and M. Jacob, "Higher degree total variation (hdtv) regularization for image recovery," *IEEE Trans. Image Process.*, vol. 21, pp. 2559–2571, May 2012.

[10] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, pp. 2419–2434, Nov 2009.

[11] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising methods, with a new one," *Multiscale Model, Simul.*, vol. 4, no. 2, pp. 490–530, 2005.

[12] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *CVPR*, 2005.

[13] G. Gilboa and S. Osher, "Nonlocal linear image regularization and supervised segmentation," *Multiscale Model, Simul.*, vol. 6, pp. 595–630, Jan 2007.

[14] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing," *SIAM Multiscale Model. Simul.*, vol. 7, no. 3, pp. 1005–1028, 2008.

[15] P. Coupe, P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot, "An optimized blockwise nonlocal means denoising filter for 3-d magnetic resonance images," *IEEE Trans. Med. Imaging*, vol. 27, pp. 425–441, April 2008.

[16] T. Thaipanich, B. Oh, P. Wu, D. Xu, and C. Kuo, "Improved image denoising with adaptive nonlocal means (anl-means) algorithm," *IEEE Trans. Consum. Electron*, vol. 56, pp. 2623–2630, April 2010.

[17] J. Wang, Y. Guo, Y. Ying, Y. Liu, and Q. Peng, "Fast non-local algorithm for image denoising," in *Abstracts of 2006 international conference on image processing*, pp. 1429–1432, IEEE, Atlanta, 2006.

[18] C. Pang, O. Au, J. Dai, W. Yang, and F. Zou, "A fast nl-means method in image denoising based on the similarity of spatially sampled pixels," in *Abstracts of*

2009 IEEE international workshop on multimedia signal processing, pp. 1–4, IEEE,Rio De Janeiro, 2009.

[19] D. Tschumperle' and L. Brum, "Non-local image smoothing by applying anisotropic diffusion pde's in the space of patches," in *Abstracts of the 16th IEEE international conference on image processing*, pp. 2957–2960, IEEE, Cairo, 2009.

[20] A. Kheradmand and P. Milanfar, "A general framework for regularized, similarity-based image restoration," *IEEE Trans. Image Process.*, vol. 23, pp. 5136–5151, Dec. 2014.

[21] L. Fan, X. Li, H. Fan, Y. Feng, and C. Zhang, "Adaptive texture-preserving denoising method using gradient histogram and nonlocal self-similarity priors," *IEEE Trans. Circuits Syst. Video. Technol. (in press)*, 2018.

[22] Y. Lou, P. Favaro, S. Soatto, and A. Bertozzi, "Nonlocal similarity image filtering," in *Abstracts of the 15th international conference on image processing*, pp. 62–71, ACM,Vietri sul Mare.

[23] S. Zimmer, S. Didas, and J. Weickert, "A rotationally invariant block matching strategy improving image denoising with non-local means," in *Abstracts of international workshop on local and non-local approximation in image processing*, pp. 103–113, IEEE,Lausanne, 2008.

[24] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[25] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "A nonlocal and shape-adaptive transform-domain collaborative filtering," in *Proc. Int. Workshop Local Nonlocal Approx. Image Process.*, pp. 1–8, 2008.

[26] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Bm3d image denoising with shape-adaptive principal component analysis," in *Proc. Workshop Signal Process. Adapt. Sparse Struct. Represent.*, pp. 1–6, 2009.

[27] C. Sutour, C. Deledalle, and J. Aujol, "Adaptive regularization of the nl-means:application to image and video denoising," *IEEE Trans. Image Process.*, vol. 23, pp. 3506–3521, Aug. 2014.

[28] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Abstracts of 2011 international conference on computer vision*, pp. 479–486, IEEE,Barcelona, 2011.

[29] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang, "Weighted nuclear norm minimization and its appliations to low level vision," *Int J. Computer Vis.*, vol. 121, pp. 183–208, Feb. 2017.

[30] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[31] M. Elad and M. Aharon, "Image denoising via learned dictionaries and sparse representation," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, pp. 895–900, 2006.

[32] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, pp. 3736–3745, Dec 2006.

[33] M. Aharon, M. Elad, and A. Bruckstein, "The k-svd:an algorithm for designing of overcomplete dictionaries for sparse representations," *IEEE Trans. Image Process.*, vol. 54, pp. 4311–4322, Nov. 2006.

[34] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, pp. 53–69, Jan 2008.

[35] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, pp. 2272–2279, Sep-Oct 2009.

[36] P. Chatterjee and P. Milanfar, "Clustering-based denoising with locally learned dictionaries," *IEEE Trans. Image Process.*, vol. 18, pp. 1438–1451, Jul. 2009.

[37] K. Zhang, X. Gao, D. Tao, and X. Li, "Multi-scale dictionary for single image super-resolution," in *Abstracts of 2012 IEEE conference on computer vision and pattern recognition*, pp. 1114–1121, IEEE,Providence, 2012.

[38] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 457–464, 2011.

[39] W. Dong, L. Zhang, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, pp. 1620–1630, April 2013.

[40] W. Dong, G. Shi, and X. Li, "Nonlocal image restoration with bilateral variance estimation: A low-rank approach," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 700–711, 2013.

[41] A. Eriksson and A. van den Hengel, "Efficient computation of robust weighted low-rank matrix approximations using the l1 norm," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 34, pp. 1681–1690, Sep. 2012.

[42] R. Liu, Z. Lin, and F. D. la Torre, "Fixed-rank representation for unsupervised visual learning," in *Abstracts of 2012 IEEE conference on computer vision and pattern recognition*, pp. 598–605, IEEE, Providence, 2012.

[43] Q. Guo, C. Zhang, Y. Zhang, and H. Liu, "An efficient svd-based method for image denoising," *IEEE Trans. Circuits Syst. Video. Technol.*, vol. 26, pp. 868–880, 2016.

[44] A. Rajwade, A. Rangarajan, and A. Banerjee, "Image denoising using the higher order singular value decomposition," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 35, pp. 849–863, April 2013.

[45] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 35, pp. 171–184, Jan 2013.

[46] C. Knaus and M. Zwicker, "Dual-domain image denoising," in *IEEE ICIP*, 2013.

[47] N. Pierazzo, M. Lebrun, M. E. Rais, J. M. Morel, and G. Facciolo, "Non-local dual image denoising," in *IEEE ICIP 2014*, 2014.

[48] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Abstracts of the 25th international conference on neural information processing systems*, vol. 1, pp. 341–349, ACM, Lake Tahoe, 2012.

[49] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser:residual learning of deep cnn for image denoising," *IEEE Trans. Image Process.*, vol. 26, pp. 3142–3155, Jul. 2017.

[50] K. Zhang, W. Zuo, and L. Zhang, "Ffdnet:toward a fast and flexible solution for cnn-based image denoising," *IEEE Trans. Image Process.*, vol. 27, pp. 4608–4622, Sep. 2019.

[51] C. Cruz, A. Foi, V. Katkovnik, and K. Egiazarian, "Nonlocality-reinforced convolutional neural networks for image denoising," in *IEEE Signal Process. Lett.*, vol. 25, pp. 1216–1220, Aug. 2018.

[52] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognition*, vol. 43, no. 4, pp. 1531–1549, 2010.

[53] I. Pitas and A. N. Venetsanopoulos, *Nonlinear digital filters: principles and applications*. Kluwer, 1999.

[54] D. Brownrigg, "The weighted median filter," *Commun.ACM*, vol. 27, pp. 807–818, Aug 1984.

[55] A. Nieminen, P. Heinonen, and Y. Neuvo, "A new class of detail-preserving filters for image processing," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. PAMI-9, pp. 74–90, Jan 1987.

[56] S. J. Ko and Y. H. Lee, "Center weighted median filters and their applications to image enhancement," *IEEE Trans. Circuits Syst.,*, vol. 38, pp. 984–993, Sep 1991.

[57] E. J. Coyle, J. H. Lin, and M. Gabbouj, "Optimal stack filtering and the estimation and structural approaches to image processing," *IEEE Trans. Acoust., Speech Signal Process*, vol. 37, pp. 2037–2066, Dec 1989.

[58] T. Sun and Y. Neuvo, "Detail-preserving median based filters in image processing," in *Pattern Recognit. Lett.,*, vol. 15, pp. 341–347, April 1994.

[59] T. Chen and H. R. Wu, "Space variant median filters for the restoration of impulse noise corrupted images," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 48, pp. 784–789, Aug 2001.

[60] T. Chen, K. K. Ma, and L. H. Chen, "Tri-state median filter for image denoising," *IEEE Trans. Image Process.*, vol. 8, pp. 1834–1838, Dec 1999.

[61] T. Chen and H. R. Wu, "Adaptive impulse detection using center-weighted median filters," in *IEEE Signal Process. Lett.*, vol. 8, pp. 1–3, Jun 2001.

[62] V. Crnojevic, V. Senk, and Z. Trpovski, "Advanced impulse detection based on pixel-wise mad," in *IEEE Signal Process. Lett.*, vol. 11, pp. 589–592, July 2004.

[63] S. Akkoul, R. Ledee, R. Leconge, and R. Harba, "A new adaptive switching median filter," in *IEEE Signal Process. Lett.*, vol. 17, pp. 587–590, Jun 2010.

[64] Y. Dong and S. Xu, "A new directional weighted median filter for removal of random-valued impulse noise," *IEEE Signal Processing Letters*, vol. 14, no. 3, pp. 193–196, 2007.

[65] W. Luo, "A new efficient impulse detection algorithm for the removal of impulse noise," *IEICE Trans. Fundam. Electron., Commun., Comput.*, vol. E88-A, pp. 2579–2586, Oct 2005.

[66] G. Pok, J. C. Liu, and A. S. Nair, "Selective removal of impulse noise based on homogenity level information," *IEEE Trans. Image Process.*, vol. 12, pp. 85–92, Jan 2003.

[67] Y. Dong, R. H. Chan, and S. Xu, "A detection statistic for random-valued impulse noise," *IEEE Trans. Image Process.*, vol. 16, pp. 1112–1120, Mar 2007.

[68] U. Ghanekar, A. k. Singh, and R. Pandey, "A contrast enhancement-based filter for removal of random valued impulse noise," *IEEE Signal Processing Letters*, vol. 17, no. 1, pp. 47–50, 2010.

[69] C. Y. Lien, C. C. Huang, P. Y. Chen, and Y. F. Lin, "An efficient denoising architecture for removal of impulse noise in images," *IEEE Transactions on Computers*, vol. 62, no. 4, pp. 631–643, 2013.

[70] N. I. Petrovic' and V. Crnojevic', "Universal impulse noise filter based on genetic programming," *IEEE Trans. Image Process.*, vol. 17, pp. 1109–1120, Jul 2008.

[71] Y. Xiao, T. Zeng, J. Yu, and M. K. Ng, "Restoration of images corrupted by mixed gaussian-impulse noise via 11-10 minimization," *Pattern Recognition*, vol. 44, no. 8, pp. 1708–1720, 2011.

[72] B. Xiong and Z. Yin, "A universal denoising framework with a new impulse detector and nonlocal means," *IEEE Trans. Image Process.*, vol. 21, pp. 1663–1675, April 2012.

[73] J. Liu, X. C. Tai, H. Huang, and Z. Huan, "A weighted dictionary learning model for denoising images corrupted by mixed noise," *IEEE Trans. on Image Processing*, vol. 22, pp. 1108–1120, March 2013.

[74] J. Jiang, L. Zhang, and J. Yang, "Mixed noise removal by weighted encoding with sparse nonlocal regularization," *IEEE Trans. on Image Processing*, vol. 23, pp. 2651–2662, June 2014.

[75] L. Liu, L. Chen, C. L. P. Chen, Y. Y. Tang, and C. M. pun, "Weighted joint sparse representation for removing mixed noise in image," *IEEE Trans. on Cybernetics*, vol. 47, pp. 600–611, March 2017.

[76] T. Huang, W. Dong, X. Xie, G. Shi, and X. Bai, "Mixed noise removal via laplacian scale mixture modeling and nonlocal low-rank approximation," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3171–3186, 2017.

[77] J. Zhang, R. Xiong, C. Zhao, S. Ma, and D. Zhao, "Exploiting image local and nonlocal consistency for mixed gaussian-impulse noise removal," *2012 IEEE International Conference on Multimedia and Expo*, 2012.

[78] Z. Lin, "A nonlocal means based adaptive denoising framework for mixed image noise removal," *2013 IEEE International Conference on Image Processing*, 2013.

[79] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2774–2781, Jun 2014.

[80] Y. Chen, W. Yu, and T. Pock, "On learning optimized reaction diffusion processes for effective image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 5261–5269, Jun 2015.

[81] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion:a flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell*, 2016.

[82] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.,*, pp. 1097–1105, 2012.

[83] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, pp. 448–456, 2015.

[84] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 770–778, 2016.

[85] T. Yamaguchi, A. Suzuki, and M. Ikehara, "Detail preserving mixed noise removal by dwm filter and bm3d," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E100.A, no. 11, pp. 2451–2457, 2017.

[86] T. Yamaguchi and M. Ikehara, "Fast and high quality image interpolation for single-frame using multi-filtering and weighted mean," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E100.A, no. 5, pp. 1119–1126, 2017.

[87] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, 1981.

[88] R. Gonzalez and R. Woods, *Digital image processing.* 3rd edn. Prentice-Hall, Inc, Upper Saddle River, 2006.

[89] R. Yang, L. Yin, M. Gabbouj, J. Astola, and Y. Neuvo, "Optimal weighted median filtering under structural constraints," *IEEE Transactions on Signal Processing*, vol. 43, no. 3, pp. 591–604, 1995.

[90] R. Timofte, V. De, and L. V. Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *2013 IEEE International Conference on Computer Vision*, pp. 1920–1927, 2013.

[91] R. Timofte, V. D. Smet, and L. V. Gool, "A+ adjusted anchored neighborhood regression for fast super-resolution," in *Computer Vision- ACCV 2014,ed. D.Cremers , I. Reid , H.Saito and M.H. Yang ,Cham*, pp. 111–126, Springer International Publishing, 2015.

[92] C. Y. Yang and M. H. Yang, "Fast direct super-resolution by simple functions," in *International Conference on Computer Vision*, pp. 561–568, 2013.

[93] C. Dong, C. C. Change, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 295–307, Feb 2016.

[94] Y. Romano, J. Isidoro, and P. Milanfar, "Raisr: Rapid and accurate image super resolution," *IEEE Transactions on Computational Imaging*, vol. 3, pp. 110–125, March 2017.

[95] S. C. Jeong and B. C. Song, "Training-based super-resolution algorithm using k-means clustering and detail enhancement," in *Proc.Eur. Signal Process. Conf*, pp. 1791–1795, 2010.

[96] G. Yu, G. Sapiro, and S. Mallat, "Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity," *IEEE Trans. Image Process*, vol. 21, pp. 2481–2499, May 2012.

[97] V. Papyan and M. Elad, "Multi-scale patch-based image restoration," *IEEE Trans. Image Process*, vol. 25, pp. 249–261, Jan 2016.

[98] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Computer Vision ECCV'94, ed.J.O.Eklundh, Berlin, Heideberg*, pp. 151–158, Springer Berlin Heideberg, 1994.

[99] X. Feng and P. Milanfar, "Multiscale principal components analysis for image local orientation estimation," in *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers,2002*, vol. 1, pp. 478–482, 2002.

[100] M. Bevilacqua, A. Roumy, C. Guillemot, and M. line Alberi Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proceedings of the British Machine Vision Conference*, pp. 135.1–135.10, BMVS Press, 2012.