4-21-2014 12:00 AM

Adaptive Edge-guided Block-matching and 3D filtering (BM3D) Image Denoising Algorithm

Mohammad Mahedi Hasan
The University of Western Ontario

Supervisor
Dr. Mahmoud R. El-Sakka
The University of Western Ontario

Graduate Program in Computer Science

A thesis submitted in partial fulfillment of the requirements for the degree in Master of Science

© Mohammad Mahedi Hasan 2014

Adaptive Edge-Guided *Block-Matching and 3D Filtering* (BM3D) Image Denoising Algorithm

(Thesis format: Monograph)

by

Mohammad Mahedi <u>Hasan</u>

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Abstract

Image denoising is a well studied field, yet reducing noise from images is still a valid challenge. Recently proposed *Block-matching and 3D filtering* (BM3D) is the current state of the art algorithm for denoising images corrupted by *Additive White Gaussian noise* (AWGN). Though BM3D outperforms all existing methods for AWGN denoising, still its performance decreases as the noise level increases in images, since it is harder to find proper match for reference blocks in the presence of highly corrupted pixel values. It also blurs sharp edges and textures. To overcome these problems we proposed an edge guided BM3D with selective pixel restoration. For higher noise levels it is possible to detect noisy pixels form its neighborhoods gray level statistics. We exploited this property to reduce noise as much as possible by applying a pre-filter. We also introduced an edge guided pixel restoration process in the hard-thresholding step of BM3D to restore the sharpness of edges and textures. Experimental results confirm that our proposed method is competitive and shows better results than BM3D in most of the considered subjective and objective quality measurements, particularly in preserving edges, textures and image contrast.

Keywords

image denoising, additive white gaussian noise, block matching, sparsity, transform domain filtering, edge guidance, and pre-filtering.

Dedication

To my loving parents

Mahbubur Rahman and **Nurun Nahar**

without whom none of my success would be possible

Acknowledgments

Table of Contents

List of Tables

List of Figures

Chapter 1

1 Introduction

A digital image is composed of set of pixels which is defined as a two dimensional function, $f(x, y)$, where x and y are spatial coordinates. The value of f at any particular pair of coordinates (x, y) is called the gray level or intensity of the image at that location. The image becomes a digital image when x, y, and the values of f are all finite, discrete quantities. These values of f are generally referred to as pixels. Pixel values in images can be noisy. Noise in images is mainly caused by sensors during acquisition, environments (e.g. poor illumination) or during transmissions. No matter how good the image acquisition devices are, an image improvement is always sought-after to improve their range of applications in various fields. The process of estimating the original image by reducing the noises from noise-contaminated image is referred to as image denoising in image processing. Image denoising is a very important task in image processing as a process itself or as an element of other image processing tasks.

Image denoising is a well-studied field and yet it's still one of the most active research areas in image processing and computer vision. It's a pre-requisite for many image processing tasks such as image segmentation, image restoration, object recognition, image classification, and image registration, where estimating the true signal is crucial to accomplish desirable results.

The form of the noise can be additive, which is generally independent of image data, or multiplicative, which is dependent on image data.

The formula for additive noise is

$$z(x) = y(x) + \eta(x), \qquad\qquad \textbf{1.1}$$

whereas the formula for multiplicative noise is

$$z(x) = y(x) \times \eta(x). \qquad\qquad \textbf{1.2}$$

Here, x represents the location of pixels, $y(x)$ is the original signal, while $\eta(x)$ denotes the noise introduced to form the corrupted image $z(x)$.

Most of the images are assumed to be contaminated by additive random noise and can be modeled by a Gaussian distribution. Hence, this type of noise is referred to as *Additive White Gaussian Noise* (AWGN). AWGN is also probably the simplest and most commonly used model in the image denoising literature. As AWGN is random in nature and it corrupts almost all areas of images, it is challenging to remove AWGN from images. It becomes increasingly difficult to preserve the small details of an image as the error level increases.

This thesis work proposes a few improvements over state of the art method for reducing the effect of AWGN, famously known as *Block Matching and 3-D Filtering* (BM3D) [1].

1.1 Motivations

The BM3D algorithm exploits non-local image modeling through the signature method of grouping and collaborative filtering in transform domain. It capitalizes on two principal properties of natural scene images: the existence of mutually similar patches within a close neighborhood and the local correlation of pixel values within a single patch. Grouping similar 2D patches into 3D data arrays allows us to exploit both intra-patch correlation and inter-patch correlation based on the above two assumptions. As a result, the group enjoys correlation in all three dimensions, allowing a very high sparse representation of the image in the transform domain. Here by the term sparse, we refer to

linear-transform domain representations which have few high magnitude coefficients and many low magnitude coefficients. Sparseness refers to the energy compaction property (i.e. most of the image details are represented by few large coefficients while noise is spread across a range of small coefficients). This kind of sparse representation allows us to separate the noise from the true signal by applying shrinkage on the coefficients in transform domain. Thus BM3D acts as a very efficient and powerful denoiser and to our knowledge it exhibits the best performance for AWGN.

BM3D exhibits good results when the number of matching blocks in the defined neighborhood is plenty enough (e.g. repeating patterns, textures and uniformed areas) to achieve high sparsity, where nonlocal image modeling is suitable. Also the assumption that image content is highly correlated on a square patch is not always true for natural image scenes. If the patch contains curved edges, small image details or singularities, the non-adaptive transform of BM3D cannot produce a sparse representation of the image, resulting in poorer denoising performance introducing certain artifacts.

The performance of BM3D also sharply degrades as the noise level increases. At relatively higher noise levels, due to the increase of noisy pixels, it becomes increasingly harder for BM3D to differentiate between noise and true image textures, edges, and image details.

1.2   Thesis contributions

To address the problem of BM3D for irregular image shapes or textures, curved edges and singularities, we propose an *Adaptive Edge-Guided Block-Matching and 3D filtering* (AEG-BM3D) algorithm. In our method, we try to adapt the estimation of pixel values according the edge activity found in a given block. We extract the edges from the corrupted image after the first stage of BM3D and use this as guidance to better estimate the pixel values. The edge information allows us to choose different block sizes for different types of regions within the same image. Our method tries to integrate the local adaptively to image features with the nonlocal image modeling of BM3D.

To address the performance issue of BM3D for higher noise levels, we also proposed a simple denoising pre-filter to show that the pre-filters can effectively increase the performance of the denoising at little computational overhead when the additive noise is strong in images. We applied a simple selective median filter based on an empirically selected threshold value on the raw corrupted input image. That is, we don't apply the median filter for all the image pixels, except the one which has been classified as a corrupted one. Our method searches the whole image for corrupted pixels in a raster scan fashion, and we apply the filter only for the pixels which is highly likely to be corrupted. Experimental results show that the proposed method shows a significant improvement over current state of the art BM3D algorithm in terms of both subjective and objective evaluation, particularly when the noise level is high ($\sigma > 60$).

1.3   Thesis outline

This thesis is divided into five chapters including this introductory discussion, Chapter 1. The rest of the thesis is organized as follows. Chapter 2 discusses the noise model considered for our work. It also presents a brief introduction of the vast field of image denoising, roughly classified into spatial domain filters and transform domain filters. At the end of this chapter we discussed our method of interest that is, Block Matching and 3D Filtering or BM3D, and its various variants proposed by various literatures till now, along with its applications in practical image processing problems. In Chapter 3, we described our proposed method in detail. Experimental results and relevant analysis are presented in Chapter 4. Finally, Chapter 5 offers the concluding remarks and future directions of the presented research work.

Chapter 2

2 Background

Noise models are of particular importance in image denoising as most denoising methods work well with a particular noise model. Probabilistic models best reflect the randomness of the noise within images. In image denoising applications, parametric models (with few parameters) of the *probability density function* (PDF) are most commonly used. This chapter introduces the noise model considered for this thesis, AWGN, along with a detailed review of existing filters to remove this type noise.

2.1 Additive white Gaussian noise

A random signal with flat (constant) power spectral density is known as white noise. In Additive White Gaussian Noise the image is contaminated with a linear combination of white noise with a constant spectral density, where the amplitude has a Gaussian distribution. Most often, it is assumed that Gaussian noise necessarily means white noise, which is not correct also. The term *Gaussian* refers to the probability density function, that is, the probability distribution with respect to the signal value, while the term *white* refers to the way signal power is distributed over time among frequencies. It is possible to have *Poisson* white noise, for example, just like *Gaussian* white noise.

Gaussian noise is the most common type of noise found in natural images. It represents many real world situations and generates mathematically tractable model in both the spatial and transform domain. White Gaussian noise is also commonly found in other fields than image processing such as music, electronics engineering, and acoustics etc. The two principal sources of Gaussian white noise in digital images are the acquisition

stage and the transmission stage. During acquisition, noise may be generated by sensor due to poor illumination and/or high temperature while electronic circuit may generate noise during transmission.

Moreover, dealing with the uniform Gaussian noise makes the discussion of image denoising methods much easier. Two papers published in 2011 on the Anscombe variance-stabilizing transform by M. Mäkitalo and A. Foi (for low-count Poisson noise) [39] and A. Foi (for Rician noise) [40] argue that, when combined with suitable forward and inverse *variance stabilizing transformation* (VST) transformations, methods designed for AWGN work just as well as *ad hoc* algorithms based on signal dependent noise models. This also explains why, in most of the image denoising literature noise is assumed to uniform, white, and Gaussian.

The PDF of a Gaussian random variable, ω, is given by

$$f(\omega) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\omega^2}{2}\right), \qquad \omega \in \Re. \qquad \textbf{2.1}$$

The mean of ω is zero and its variance is *1*.

A general Gaussian random variable is given by

$$z = \sigma\omega + \mu, \qquad \textbf{2.2}$$

where, the mean of the variable z is μ, whereas σ^2 is the variance of z.

We can rewrite the probability density function of Equation 2.1 as follows:

$$f(z) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{(z-\mu)^2}{2\sigma^2}\right). \qquad \textbf{2.3}$$

In case of image, z represents the gray level of pixel values.
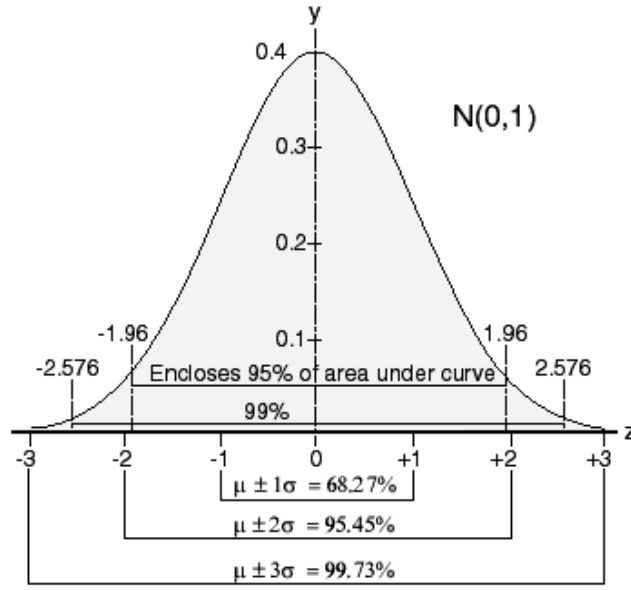
Figure 2-1: Probability density function for the Gaussian noise model [98].

Figure 2-1 illustrates the Gaussian probability density function. Approximately 68.27% of the values are contained between $\mu \pm \sigma$ and 95.45% of the values are contained between $\mu \pm 2\sigma$.

2.2   Existing filters to remove AWGN

In this section we will try to give an introduction to the common AWGN filtering techniques in past few decades. Throughout the rest of this thesis, unless the noise model is explicitly specified, we implicitly assume the AWGN model. We broadly classify the image denoising filters into two major categories: *Spatial Domain Filters* and *Transform Domain Filters*. We will also attempt to group them together according to their characteristics within these categories.

2.2.1   Spatial Domain Filters

One of the most common ways to reduce noise from images is to use *Range filters*. These classes of filters are computationally simple to calculate, easy to implement and are neighborhoods adaptive to the image data. Bilateral filter [41], the SUSAN filter [93] and the sigma filter [43] are noteworthy among these filters. The range filters use nonlinear weighted averaging with adaptive weights that depends on the spatial distance and the photometric distance, which is the distance between the pixel intensities (i.e. the image range). The general equation for the range filters is

$$y(x) = \sum_{k \in x} z(k) \, g_{Sp}(x - k) \, g_{rng}(z(x) - z(k)),$$
<div align="right">2.4</div>

where $z(k)$ is the noisy signal and $y(x)$ is the filtered or denoised signal. The function g_{sp} depends on the spatial distance between pixels and the function g_{rng} depends on the distance between intensities. The g_{rng} function here gives the range filters ability to adapt to image data and better preserve salient details such as edges, without blurring while doing spatial smoothing in regions with relatively homogeneous intensities. However, as the noise level increases it becomes increasingly difficult to differentiate between noisy pixels and their noise free counterparts with edges, resulting in decreased performance of range filters. Staircase effects are also visible in these types of filters. One approach to overcome this drawback is to take advantage of the similarities between image neighborhoods (e.g. nonlocal filters), which will be discussed later in this section.

Some satisfactory results have been obtained for image restoration through *Bayesian filters*. The general idea in the Bayesian estimation framework is to find the noise free image given the prior information of the noise and the observed noisy image z. The Bayes formula for the posterior probability is given below:

$$p_{post}(\gamma|z) = \frac{p_{fit}(z|\gamma)p_{prior}(\gamma)}{p_z(z)}.$$
<div align="right">2.5</div>

Here, z is the image being observed or the noisy image, while γ is an estimation of the noise free image y, the image prior $p_{prior}(\gamma)$ denotes the probability that γ belongs to family of noise free images, the *likelihood* (or *data fit*) $p_{fit}(z|\gamma)$ is the conditional probability of generating z according to the assumed noise model from γ, and $p_z(z)$ is the marginal probability of obtaining the noisy realization z. Bayesian estimation based filters are based on finding a solution γ using the posterior probability $p_{post}(\gamma|z)$ from Equation 2.5. The main challenge for Bayesian estimation in image processing is the derivation of a better image prior p_{prior} [89].

Another class of spatial domain filters is the *Variational and PDE-based filters*. The *Total Variation* (TV) minimization was introduced by *Rudin, Osher and Fatemi* [44] as a regularizing functional for image denoising. *Total variation denoising* (TVD) is defined in terms of an optimization problem by minimizing a particular cost function. TVD removes unwanted detail whilst preserving sharp edges in the original signal. Numerous methods have been proposed to solve the TDV problem [45] [46] [47]. The original TV model considered the removed noise as an error and no longer considered them in the reconstruction process. However, some structures and textures are always present in the removed errors. Later on, the authors tried to reduce this effect by using the *Rudin, Osher, and Fatemi* total variation minimization iteratively [48] [49]. Total variation is also used by researchers for general image restoration problems such as compressed sensing, interpolation, deconvolution, etc. [50] [51].

Partial differential equations (PDE) based diffusion techniques are yet another effective way to denoise images and are very similar to variational techniques. These technique has wide range of application in image processing [53] [54] such as image enhancement/denoising, edge detection, and flow field visualization, since the first model was introduced by Perona and Malik in 1987 [52]. This type of diffusion can effectively remove noises and at the same time can enhance edges. One of the most widely used PDE formulation in image processing is the anisotropic diffusion proposed my Perona and Malik in 1990 [55], which preserves the edges by guiding the smoothing with spatial derivative. A range of applications of PDEs in image processing can be found [53] [54] [56].

In recent years, the *Non-Local means* (NL-means) algorithm, has attained great deal of attention within the image processing community [57] [58]. A detailed study of NL-means with various extensions can be found in [58]. Fundamentally, it is a relatively simple generalization of the range filters like bilateral filter, in particular the point-wise photometric term in the similarity kernel of range filters is replaced by the block-based (or patch-based) similarity kernel. Another difference is that the geometric distance between the blocks is in fact ignored, which leads to the strong contribution of blocks that may not be essentially near the pixel of interest. The NL-means algorithm tries to exploit the property that natural images have high degree of self-similarity or redundancy between image neighborhoods, from which the name Non-Local originated. It is worth mentioning that the NL-means filters exploits resemblance between surrounding neighborhoods while the range filters exploits resemblance between individual pixels. Boulanger and Kervrann [59] demonstrated the true potential of NL-means filters with the *optimal spatial adaptation* (OSA) which is based on adaptive estimation neighborhoods. In recent years, the non-local priors variational formulation has also became a very popular research topic [60] [61] [62] in the image processing community. In these approaches, the authors have formulated the nonlocal image modeling as particular regularization functionals.

2.2.2   Transform Domain Filters

Since their introduction, multiscale transforms such as wavelets, ridgelets, curvelets etc. [64] [65] [66] [67]. has been widely use by the image processing community to get sparse representation of images [68]. A recent study [63] showed that natural images contain areas which are similar to other areas at the same resolution/orientation and across resolution/orientations. Multiscale transform is concerned with representation and analysis of signal at more than one resolution based on the fact that features that might go undetected at one resolution may be easily detected at another resolution. One of the most commonly used multiscale transform is wavelet decomposition. In wavelet decomposition, a wavelet generating function ψ of zero mean is used to generate a dictionary of basis elements, which is expressed in the continuous domain as follows

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}} \psi \left(\frac{t - u}{s} \right), \qquad\qquad \textbf{2.6}$$

where, *s* and *u* are the scale and shift (translation) parameters respectively. Multiscale decompositions are localized in both time and frequency and, possibly with varying orientation whereas transforms with fixed spatial localization (e.g. short time Fourier transform) or trigonometric transforms (e.g. DCT, DFT) are only localized in frequency.

In recent years, the over-completeness of multiscale representations has led to much interesting research problems in image restoration [68] [69] [70]. Formally, an over-complete dictionary contains prototype signal-atoms and any signal can be represented by more than one combination of these atoms. Over-completeness is required for translation invariance [69] of multiscale representations. Dual-tree complex wavelet [71] and steerable pyramid [72] [73] are amongst the most commonly used over-complete transforms.

All of the above mentioned methods use fixed dictionaries. Dictionaries can also be adaptive to the input image. *Independent Component Analysis* (ICA) [75] and *Principal Component Analysis* (PCA) [74] are two well recognized such data adaptive transforms that have found lots of interesting applications in image processing. PCA is particularly successful in reconstructing oscillatory patterns and textures while ICA has been successfully applied in image denoising, as it can achieve better sparsity for natural images. To name a few other adaptive transform domain filters that have been applied in image denoising very successfully, we would like to mention shape adaptive DCT or SA-DCT [86], *point-wise shape adaptive DCT* (P.SA-DCT) [76] [87], and K-SVD [34]. Among these methods we would like single out the state of the art method, K-SVD, a generalization of the k-means clustering method, which is regarded as one of the major advancements in the field of adaptive transform domain filters. It is an iterative method that iteratively alternates between sparse coding coefficients of the examples based on the current dictionary and a method of updating the atoms in the dictionary to better fit the data.

Among the shrinkage filters, Wiener filters showed good potential for denoising. For AWGN a parametric version of Weiner filter, $R(u, v)$, is used which is defined by the following equation,

$$R(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + K}$$ **2.7**

where, $H^*(u, v)$ is the complex conjugate of the degradation function $H(u, v)$ and K is a pre specified constant. However, wiener filtering for white noise tends to weaken the energy signal spectral coefficients leading to poor denoising performance.

In recent years, wavelet shrinkage filters has became one of the most important and powerful tools for image and video processing. The wavelet transform has the important properties such as compression or sparsity, which means that wavelet transforms tends to be sparse. Therefore, wavelet shrinkage is based on the fact that in the wavelet domain, a signal is represented with few large coefficients, which are processed or kept according to particular shrinkage function, whereas noises in signal is distributed across small coefficients, which are removed. The *WaveShrink* or the shrinkage of wavelet coefficients was introduced and extensively studied by Donoho and Johnstone in their seminal work [77] [78]. Donoho and Johnstone proposed two basic shrinkage algorithms [77]: soft thresholding and hard-thresholding, governed by the following equations:

$$D_{soft}(u, \tau) = \begin{cases} sign(u)(|u| - \tau), & if \ |u| > \tau \\ x, & otherwise \end{cases},$$ **2.8**

and $$D_{hard}(u, \tau) = \begin{cases} u, & if \ |u| > \tau \\ 0, & otherwise \end{cases}.$$ **2.9**

Here, τ is the thresholding parameter, which is predefined, and selection of an optimal value for τ is a critical problem. The authors recognized this issue and proposed an expression of the optimal threshold as a function of the noise variance σ^2 and the number of samples N,

$$\tau = \sqrt{2\,\sigma^2 \ln n}\;.$$
<div align="right">**2.10**</div>

This is known as *universal threshold*, application of which in wavelet domain is called *VisuShrink* [77].

The above mentioned shrinkages are non-adaptive and don't perform satisfactorily in most cases. A lot of work has been done to propose different data-adaptive thresholding methods to improve the performance of the *WaveShrink* estimators. Donoho and Johnstone themselves acknowledged this drawback and proposed the *SureShrink* approach [78] in which they introduced an data adaptive threshold that chooses the thresholding parameter, τ, by minimizing the *Steins unbiased risk estimator* (SURE) [79]. More recent studies have shown that better denoising can be achieved by exploiting intra-scale and inter-scale correlations of the wavelet coefficients. One such method is *ProbShrink* [80], which is driven by the estimation of the probability that a given coefficient contains considerable level of information. The bivariate shrinkage or *BiShrink* [81] exploits inter-scale correlations by using a new non-gaussian bivariate distribution to model the dependencies. In another article [82], the authors, Sendur and Selesnick, accounted the intra-scale dependency by extending their previous approach. Luisier *et al.* proposed a new SURE-based method *SURE-LET* [83] to wavelet based image denoising exploiting inter-scale correlation, which was later extended by Yan *et al.* [84] including inter-scale correlation. More recently, the *trivariate shrinkage* was introduced by Yu *et al.* [85], here wavelet coefficients are modeled as a trivariate Gaussian distribution and then a trivariate shrinkage filter by using the *maximum a posteriori* (MAP) estimator is then derived. It successfully models both the inter-scale and intra-scale correlations of wavelet coefficients. All these shrinkage based methods discussed above rely on one fundamental property, the sparsity of natural images in wavelet domain.

However for our study, the most important class of filters is the *non-local transform domain* filters. This class of filters is blessed by both the sparsity of the transform domain representations and the nonlocal modeling. Mallat [88] proposed a geometric grouplet

transform, which is a multi-scale adaptive transform in which the main component is a modified Haar dyadic decomposition. Though this method doesn't exhibit competitive results in terms of MSE, it effectively reconstructs image details. BM3D, our algorithm of interest, falls in this class of non-local transform domain filters, which is a recent development by Dabov *et al.* [1]. BM3D is basically a generalization of their previous work [2]. In that work, the authors exploited nonlocal modeling by grouping similar neighboring blocks, sparsity in transform domain by collaborative filtering and aggregation by combining different estimates from the previous collaborative filtering stage. Based on the achievement of BM3D in image denoising it is extended to other image processing tasks successfully (e.g. image restoration, deblurring, sharpening and video denoising). A detailed discussion on BM3D is in order.

2.3   Introduction to BM3D and its Variants

In past few years there has been a growing interest in the development of sparse representation of signals. In image denoising, this approach has gained much traction and led to several state of the art denoising algorithms, such as K-SVD [34] and BM3D [1]. Image denoising by sparse 3D transform-domain collaborative filtering [1], famously known as block-matching and 3D filtering, which exploits nonlocal image modeling [5] with a method termed grouping and collaborative filtering was proposed in 2007 by Dabov *et al.* It was an improvement over the novel algorithm [2] proposed by the same authors. In this section we will investigate the different aspects of the BM3D algorithm giving a sufficient introduction to the background of BM3D, to understand the basic underpinnings of the method. We also try to provide a comprehensive analysis of BM3D and its variants on various natural images, as well as describing the limitations of existing methods.

BM3D is currently known as the state of the art method for image denoising and outperforms all other algorithm when it comes to removing AWGN at a reasonable computational complexity. BM3D relies on the assumption that an image has a locally sparse representation in the transform domain. In BM3D the sparsity is enhanced by

grouping similar 2D fragments into 3D data array which the authors called "groups". Because each block in the group was chosen according to some similarity measure with respect to a reference block, the use of a higher dimensional filtering of each group was possible. This exploits the potential similarity (correlation, affinity, etc.) between grouped blocks to estimate the true signal in each of them by producing a highly sparse representation in 3D transform domain, so that the noise can be removed by wavelet shrinkage. This approach of exploiting similarity and estimating the original signal is called as *collaborative filtering*. Collaborative filtering has three successive steps:

1. For each reference patch, find similar patches from the input image by classifying them according to some similarity criteria and transform them into a 3D data array by grouping the matched 2D blocks.
2. Shrinkage of the coefficients in transformed 3D spectrum is applied to attenuate the noise.
3. Apply inverse 3D transform to the shrunken coefficients and return the obtained 2D estimates of the grouped blocks to their original positions.

In this way the collaborative filtering process gives a 3D estimation of the jointly filtered 2D blocks. As the grouped 2D blocks are similar, the transformation can achieve a very high sparse representation of the original signal. This process reveals the finest details shared by grouped blocks by preserving the unique features of each individual block.

2.3.1   Architecture of the BM3D algorithm

The BM3D algorithm can be distinctively decomposed into two almost identical steps which are illustrated in Figure 2-2.
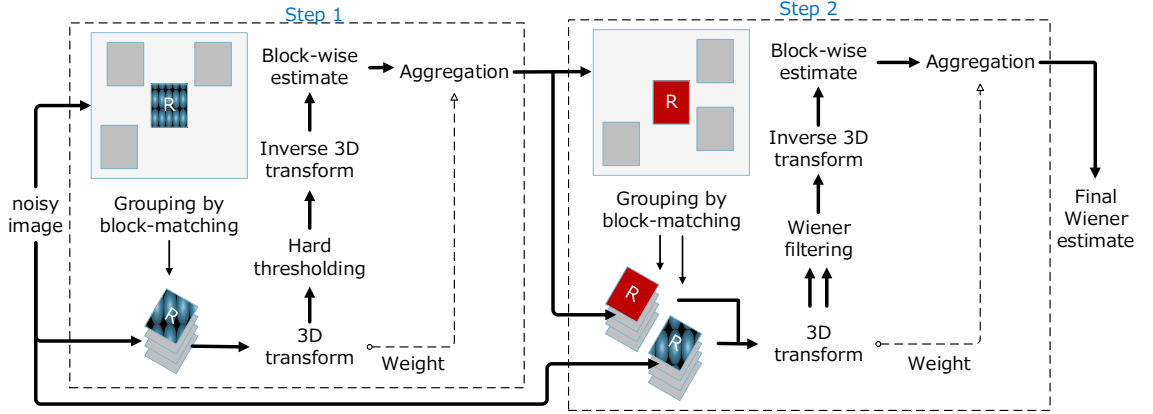
Figure 2-2: Flowchart of BM3D image denoising algorithm.

The general concept of the two steps in BM3D is as follows:

Step 1: In this step an intermediate (i.e. basic estimate) denoised image is estimated using hard thresholding during the collaborative filtering process by taking the original image (noisy image) as the input:

a) ***Grouping and collaborative filtering:*** The input image is processed block by block. These blocks are called reference blocks. For each reference block similar blocks to the currently processed one are found using a similarity measure (block matching). A 3D group (array) is then built by stacking the matched blocks and a collaborative filter is applied to the grouped blocks. In this step, hard thresholding is applied during shrinkage of the coefficients in the transform domain.

b) ***Aggregation:*** After collaborative filtering, we get an estimate of each block and a variable number of estimates for each pixel due to the overlapping of blocks. The output of the first step is obtained by weighted averaging of all the achieved block-wise estimates that have overlapped.

Step 2: This step produces the final denoised estimate based on both the input noisy image and the basic estimate obtained from step 1. Here, instead of hard thresholding,

Wiener filtering [35] is used as the shrinkage method. It applies Wiener filtering to the original noisy input image by using the basic estimate obtained from step 1 as an oracle:

a) *Grouping and collaborative filtering:* The basic estimate is used to find locations of the matched blocks similar to the currently processed one. Two 3D groups (arrays) are formed from the locations of the basic estimate, one from the original input (noisy) image and the other from the basic estimate. Then, collaborative filtering is applied on the noisy image. Here, a 3D transform is applied on both groups and during shrinkage, Wiener filtering is applied on the noisy image using the energy spectrum of the basic image as the true energy spectrum.

b) *Aggregation:* By aggregating all the estimates using a weighted average the final denoised image is obtained.

2.3.2   Applications of BM3D

Though the major application of BM3D is image denoising, it is shown to be equally effective by Dabov *et al.* [11][12] in different areas of image processing such as generic image and video restoration, image deblurring [7][8], image sharpening [9][10] and video denoising [13][14][15]. In this section we address different applications of BM3D, its limitations, and published extensions.

2.3.2.1   BM3D in medical imaging

BM3D approach is used to reconstruct medical imagery. Significant denoising performance in clinical MRI image denoising has been attained by optimizing the cost functions for noise removal [16]. ART-BM3D [17] is applied to limited-angle reconstruction in CT reconstruction which uses BM3D to find a sparse representation of the original image. Kang *et al.* [18] applied adaptive BM3D by estimating the noise using a wavelet based noise estimation technique [19] for low-radiation dose coronary CT angiography images. Other similar efforts [20][21][22][23][24][25] of medical image denoising by using BM3D are proposed by various authors.

2.3.2.2   Application in removing other kinds of noise than AWGN

BM3D is also proven to be effective for removing other kinds of noise than AWGN. In the PIDD-BM3D algorithm Danielyan *et al.* [26] used data adaptive BM3D-frames and formulated image reconstruction as a generalized Nash equilibrium problem to remove Poisson noise from images. Begovic *et al.* [27] proposed two separate contrast enhancement and denoising framework based on two popular techniques, K-SVD and BM3D for solar images corrupted with a mixture of pixel-dependent Poisson noise and white Gaussian noise. BM3D is also successfully tailored to remove power-law noise [28] and correlated noise from images [29].

2.3.3   Limitations and enhancements of BM3D

BM3D has generated much interest in the field of image and video denoising since it was published in 2007, followed by several improvements proposed by Dabov *et al.*, quickly followed by different variations, enhancements and applications of BM3D by the image processing community. Dabov *et al.* [3] exploited both locally adaptive anisotropic estimation and non-local image modeling, which only improves the denoising performance when the noise level is low but for higher noise level its performance is inferior to BM3D [1]. Dabov *et al.* also exploited local shape-adaptive anisotropic estimation, *principal component analysis* (PCA) and nonlocal image modeling [4] which preserves image details and introduces few artifacts than [1], but only shown to be effective up to noise level of standard deviation 35.

BM3D performs best when the number of matched blocks is higher (e.g., textures, regular shaped image structures, or uniform areas). It is not effective when a large amount of matching blocks is not found. As the content of natural images is not always highly correlated and large numbers of matching blocks are not always guaranteed in natural images, BM3D introduces artifacts and doesn't perform as well. Also, if the image is highly contaminated by noise, the image features becomes inseparable from the

noise itself. In these situations, though it preserves the fine image details after denoising, it can blur the edges after the collaborative filtering and aggregation step. In the transform domain the edges and the noises are not distinguishable and this forces the filtering process to remove some of the edge information making edges blurry. Several enhancements were proposed to overcome these shortcomings of BM3D [1]. The shortcomings of BM3D and efforts made to mitigate these issues are described below.

2.3.3.1 Performance decreases if the noise level is high and artifact appears:

Though BM3D introduces fewer artifacts than other denoising methods it still exhibits a few artifacts under certain conditions. A number of periodic artifacts appears when 2D-*Bior1.5* (bi-orthogonal spline wavelet with vanishing moments of the decomposing and the reconstructing wavelet functions of 1 and 5, respectively) is replaced by 2D-DCT for high noise level ($\sigma > 40$) and the block size (N_1^{ht}) is increased to 12 from 8 in the first step of BM3D [1]. Hou *et al.* argued [36] that these changes are unnecessary and proposed a scheme which gives better *peak signal to noise ratio* (PSNR) and also superior subjective visual quality than [1]. The authors simply changed the DCT transform to Bior1.5, increasing the maximum number of matched blocks (N_2^{ht}) by increasing the maximum threshold (τ_{match}^{ht}) while matching. Hou *et al.* [32] separated the 3D transform of BM3D to two steps 1D transform, which also shows better denoising performance than original BM3D algorithm for greater noise levels in terms of peak signal-to-noise ratio, structural similarity and subjective visual quality, and introduces lesser periodic artifacts. BM3D is more effective when it finds a sufficient number of matches for the reference patch. Separating edges from noise becomes increasingly difficult as the level of noise increases which results in poor matching. The calculation of each pixel values by weighted averaging of block-wise estimates also leads to blurring of edges as averaging works like a low pass filter. These issues are addressed by Chen and Wu [6] and they proposed a bounded BM3D approach which partitions the image in several regions before block matching. In addition, the authors used partial block matching instead of the block as a whole if the block contains edges and it belongs to more than one region for each coherent segment. Another approach that tries to improve

performance, when the noise standard deviation reaches 40, is proposed by Li *et al.* [33], they combine BM3D with Tetrolet prefiltering [92]. Here the Tetrolet transform is applied to the initial noisy image to remove part of the noise before executing BM3D. Though this approach shows slightly lower denoising performance than BM3D-SAPCA [4], its execution time is significantly less than the later one.

2.3.3.2 The values of the parameters in BM3D are fixed for all images:

The values of all the parameters are given *a priori,* irrespective of the type of image and the level of noise provided as input. This feature of BM3D sometimes can lead to poor denoising performance. One important such parameter is the threshold value in the shrinkage step. Zhang *et al.* [30] used an adaptive approach that applies a weak (i.e. low) threshold for low activity (flat) blocks and a strong (i.e. high) threshold for high activity (detailed) blocks based on *Structural SIMilarity* (SSIM) between similar blocks. The authors claimed that this method gives comparable performance for low noise levels but shows better denoising performance for high noise levels. BM3D relies on the sparsity of the image in the transform domain. Sparsity of the blocks is not achievable when the image contains fine details, curved edges or singularities. Sparsity is also hard to achieve when the level of noise is elevated. To overcome this difficulty, Poderico *et al.* [31] proposed a modified BM3D approach that works on the shrinkage parameter when enough sparsity is not achievable. They proposed a more agile and smoother shrinkage based on Smooth Sigmoid-Based Shrinkage, resulting in better denoising performance especially for large noise levels [37]. Mittal *et al.* [38] showed that a blind parameter estimation for BM3D based on natural scene statistics, which maps statistical features to noise variance using support vector machine regressor exhibits better denoising performance than the reference BM3D implementation in [1], measured using the *Multi-scale Structural Similarity Index* (MS-SSIM).

Chapter 3

3 Methodology

BM3D is one of the most powerful and efficient image denoising algorithms available currently. In recent years, it has gain much interest within the image processing community due to its improved performance, both in terms of reducing noise from images and in terms of maintaining visual image details. It takes advantage of a locally sparse representation of images in the transform domain. In BM3D, highly sparse representation of images in the transform domain is achieved through grouping and collaborative filtering followed by a weighted averaging. Enhancement of sparsity is achieved by finding and grouping similar 2D image blocks and stacking them together into 3D arrays or groups. The data exhibits a high level of correlation due to inherent similarity in a group. Besides reducing the noise, the collaborative filtering procedure reveals even the finest image details shared by grouped blocks, while preserving the unique features of each block.

Despite excellent results, BM3D has certain drawbacks and some improvement is still possible, especially for higher noise levels. BM3D produces many artifacts when the noise level is high ($\sigma > 30$). The performance of noise reduction also significantly drops as the noise level increases. When the noise level is large, block matching is not reliable any more, as blocks which are not similar to the referenced block can easily be grouped together into the 3D array, resulting in less sparser representation in transform domain. It also tends to give poor visual results when exposed to micro-textured zones in natural images. Another important drawback of BM3D is that it blurs sharp edges, as it uses a weighted averaging at the end of each step (i.e. first step and second step), which works more like a low pass filter. BM3D also reduces the contrast of images which can easily

be observed in almost all of its denoised images. The Lena image in Figure 3-1, for example, demonstrates all these shortcoming of BM3D for higher noise levels:

(a) noise free image

(b) noisy image ($\sigma = 80$)

(c) denoised by BM3D

(d) noisy image ($\sigma = 90$)

(e) denoised by BM3D

(f) noisy image ($\sigma = 100$)

(g) denoised by BM3D
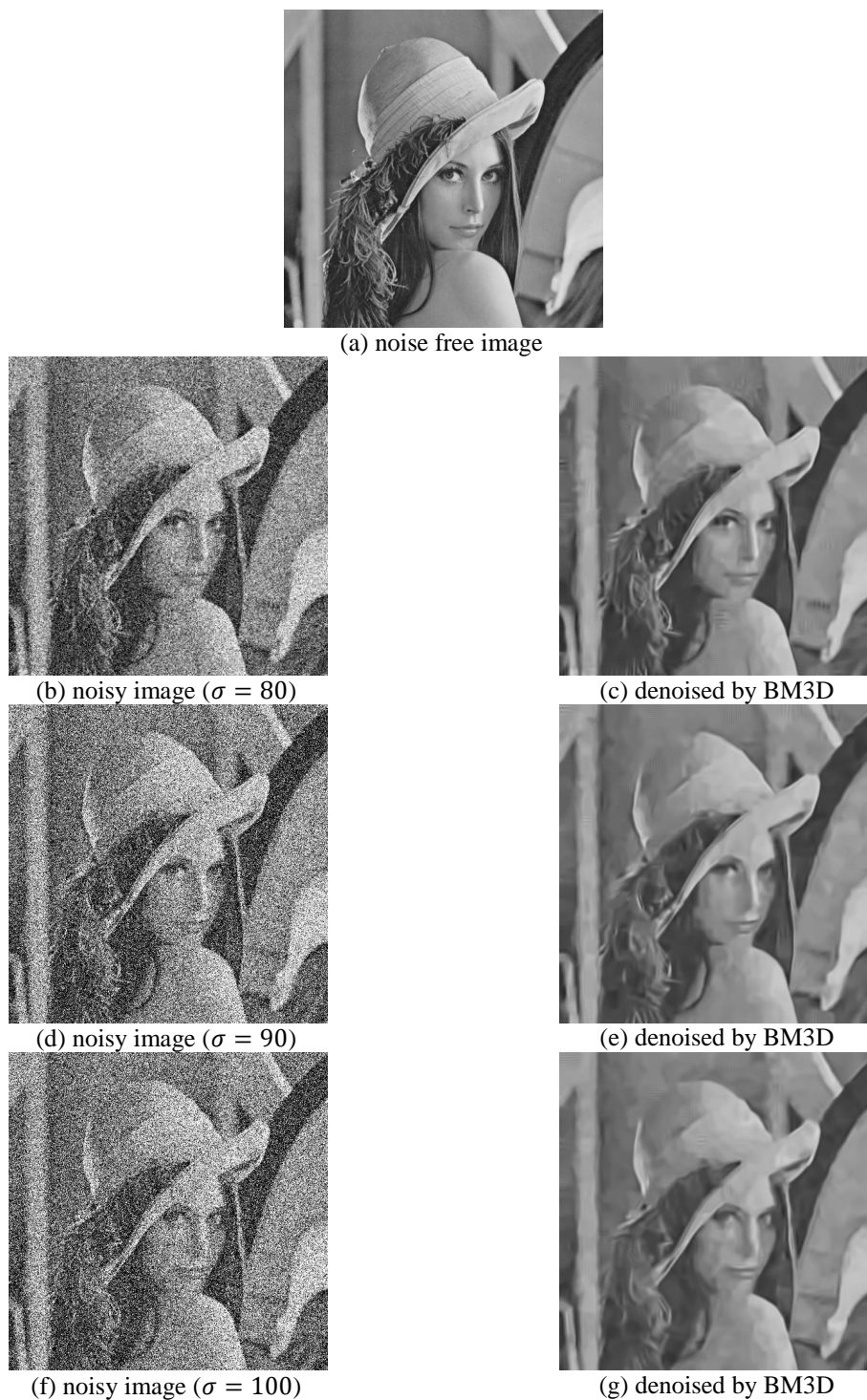
Figure 3-1: A pictorial demonstration of the major weaknesses of BM3D. (a) Noise free standard Lena image. (b), (d), and (f) noisy images with different noise levels. (c), (e), and (g) BM3D denoised image.

In this thesis, we present an enhancement over BM3D to overcome its above mentioned limitations. For higher noise levels some noisy pixel can be detected and restored based on its neighboring pixels to reduce the effect of higher noise during grouping and collaborative filtering. We used this pre-filter when the noise level is high enough($\sigma >$ 60). Also to reduce the effect of blurring sharp edges, we used an edge guided pixel estimation technique. Experimental results show that our method outperforms BM3D in terms of both objective and subjective measures. The proposed method also better preserves contrast and edges as compared to BM3D, in particular for higher noise levels.

3.1   Data adaptive BM3D with selective median filtering

Our proposed method is divided into three major steps which are illustrated in Figure 3-2. In the first step we detect the noisy pixels and try to restore these values using median filter. This pre-filtering is applied only for higher noise levels ($\sigma > 60$). If the noise level is low ($\sigma \leq 60$) or if a pixel cannot be decided as corrupted, this first step works as an identity filter. Here we assume standard deviation of noise is a priori knowledge, as it can be accurately estimated [99]. Moreover, assessing the value of sigma is out of the scope of my thesis. In second step, we group the matching blocks into a 3D data array for each reference block and apply collaborative hard thresholding for two different block sizes based on the noise level present in the input noisy image as described in Section 3.3. Unlike BM3D, here we get two preliminary basic estimates which we process to produce the final estimate. We always choose the basic estimate with block size 8×8 for determining the edge map. After that, based on the obtained edge map, we aggregate the estimation from the two different preliminary basic estimates, producing the final basic estimate. Then together with the pre-filtered noisy image we fed this more refined basic estimate as an oracle into the collaborative Wiener filtering stage which is identical to the original BM3D algorithm [1].
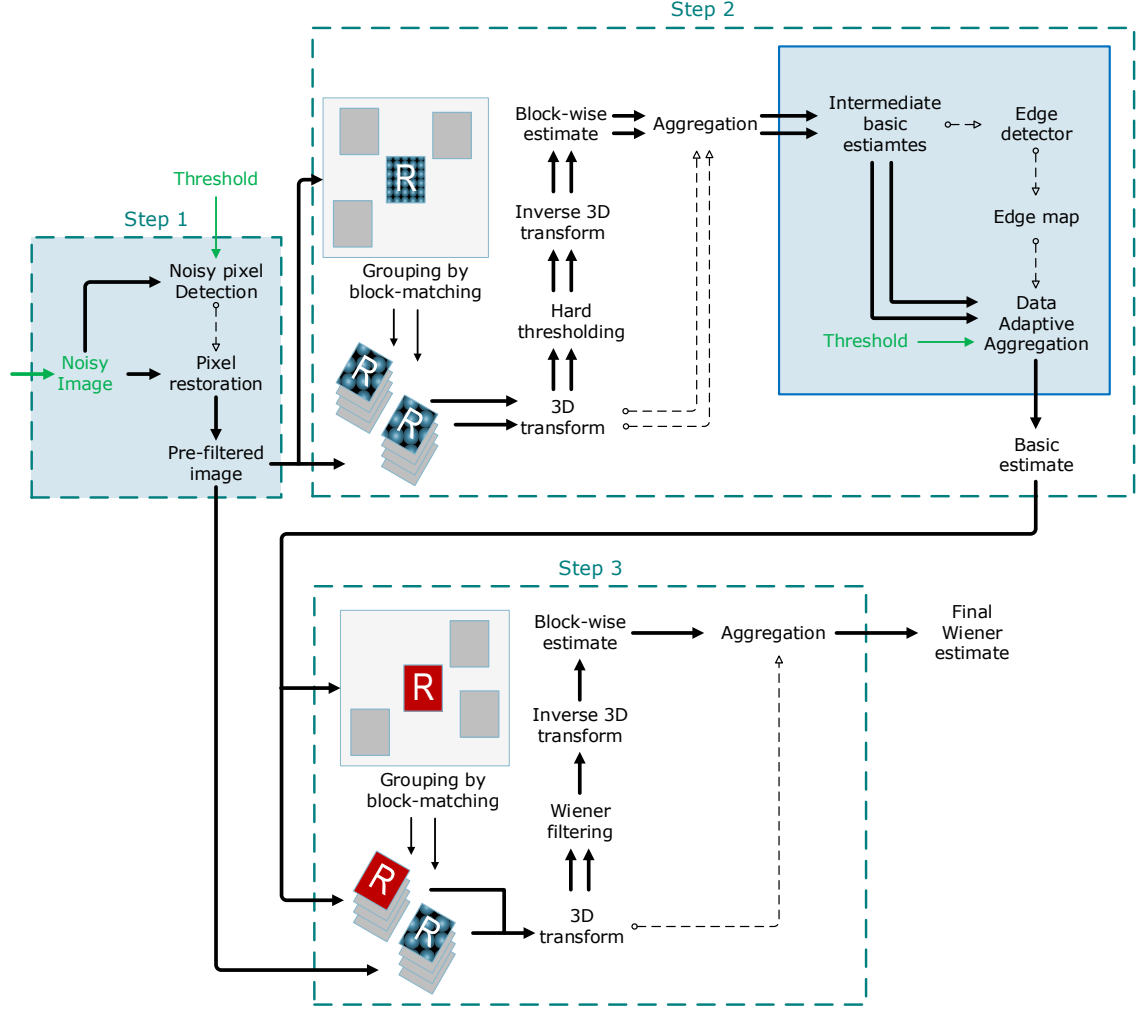
Figure 3-2: Scheme of data adaptive BM3D with selective median filtering

Let us assume an image has been corrupted by AWGN resulting in a noisy image $z: X \rightarrow \mathbb{R}$ of the form.

$$z(x) = y(x) + \eta(x), \quad x \in X, \qquad \textbf{3.1}$$

where y is the uncorrupted or noiseless image, x represents the pixels of an image in 2D spatial domain $X \subset \mathbb{Z}^2$, η is independent and identically distributed zero mean AWGN with variance σ^2.

3.1.1   First step: Selective pixel restoration with median filter

In this step we apply a simple decision-based linear filter to reduce noisy pixels from the image before applying the actual denoising method. This approach gives considerably good results for higher noise levels.

I_{-4}	I_{-3}	I_{-2}
I_{-1}	$I_0 \; or \; I_x$	I_1
I_4	I_3	I_2

Figure 3-3: Illustration of a window of size $\mathbf{3 \times 3}$ (i.e. $\boldsymbol{k = 3}$) or $\mathbf{2 \times 4 + 1}$ (i.e. $\boldsymbol{n = 4}$)

We use a sliding window method from left to right, top to bottom for each pixel location to identify the noisy pixel and then restore it subsequently. For this section, with W_x^k we denote a filter window of size $k \times k$ or $2n + 1$ taken from the noisy input image z,

$$W_x^k = (I_{-n,}, \dots, I_{-1}, I_0, I_1, \dots, I_n,), \qquad \textbf{3.2}$$

where I_0 or I_x is the center pixel of the window W_x^k under processing. For our work, we use a 3×3 filter window W_x^3 (depicted in Figure 3-3) centered at x,

$$W_x^3 = (I_{-4,}, \dots, I_{-1}, I_0, I_1, \dots, I_4,). \qquad \textbf{3.3}$$

To identify the noisy pixel we use a binary classification method to classify the pixel in consideration as *corrupted* or *not corrupted*, based on the current pixel value and its neighboring pixel values. The current pixel I_x can be classified as *corrupted* if it is significantly lower or greater than all other pixel values in the considered neighborhood. If the current pixel is a maximum or a minimum, we sort the pixel values (excluding I_x) of the filter window W_x^3 in ascending order.

The re-arranged vector is shown below

$$s(x) = (s_1(x), s_2(x), \dots, s_8(x)) , \qquad\qquad \textbf{3.4}$$

where $s_1(x), s_2(x), \dots, s_8(x)$ are the pixel values of the window W_x^3 sorted in ascending order. To classify the current pixel, we choose a simple classifier that operates on the differences of the current input pixel and maximum or minimum of the vector, $s(x)$:

$$\delta(x) = \begin{cases} 1, & \{I_x < (\min\{s(x)\} - \tau_{pre})\} \vee \{I_x \geq (\max\{s(x)\} + \tau_{pre})\} \\ 0, & otherwise \end{cases} , \qquad \textbf{3.5}$$

where τ_{pre} is a threshold value determined through empirical analysis. If the pixel is classified as *corrupted (i.e. $\delta(x) = 1$)* then we use a median filtering function on the window W_x^3 to determine the value of the pixel and replace it. Otherwise, the original pixel value is kept,

$$SMEDIAN \ f(x) = \begin{cases} \underset{q \in W_x^3, q \neq I_x}{Median}\{q\}, & if \ \delta(x) = 1 \\ I_x, & otherwise \end{cases} , \qquad \textbf{3.6}$$

where SMEDIAN refers to the selective median operation. The effect of this simple selective prefilter is considerable for noisy images with relatively higher noise levels, where the contrast and edges are best preserved with a higher peak signal-to-noise ratio (PSNR) in denoised images than the original BM3D algorithm [1].

3.1.2   Second step: data adaptive estimation of the basic denoised image

In this step we obtain the basic estimate using the pre-filtered version of the noisy image. We follow the following procedure for two different block sizes of $N_{1_{guide}}^{ht}$, and $N_{1_{est}}^{ht}$, depending on the noise level present in the examined image. Unless otherwise mentioned, for the rest of this step we will drop the subscript *guide* and *est* as the process is similar for both the case.

Let us denote the basic estimate by $\hat{y}^{basic}$ and the final estimate by $\hat{y}^{final}$. We define Z_x to denote a block of size $N1 \times N1$ extracted from the provided noisy image z. Here x in

Z_x is the coordinate of the top left corner of the patch. In other word, we can say that the patch Z_x is located at location x in the image z. A 3D group or a group of selected 2D patches is denoted by a bold face capital letter with a subscript representing the set of its grouped patches coordinates (e.g. $\mathbf{Z}_s$ represents a collection of 2D patches composed of patches Z_x located at $x \in S \in X$). As we have two near identical steps in BM3D we use the superscript *ht* (stands for hard thresholding) and *wie* (stands for Wiener filtering) to represent the parameters in the first step and the second step respectively. That is, the patch size used in first step is denoted by N_1^{ht} and the patch size used in second step is denoted by N_1^{wie}. We process the blocks in sliding-window manner by following a raster scan where each next block is taken with a fixed pixel shift from the previous one. This ensures there is at least one estimate for each image pixel. We denote the currently processed block by Z_{x_R} and call it *reference block*.

Firstly, we find the blocks that are similar to the reference one within a fixed neighborhood. We define the similarity based on the block-distance and only consider those blocks whose distance is less than a pre-defined threshold. The distance is obtained by applying a normalized 2D linear transform on both blocks followed by a hard-thresholding on the obtained coefficients.

$$d\left(Z_{x_R}, Z_x\right) = \frac{\left\|\Upsilon'\left(\mathcal{T}_{2D}^{\mathrm{ht}}\left(Z_{x_R}\right)\right) - \Upsilon'\left(\mathcal{T}_{2D}^{\mathrm{ht}}\left(Z_x\right)\right)\right\|_2^2}{\left(N_1^{\mathrm{ht}}\right)^2},$$ **3.7**

here, $d(Z_{x_R}, Z_x)$ is the distance between the two blocks in consideration. Υ' denotes the hard thresholding operator with threshold value $\lambda_{2D}\sigma$ and T_{2D}^{ht} denotes the linear transform.

By applying the above equation and defining a maximum distance τ_{2D}^{ht} to match the similar blocks we get the coordinates of a set of blocks, $S_{x_R}^{ht}$, that are similar to Z_{x_R},

$$S_{x_R}^{\mathrm{ht}} = \left\{x \in X : d\left(Z_{x_R}, Z_x\right) \leq \tau_{\mathrm{match}}^{\mathrm{ht}}\right\}.$$ **3.8**

$d(Z_{x_R}, Z_{x_R}) = 0$ as the reference block matches with itself. It implies that we will find at least one match for each reference block leaving the set $S_{x_R}^{ht}$, always non-empty. By using the found set $S_{x_R}^{ht}$ we form a 3D group $\mathbf{Z}_{S_{x_R}^{ht}}$, stacking the noisy blocks $Z_{x \in S_{x_R}^{ht}}$.

Then we perform the collaborative filtering on the formed 3D group by applying hard thresholding in 3D transform domain. This is done by first applying the 3D transform to the group, performing shrinkage of the transform coefficient by applying hard-thresholding, inverting the 3D transformed coefficients and finally returning the estimates of the blocks back to their previous positions,

$$\widehat{\mathbf{Y}}_{S_{x_R}^{ht}}^{ht} = \mathcal{T}_{3D}^{ht^{-1}} \left(\Upsilon \left(\mathcal{T}_{3D}^{ht} \left(\mathbf{Z}_{S_{x_R}^{ht}} \right) \right) \right),$$
$$\text{3.9}$$

here, Υ represents the hard thresholding operator with a threshold value of $\lambda_{3D}\sigma$. The 3D array $\widehat{\mathbf{Y}}_{S_{x_R}^{ht}}^{ht}$ is made of the block-wise estimates $\hat{Y}_x^{ht,x_R}$ located at the matched coordinates $x \in S_{x_R}^{ht}$. In $\hat{Y}_x^{ht,x_R}$ the subscript x represents the location or the coordinates of this block-estimate and the superscript x_R refers to the reference block.

As the block-wise estimates $\hat{Y}_{x \in S_{x_R}^{ht}}^{ht,x_R}$ of pixels overlap, we obtain a overcomplete representation of the true image. A block can be matched and grouped with multiple reference blocks. For example, $\hat{Y}_{x_q}^{ht,x_p}$ and $\hat{Y}_{x_p}^{ht,x_q}$ are both located at x_p and x_q but matched during the processing of reference blocks located at x_p and x_q respectively. This process produces a substantially overcomplete representation of the true image if there is plenty of overlapping block-wise estimates. To get basic estimate at the end of the first stage, we perform aggregation on the corresponding block-wise estimates $\hat{Y}_{x \in S_{x_R}^{ht}}^{ht,x_R}$ by applying a weighted averaging. The aggregation of block-wise estimates is a critical step which relies on the proper selection of aggregation weights. Inspired by [76] and [90], the authors of BM3D proposed aggregation weights that are inversely proportional to the total sample variance of the corresponding block-wise estimates. That is, the more dissimilar or noisy the block-wise estimate is the less the weight is. This ensures higher contribution from the similar block-wise estimates in the final estimation of the basic

image. The following weight is applied to the group of the overlapping block-wise estimates $\hat{Y}_{x \in S_{x_R}^{ht}}^{ht,x_R}$ for each $x_R \in X$,

$$w_{x_R}^{ht} = \begin{cases} \frac{1}{\sigma^2 N_{har}^{x_R}}, & \text{if } N_{har}^{x_R} \geq 1 \\ 1, & \text{otherwise} \end{cases} \tag{3.10}$$

where σ is the variance and $N_{har}^{x_R}$ denotes the number of retained coefficients after the hard thresholding.

Finally, we compute the preliminary basic estimate $\hat{y}^{basic}$ by a weighted average of the block-wise estimates $\hat{Y}_{x \in S_{x_R}^{ht}}^{ht,x_R}$ located at the same place, using the weights from Equation 3.10,

$$\hat{y}^{basic}(x) = \frac{\sum_{x_R \in X} \sum_{x_m \in S_{x_R}^{ht}} w_{x_R}^{ht} \hat{Y}_{x_m}^{ht,x_R}(x)}{\sum_{x_R \in X} \sum_{x_m \in S_{x_R}^{ht}} w_{x_R}^{ht} \chi_{x_m}(x)}, \forall x \in X, \tag{3.11}$$

where $\chi_{x_m} : X \to \{0,1\}$ is the characteristic function of the square support of a block located at $x_m \in X$.

At this stage, unlike BM3D, which computes final basic estimate $\hat{y}^{basic}$ by a weighted average of the block-wise estimates $\hat{Y}_{x \in S_{x_R}^{ht}}^{ht,x_R}$ located at the same place, using the weights from Equation 3.10, we compute two preliminary estimates (i.e. $\hat{y}^{prel}$) namely, $\hat{y}^{guide}$ and $\hat{y}^{est}$ for the two different block size. We apply the Canny edge detector [91] on the preliminary basic estimation $\hat{y}^{guide}$ to construct a edge map. The Canny edge detector highlights regions with high spatial derivatives by applying a Gaussian smoothing with a kernel size $k_{gaussian}$ to reduce the level of noise in the input image, followed by the processing of derivative in both x and y direction to get the magnitude and direction of gradient,

$$D = \sqrt{D_x^2(x,y) + D_y^2(x,y)}, \tag{3.12}$$

and
$$\theta = \arctan\left(\frac{D_x(x,y)}{D_y(x,y)}\right),$$
3.13

where D is the magnitude of gradient and θ is angle of gradient. Given the values of the gradient, the algorithm applies non-maximum suppression, which suppresses any pixel that is not at the maximum. This is done by preserving all local maxima in the gradient image, and deleting everything else. Furthermore, the gradient array is reduced by hysteresis that uses two thresholds, τ_{low} and τ_{high}, determined empirically. Pixels with gradient magnitude $D < \tau_{low}$ are discarded, while pixels with gradient magnitude $D > \tau_{high}$ are kept as edges. If the magnitude is between the thresholds (i.e. $\tau_{low} < D < \tau_{high}$), it is only kept as an edge if and only if there is a path from this pixel to a pixel with gradient $D > \tau_{high}$. Once we get the edge map by applying the Canny, we binarize the array values,

$$\beta_{\hat{y}^{guide}} = \zeta\left(C_{\tau_{low},\tau_{high}}^{k_{gaussian}}\left(\hat{y}^{guide}\right)\right),$$
3.14

where, $C_{\tau_{low},\tau_{high}}^{k_{gaussian}}$ represents the Canny edge detection process and ζ represents the binarization method.

The obtained edge guide $\beta_{\hat{y}^{guide}}$ is then used to combine the values from the two preliminary basic estimates $\hat{y}^{guide}$ and $\hat{y}^{est}$ to form the final basic estimate $\hat{y}^{basic}$. This is done by processing each pixel in a raster scan fashion and obtaining the edge strength for that location based on its neighboring pixels. For this processing, with $B_x^{k^{es}}$ we denote a block of size $k^{es} \times k^{es}$ taken from $\beta_{\hat{y}^{guide}}$, where x is the center of the block; and with I_x we denote a pixel at location x. The edge strength $\gamma_{k^{es}}$ of a $k^{es} \times k^{es}$ block is defined as the number of pixels indicating edge over the total number of pixels,

$$\gamma_k = \frac{\sum_{x \in B_x^{k^{es}}}(I_x\,[I_x \neq 0])}{\left|B_x^{k^{es}}\right|}.$$
3.15

Here, $\left|B_x^{k^{es}}\right|$ denotes the cardinality of block $B_x^{k^{es}}$.

Now we select the estimation for our current pixel based on the calculated edge strength $\gamma_{k^{es}}$. We define a threshold $\tau^{ht}_{combine}$ and if the edge strength $\gamma_{k^{es}}$ is greater than this threshold value (i.e. $\gamma_{k^{es}} > \tau^{ht}_{combine}$) we call this block, $B_x^{k^{es}}$, a block with edges or edged block, otherwise (i.e. $\gamma_{k^{es}} \leq \tau^{ht}_{combine}$), we call this a smooth block.

Once the edge strength in the block is calculated for the current pixel of interest we get the value for each pixel of the final basic estimate by the following equation:

$$\hat{y}^{basic}(I_x) = \begin{cases} \hat{y}^{guide}(I_x) & , \; \gamma_{k^{es}} \leq \tau^{ht}_{combine} \\ \hat{y}^{est}(I_x) & , \; \gamma_{k^{es}} > \tau^{ht}_{combine} \end{cases} . \qquad \textbf{3.16}$$

Where, the threshold $\tau^{ht}_{combine}$ is defined empirically, which we will discuss later in Section 3.2. In the above equation, what we basically do is select the estimation for each pixel from different block size based on the edge activity present in its defined neighborhood $B_x^{k^{es}}$.

The obtained basic estimate $\hat{y}^{basic}$, together with the pre-filtered noisy image, is used in the third step to process the final denoised image.

3.1.3   Third step: estimating the true image from the basic estimate and the noisy image

The third and final step is similar to the original BM3D algorithm except instead of original noisy image we use the pre-filtered image as the input to this step. In this step, both the basic estimate $\hat{y}^{basic}$ and the pre-filtered noisy image are used to improve the denoising. We take advantage of the basic estimate by processing the grouping within the basic estimate and using collaborative Wiener filtering. The significant reduction of noise in the basic estimate allows us to use the normalized ℓ^2-distance for grouping the similar blocks for reference blocks

$$d^{ideal}(Z_{x_R}, Z_x) = \frac{\|Y_{x_R} - Y_x\|_2^2}{\left(N_1^{ht}\right)^2}, \qquad \textbf{3.17}$$

here, $\|.\|_2$ indicates the ℓ^2-norm. By applying the distance measure from the above equation we get the set of matched blocks for each reference block

$$S_{x_R}^{\text{wie}} = \left\{ x \in X : \frac{\left\| \widehat{Y}_{x_R}^{\text{basic}} - \widehat{Y}_x^{\text{basic}} \right\|_2^2}{\left(N_1^{\text{wie}}\right)^2} < \tau_{\text{match}}^{\text{wie}} \right\}.$$

3.18

By using the coordinates from the set $S_{x_R}^{wie}$, two groups are $\widehat{\mathbf{Y}}_{S_{x_R}^{wie}}^{basic}$ and $\mathbf{Z}_{S_{x_R}^{wie}}$ are formed from the basic estimate and the noisy image respectively.

The 3D transform domain collaborative Wiener filtering of $\mathbf{Z}_{S_{x_R}^{wie}}$ is done by element by element multiplication of the Wiener shrinkage coefficients $\mathbf{W}_{S_{x_R}^{wie}}$ and the 3D transform domain coefficients of the noisy data $T_{3D}^{wie}\left(\mathbf{Z}_{S_{x_R}^{wie}}\right)$,

$$\mathbf{W}_{S_{x_R}^{\text{wie}}} = \frac{\left| T_{3D}^{\text{wie}}\left(\widehat{\mathbf{Y}}_{S_{x_R}^{\text{basic}}}^{\text{basic}}\right)\right|^2}{\left| T_{3D}^{\text{wie}}\left(\widehat{\mathbf{Y}}_{S_{x_R}^{\text{basic}}}^{\text{basic}}\right)\right|^2 + \sigma^2}$$

3.19

$$\widehat{\mathbf{Y}}_{S_{x_R}^{\text{wie}}}^{wie} = T_{3D}^{\text{wie}^{-1}}\left(\mathbf{W}_{S_{x_R}^{\text{wie}}} T_{3D}^{\text{wie}}\left(\mathbf{Z}_{S_{x_R}^{\text{wie}}}\right)\right).$$

3.20

Here, the 3D array $\widehat{\mathbf{Y}}_{S_{x_R}^{wie}}^{wie}$ is made of the block-wise estimates $\hat{Y}_x^{wie,x_R}, \forall_x \in S_{x_R}^{wie}$. The aggregation of the block-wise estimate for the global final estimate $\hat{y}^{final}$ is done same way as the first step except the aggregation weights, which are obtained by the following equations,

$$w_{x_R}^{\text{wie}} = \sigma^{-2}\left\| \mathbf{W}_{S_{x_R}^{\text{wie}}} \right\|_2^{-2},$$

3.21

$$\widehat{y}^{\text{final}}\left(x\right) = \frac{\displaystyle\sum_{x_R \in X} \sum_{x_m \in S_{x_R}^{\text{wie}}} w_{x_R}^{\text{wie}} \widehat{Y}_{x_m}^{\text{wie},x_R}(x)}{\displaystyle\sum_{x_R \in X} \sum_{x_m \in S_{x_R}^{\text{wie}}} w_{x_R}^{\text{wie}} X_{x_m}\left(x\right)}, \forall x \in X. \qquad \textbf{3.22}$$

3.2   Different schemes and selection of parameters

We have experimented with various variations of our new proposed method. They vary mainly in terms of the values of different parameters and the selection of pre-filter depending on the level of noise present in the noisy input image. Let us review some parameters that we have introduced in the previous two sections in addition to the parameters introduced in the original BM3D article [1]:

- τ_{pre}: threshold used for the selection of corrupted pixel in the selective pixel restoration step.
- $k_{gaussian}$: size of the Gaussian kernel used for Canny edge detector in second step of our algorithm.
- τ_{low} and τ_{high}: two thresholds used during hysteresis in Canny edge detector.
- $\tau_{combine}^{ht}$: a threshold value for edge strength, used for determining the patch size for higher noise levels.
- k^{es}: patch size used in data adaptive estimation phase.

We examined the proposed method using different set of values for these parameters. For example we have studied our algorithm for different values and the combination of the following parameters:

- τ_{pre}: 20, 30, 40, 50, and 60.
- $k_{gaussian}$: 3, 5, and 7.
- τ_{low}: 20, 25, 30, and 35 for each $k_{gaussian}$.
- τ_{high}: 50, 75, 100, 125 and 150 for each τ_{low}.
- $\tau_{combine}^{ht}$: 0.1, 0.2, 0.3 0.4 and for point operation (i.e. $k^{es} = 1$) we used 1.
- k^{es}: 1, 3, 5, and 7.

We call the method combining each set of these parameters a *scheme*. We examined a substantial amount of *schemes,* but here we will present a few of them as the other *schemes* do not produce comparable results with respect to the *schemes* presented here. Later we will combine all these different *schemes* to present our final proposed method.

We empirically determined best values for the parameters (i.e. τ_{low}, τ_{high}, and $k_{gaussian}$) used in the Canny edge detector and used it for all our experiments. We used minimum Gaussian smoothing by setting $k_{gaussian} = 3$, as our input image to the Canny edge detector is already smoothed by its previous steps. The value of τ_{low} and τ_{high} are determined to be 25 and 75 respectively by empirical analysis. For simplicity of our discussion, we introduce two new parameter, N_1^{smooth} and N_1^{edged} , which are nothing but the block sizes taken for the smooth block and edged block based on the edge activity discussed at the end of section 3.1.2. The following table shows the selected different *schemes* that we have studied:

Table 3-1: Parameters used for different schemes

Scheme	σ	k^{es}	$\tau_{combine}^{ht}$	N_1^{smooth}	N_1^{edged}	τ_{pre}
SCHEME1	$\sigma < 10$	1	1	4	8	*N/A*
	$10 \leq \sigma$	5	0.2	8	12	
SCHEME2	$\sigma < 10$	1	1	8	4	*N/A*
	$10 \leq \sigma$	5	0.2	12	8	
SCHEME3	$\sigma < 10$	1	1	8	12	*N/A*
	$10 \leq \sigma$	5	0.3	8	12	
SCHEME4	$\sigma < 10$	1	1	12	8	*N/A*
	$10 \leq \sigma$	5	0.3	12	8	
SCHEME3-1	$\sigma < 10$	1	1	8	12	30
	$10 \leq \sigma$	5	0.3	8	12	
SCHEME3-2	$\sigma < 10$	1	1	8	12	40
	$10 \leq \sigma$	5	0.3	8	12	
SCHEME3-3	$\sigma < 10$	1	1	8	12	50
	$10 \leq \sigma$	5	0.3	8	12	

The *N/A* in the above table means no pre-filtering is used for those schemes. *SCHEME3-1, SCHEME3-2, and SCHEME3-3* are the pre-filtered extensions of *SCHEME3,* specifically designed for higher noise levels. Note again, Table 3-1 only presents the schemes that we are interested in as they show relatively better image denoising; the others have been discarded.

Unless otherwise mentioned, for all other parameters we have used the default values suggested by the original BM3D article [1] for all the above mentioned *schemes*, which are given below:

Table 3-2: Value of Parameters inherited from BM3D

	$\sigma \leq 40$	$\sigma > 40$
N_1^{wien}	8	12
N_2^{ht}	16	16
N_2^{wien}	32	32
τ_{match}^{ht}	2500	5000
τ_{match}^{wie}	400	3500
λ_{3D}	2.7	2.7
N_{step}^{ht}	3	3
N_{step}^{wie}	3	3
T_{2D}^{ht}	DCT	DCT
T_{2D}^{wie}	DCT	DCT

The set of parameters mentioned in Table 3-1 table is:

- N_1^{wien}: patch size used in the collaborative Wiener filtering step.

- N_2^{ht} and N_2^{wien}: are the maximum number of similar patches kept during the hard thresholding step and collaborative Wiener filtering step.

- τ_{match}^{ht} and τ_{match}^{wie}: maximum thresholds for the distance between two similar blocks.

- λ_{3D}: threshold for the coefficient of hard thresholding stage.

- N_{step}^{ht} and N_{step}^{wie}: rather than sliding by one pixel to every next reference block, to speed up the processing, step size N_{step}^{ht} and N_{step}^{ht} is used in the hard thresholding and Wiener filtering stage respectively.

- T_{2D}^{ht} and T_{2D}^{wie}: transforms applied in the hard thresholding step and Wiener filtering step respectively.

One notable change in table Table 3-1 from the original article is the choice of DCT as T_{2D}^{ht} for noise level. It is well defined from the original BM3D article and several other articles that this choice of T_{2D}^{ht} between different transforms (e.g. DCT, Bior1.5) does not have significant impact on the denoising results, which has been shown experimentally in [1]. Choosing DCT instead of Bior1.5 lets us to experiment with block sizes which are not a power of 2 (Bior1.5 transform requires the block sizes to be a power of 2). This allowed us to work with a patch size of 12 instead of 16 for higher noise levels, which significantly reduced the computation time.

3.3 Selected parameters for our proposed method

Experimental results demonstrate that all these *schemes* achieves significantly better results compared to BM3D in terms of both objective and subjective qualities at certain noise level. Each *scheme* has its own strength and weakness due the inherent structure of those *schemes* and outperforms BM3D at certain noise levels. For example *SCHEME1* and *SCHEME3* perform better when noise level is less than or equal to 60 (i.e. $\sigma \leq 60$), while *SCHEME3-1* and *SCHEME3-3* outperforms BM3D when noise level is

relatively high ($\sigma > 60$). Based on the experimental results, our final proposed method AEG-BM3D is governed by the following parameters and their mentioned values:

Table 3-3: Parameter values for our proposed method

σ	k^{es}	$\tau^{ht}_{combine}$	N_1^{smooth}	N_1^{edged}	τ_{pre}
$\sigma < 10$	1	1	4	8	X
$10 \leq \sigma \leq 60$	5	0.3	8	12	X
$\sigma > 60$	5	0.3	8	12	50

We have used 32 random images to perform our denoising experiment, including some popular standard images used in image denoising literatures. Our results show that our proposed method significantly outperforms BM3D in terms of both subjective and objective qualities. Our method also preserves contrast and edges well as compared to BM3D [1] for higher noise levels.

Chapter 4

4 Experimental Results and Analysis

In this chapter we will discuss the details of our experiments and the obtained results. Before evaluating and analyzing our proposed algorithm, we also present a brief introduction to the dataset used and performance measures employed during experimentation.

In this result chapter, we compare our proposed extension of BM3D only with the original BM3D algorithm [1]. We found it superfluous to compare our results with other existing methods as the superiority of BM3D for AWGN has already been established by several scholarly articles [94] [95]. In these articles, the authors studied the performance of various current state of the art image denoising methods and demonstrated that BM3D is among the best performing algorithm considered therein.

We used the implementation found in Image Processing Online (IPOL) [95] to reproduce and extend the original BM3D algorithm. The code was implemented in C. We also used OpenCV for some part of our implementation and Matlab to obtain the SSIM values.

4.1 Dataset and performance measures

4.1.1 Dataset

We used 32 different images to quantify the performance of the proposed algorithm. Our dataset includes standard test images along with few other selected images from the CMU image database [96] and one synthetic image. For our discussion, we have roughly classified these images into *less textured* images (i.e. images with relatively more smooth

area or details) and *heavily textured* images (i.e. images with high textures). Figure 4-1 and Figure 4-2 shows the set of less textured and heavily textured images. For the sake of simplicity, out of 16 less textured and 16 heavily textured images we have selected 5 from each of them to compare our denoising performance. In this chapter, for less textured images we will evaluate the data for Lena, Man, Boat, Cameraman, Couple and Hill, while for heavily textured images we will evaluate data for Mandril, Barbara, Fingerprint, House, and Shoe.

(a) Lena	(b) Man	(c) Boat	(d) Cameraman (256x256)
(e) Couple	(f) Hill	(g) Gray Shades	(h) Castle
(i) Field	(j) Hill 2	(k) Pine	(l) Town
(m) Pingpong	(n) Building 01	(o) Bush	(p) Building 02

Figure 4-1: Set of less textured noise-free images used for the comparison tests.

(a) Mandril (b) Barbara (c) Fingerprint (d) House (256x256)

(e) Shoe (f) Golf (g) Grid (h) Hotel

(i) Tiger (j) Train (k) Artichoke (l) Lab

(m) Building 03 (n) Corner (o) Texture 01 (p) Texture 02
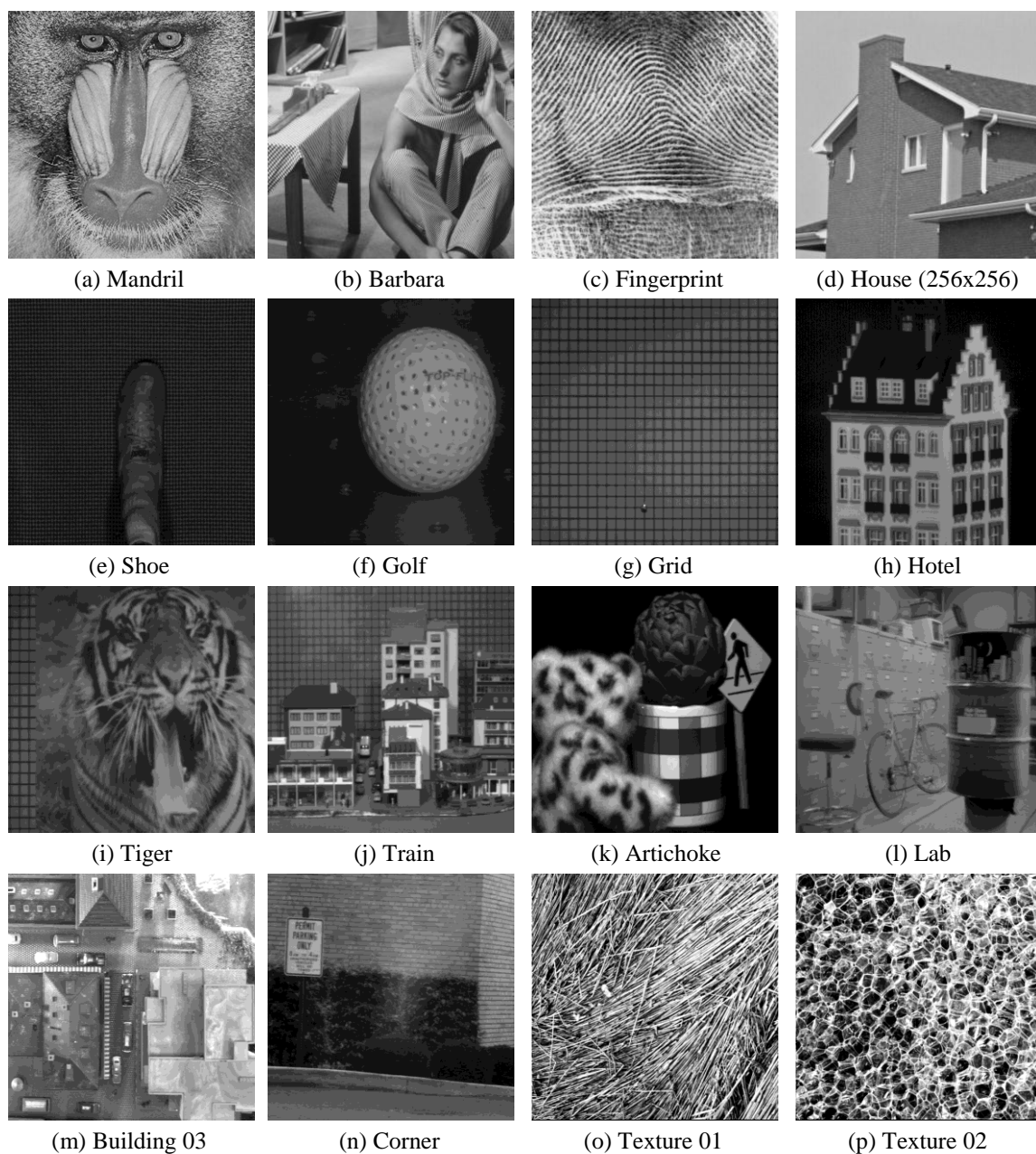
Figure 4-2: Set of heavily textured noise-free images used for the comparison tests.

The results presented in this thesis are obtained by adding simulated AWGN to true noiseless images. After denoising the results are compared with the true noiseless image for performance evaluation. We analyzed our images with noise levels (i.e. σ) 2, 5, 10,

20, 30, 40, 60, 80, 90 and 100. Results presented here are acquired on exactly the same noisy image for both the BM3D and our proposed method.

All of our images are 8-bit gray scale images and of dimension 512×512, except House and Cameraman images, which have 256×256 pixels.

4.1.2   Performance Measures

There is no universally accepted image denoising quality measurement. Therefore, we compare our denoised images both objectively and subjectively. For objective measurement or numerical measurement we present the widely used PSNR and SSIM for each observed image, as there is no precise rule to select one measure over the other. Evaluation of the subjective quality is done by comparing fragments of images side by side. In visual evaluation we particularly compare sharpness of edges and preservation of contrast. We also present intensity profiles of the true image along with BM3D and our proposed method to check the preservation of edges and denoising quality.

4.1.2.1   Peak Signal to Noise Ratio (PSNR)

Peak Signal to Noise Ratio (PSNR) is the most widely used quality measurement metric for image denoising based on Mean Squared Error (MSE). Larger PSNR values indicate better signal restoration. PSNR does not depend on any visual analysis. The PSNR in decibels (dB) is defined as follows:

$$MSE = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} (f_{xy} - \hat{f}_{xy})^2 \ , \qquad\qquad \textbf{4.1}$$

$$PSNR = 10 \log_{10}(\frac{255^2}{MSE}) \ . \qquad\qquad \textbf{4.2}$$

The Mean Squared Error is basically defined as the squared Euclidean distance of true noiseless image and resultant (i.e. denoised or output) image pixels. If the distance between them is huge, the MSE is larger and therefore PSNR is lower. Again, if the

distance between two pixels of same coordinate is small, MSE is small and hence the PSNR is higher. In brief, if PSNR is high, it indicates that the reconstructed image is close to the original image or more noise is reduced. Most denoising literature dealing with AWGN use only PSNR for performance comparison, as it is more sensitive to additive Gaussian noise than other quality measures such as SSIM. In this study, our key goal is to improve PSNR for most noise levels, in particular for higher noise levels.

4.1.2.2   Structural Similarity Index (SSIM)

SSIM was introduced by Wang *et al.* [97], which provide higher values when two images are structurally similar. It means that an acceptable value of SSIM ensures the reconstructed image (denoised image in our case) is visually similar to the original one. In other words, we can say, if we get an acceptable SSIM value, we can say that our denoised image is not changed as a result of applying our proposed mechanism. The following equation defines SSIM,

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)},$$
  4.3

where x and y are two windows of common size (e.g. $N \times N$), μ_x and μ_y is the average of x and y, respectively. σ_x and σ_y are the standard deviation of x and y, and σ_{xy} is the co-variance of x and y. c_1 and c_2 are two variables to stabilize the division with weak denominator and are defined as follows,

$$c_1 = (K_1 L)^2,$$
  4.4

and
$$c_2 = (K_2 L)^2,$$
  4.5

where L is the dynamic range of the pixels and $K1 << 1$, $K2 << 1$ (0.01 and 0.03 respectively by default). The covariance σ_{xy} is defined as follows,

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y).$$
  4.6

4.2 Experimental results and analysis

4.2.1 Signal Restoration

In this section we compare the signal restoration capability of our proposed method to that of BM3D. First we analyze our experimental results for less textured images and then we do the same for heavily textured images. Finally, we will give a summary of our obtained results for PSNR at the end of this section. In this chapter, we will present and discuss our result for few selected *schemes* (namely, *SCHEME1*, *SCHEME3*, *SCHEME3-1*, and *SCHEME3-3*) from the ones that we mentioned in Table 3-1. We have selected these schemes empirically based on their performance for the ease of presentation of experimental results.

In terms of PSNR, our method demonstrates the best performance and outperforms BM3D in almost all noise levels for less textured images. Let us start by an example for the image Lena. Table 4-1 summarizes the PSNR obtained for this image for four selected schemes or variations of our experimental methods, final proposed method (i.e. AEG-BM3D) and original BM3D.

Table 4-1: PSNR (dB) comparison for Lena image among BM3D, our four experimental schemes and final proposed method for different noise levels

σ	SCHEME1	SCHEME3	SCHEME3-1	SCHEME3-3	AEG-BM3D	BM3D
2	**43.5495**	43.5388	43.5058	**43.5495**	**43.5495**	43.5389
5	**38.8128**	**38.728**	**38.7966**	**38.8063**	**38.8128**	38.7264
10	35.883	**35.885**	35.6882	35.6971	**35.885**	35.8841
20	**33.0003**	**33.0008**	32.8148	32.9886	**33.0008**	32.9968
30	**31.179**	**31.1758**	30.7269	31.0603	**31.1758**	31.1574
40	**29.8065**	**29.8042**	29.3625	29.6822	**29.8042**	29.7966
60	**27.731**	**27.7289**	27.3329	27.5946	**27.7289**	27.7087
80	**25.5363**	**25.5375**	**25.7454**	**25.9448**	**25.9448**	25.5006
90	**24.6505**	**24.65**	**25.2148**	**25.3037**	**25.3037**	24.5864
100	**23.8188**	**23.8203**	**24.6355**	**24.673**	**24.673**	23.7521

The bolded PSNR values in Table 4-1 are greater than (in very few cases, equal to) the PSNR values obtained by BM3D. From the table we can see that edge guided schemes without pre-filtering (i.e. *SCHEME1*, *SCHEME3*) performs well for lower noise levels, while the addition of pre-filter (i.e. *SCHEME3-1*, *SCHEME3-3*) increases the PSNR for higher noise levels significantly (details of these intermediate schemes can be found in Section 3.2). In particular, for noise levels $\sigma > 60$, the schemes with selective pixel restoration step produce PSNR values almost one decibel higher than the original BM3D algorithm. Moreover, one can observe that our final proposed method, AEG-BM3D, easily outperforms BM3D for all noise levels. Notably, these better PSNR values also corresponds to better edge preservation and contrast preservation as we will see in the subjective quality comparison Section 4.2.3.

In Table A-1 of Appendix A we present experimental results for four more standard less textured images namely Man, Boat, Cameraman, and Couple. From this table we can also observe that our proposed method, AEG-BM3D, produces better denoising results for almost all noise levels except for a few cases.

Table 4-2: PSNR (dB) comparison for standard Mandril image among BM3D, our four experimental schemes and final proposed method for different noise levels

σ	SCHEME1	SCHEME3	SCHEME3-1	SCHEME3-3	AEG-BM3D	BM3D
2	44.2744	**44.5326**	43.5279	44.2744	44.2744	44.5166
5	37.9243	**38.0936**	37.7245	37.9102	37.9243	38.092
10	33.4354	33.4384	33.2251	33.3584	33.4384	33.4406
20	**29.1274**	**29.1224**	28.8142	29.0787	**29.1224**	29.1114
30	**26.846**	**26.8315**	26.2737	26.6569	**26.8315**	26.8046
40	**25.4003**	25.3887	24.8314	25.165	25.3887	25.3906
60	**23.4031**	**23.4056**	23.0906	**23.4873**	**23.4056**	23.3718
80	**22.0596**	**22.0756**	**22.0723**	**22.1969**	**22.1969**	21.9692
90	**21.5781**	**21.5849**	**21.7334**	**21.8169**	**21.8169**	21.4804
100	**21.169**	**21.1736**	**21.4005**	**21.4881**	**21.4881**	21.0728

However, for highly textured images, which has lots of repetitive patterns, the performance of our experimental schemes and proposed methods decreases a little as compared to the performance showed for less textured images, but still outperforms BM3D for most of the cases.

From Table 4-2, it is clear that, our experimental schemes and final proposed method outperforms BM3D for most noise levels for Mandril, the heavily textured image under discussion. However, the results obtained for lower noise level are not comprehensively greater than BM3D for Mandril. In fact, in some cases BM3D shows better denoising result. This is because, we used a block size of 4×4 for smooth regions to estimate pixel values when noise levels less than 10 and this fails to give better estimation when that particular pixel is not on a edge but there is high edge activity around that pixel. However, still for higher noise levels $\sigma > 60$ our method gives significantly better denoising result as compared to BM3D. In Table A-3 and Table A-4 of Appendix A, we present PSNR values for four more heavily textured images just like we did in our discussion for less textured images and we can see from the table that the same pattern, discussed above, follows for all of these heavily textured images.

From the above discussion it is clear that our proposed method consistently yields higher denoising PSNR values for all of the images and noise levels. If we calculate the results in percentage for all 16 less textured and 16 heavily textured test images for all noise levels we are using, we end up with Table 4-3. In this table, we present the performance of our intermediate experimental schemes and of the proposed method against the original BM3D method. We present the percentage as *Better* or *Less* when our methods exhibit higher or lower PSNR values respectively than BM3D. In all other cases the values are same.

One notable thing from this analysis is that our proposed method demonstrates excellent PSNR values for less textured images, in particular for noise levels greater than 40. Overall, for all images and all noise levels our proposed method AEG-BM3D exhibits significant improvement (i.e. 96.09%) over BM3D.

Table 4-3: PSNR performance comparison of examined schemes and our proposed method with BM3D for all 32 images in our test dataset

σ	SCHEME1		SCHEME3		SCHEME3-1		SCHEME3-3		AEG-BM3D	
	Better %	Less %	Better %	Less %	Better %	Less %	Better %	Less %	Better %	Less %
less textured Images										
All	80.59	19.41	79.41	20.59	50.59	49.41	61.18	38.82	84.12	15.88
> 40	100	0	100	0	88.24	11.76	94.12	5.88	100	0
heavily textured Images										
All	62.50	36.88	66.25	33.75	42.50	57.50	48.75	51.25	67.50	31.88
> 40	87.50	12.50	87.50	12.50	81.25	18.75	92.19	7.81	92.19	7.81
total (less textured Images + heavily textured Images)										
All	71.88	27.81	73.44	26.56	45.63	54.38	54.69	45.31	76.25	23.44
> 40	93.75	6.25	93.75	6.25	84.38	15.63	92.97	7.03	96.09	3.91

4.2.2   Structural similarity preservation

To compare the structural similarity preservation we use the same dataset and follow the same procedure employed in section 4.2.1 to present the experimental results for PSNR. Though our main goal was to improve PSNR, the most widely used numerical measure for comparing denoising performance; we still present the SSIM results for our test cases and show that while dealing with PSNR we do not let our proposed method down with another important metric, SSIM, which correlates aptly with human perception. Table 4-4 and Table 4-5 show the SSIM values for Lena and Mandril respectively. The SSIM values of other selected test images are presented in Table B-1 to Table B-4 of Appendix B.

Table 4-4: SSIM comparison for standard Lena image among BM3D, our four
experimental schemes and final proposed method for different noise levels

σ	SCHEME1	SCHEME3	SCHEME3-1	SCHEME3-3	AEG-BM3D	BM3D
2	**0.9777**	**0.9775**	**0.9776**	**0.9777**	**0.9777**	0.9775
5	**0.9461**	**0.9438**	**0.946**	**0.9461**	**0.9461**	0.9438
10	**0.9157**	**0.9157**	0.9098	0.9099	**0.9157**	0.9155
20	**0.8753**	**0.8752**	0.8725	**0.8752**	**0.8752**	0.8749
30	**0.8417**	**0.8416**	0.8331	0.8397	**0.8416**	0.841
40	**0.8118**	**0.8117**	0.8018	**0.8105**	**0.8117**	0.8083
60	**0.7729**	**0.7729**	0.7409	0.7552	**0.7729**	0.7699
80	0.7339	0.734	0.6822	0.7036	0.7036	0.7344
90	**0.7261**	**0.7261**	0.6708	0.6933	0.6933	0.7259
100	0.7109	0.7109	0.6492	0.6758	0.6758	0.7134

Table 4-5: SSIM comparison for standard Mandril image among BM3D, our four
experimental schemes and final proposed method for different noise levels

σ	SCHEME1	SCHEME3	SCHEME3-1	SCHEME3-3	AEG-BM3D	BM3D
2	0.9933	**0.9936**	0.993	0.9933	0.9933	0.9936
5	0.9744	**0.975**	0.9741	0.9744	0.9744	0.975
10	**0.9347**	**0.9346**	0.9339	**0.9348**	**0.9346**	0.9344
20	**0.851**	**0.8505**	0.8416	**0.8496**	**0.8505**	0.8495
30	**0.7736**	**0.7725**	0.7474	0.7648	**0.7725**	0.7705
40	**0.7001**	**0.6989**	0.6713	0.6902	**0.6989**	0.6988
60	**0.5731**	**0.5732**	0.5645	**0.5752**	**0.5732**	0.568
80	**0.4731**	**0.4749**	**0.4964**	**0.4978**	**0.4978**	0.46
90	**0.4348**	**0.4356**	**0.4668**	**0.4634**	**0.4634**	0.4235
100	**0.4123**	**0.4127**	**0.442**	**0.4416**	**0.4416**	0.3991

In Table 4-6 we present the total performance of our experimental schemes and final proposed method with compare to BM3D in terms of SSIM. From this table we can see that in terms of structural similarity preservation capability, our proposed method is better than BM3D for 60.31% of the cases, while BM3D is only better for 30.63% of the cases in total. However, no specific conclusion about superiority can be made from the values of SSIM as the results are mixed.

Table 4-6: Total SSIM performance comparison of examined schemes and our proposed method with BM3D for all 32 images in our test dataset

σ	SCHEME1		SCHEME3		SCHEME3-1		SCHEME3-3		AEG-BM3D	
	Better %	Less %	Better %	Less %	Better %	Less %	Better %	Less %	Better %	Less %
All	64.69	28.75	59.38	19.06	25.00	71.56	42.19	52.19	60.31	30.63
$5 < \sigma < 80$	76.25	19.38	75.00	14.38	18.75	81.25	39.38	55.00	75.00	14.38
$\sigma > 40$	69.53	28.91	66.41	32.03	33.59	65.63	52.34	47.66	59.38	39.84
$\sigma > 60$	67.71	30.21	64.58	34.38	37.50	61.46	55.21	44.79	55.21	44.79

4.2.3   Visual quality comparison and intensity profile

Visual quality comparison is amongst the most important criteria to perceive the performance of a denoising algorithm. Producing visually improved image is also one of the key purposes of image denoising, which also allow us to gain better performance for further image processing tasks, for example, object recognition or classification. Let us present some of experiments on standard test images. In this section we will compare BM3D only with our final proposed method dropping out comparisons with our intermediate experimental schemes for simplicity, as it is difficult to present a visual comparison of several methods in a concise manner.

Visual comparison of our proposed method, AEG-BM3D and original BM3D are presented in Figure 4-3 to Figure 4-9 for Mandril. We have also presented the same set of comparison for Lena and Boat from Figure C-1 to Figure C-14 in Appendix C. Here we have presented comparison for three standard images (i.e. Boat, Lena and Mandril) for

noise levels $\sigma \geq 20$. For noises levels less than that, the difference is not that perceivable as the amount of added noise is small. Both BM3D and our proposed method perform equally well, at least visually, to reconstruct image details and edges for these kinds of noisy images.

Though the obtained PSNR of BM3D is beyond the capabilities of most of the sophisticated and recent denoising methods, the non adaptive approach of original BM3D is not able to deliver highly sparse representation of images for finer image details or sharp and curved edges. This can be perceived in the examples shown in the following figures, in particular for higher noise levels. In contrast to BM3D, one can see that our proposed method is able to effectively reconstruct fine image details, sharp edges and at the same time preserves contrasts better. The adaptive edge guided pixel estimation step in our proposed method enable local adaptively to image features such as edges or small details which improves the sparsity as compared with the original BM3D. This enhanced sparsity allows the Wiener filtering stage (see Section 3.1.3) to reconstruct edges and preserve contrast better. This observation is also supported by improved PSNR and SSIM values presented earlier.

Our proposed method also outperforms BM3D for micro-texture images, which can be seen for the image of Mandril images (Figure 4-3 to Figure 4-9). From these images it is clear that edges were reconstructed significantly better than BM3D due to edge guided pixel restoration step in our method.

The denoising performance of our proposed method as compared to BM3D is further depicted in Figure 4-10 and Figure 4-11, where we present fragments of two standard noisy (for $\sigma = 70, 80, 90, and\ 100$) test images along with the fragment of the noiseless image and denoised ones. This comparison allows us to easily distinguish the visual improvement of our proposed method with respect to BM3D for edge reconstruction and contrast preservation. Although in Figure 4-10 and Figure 4-11, the true signal is almost completely buried under noise we can see that our method effectively restored the image details and edges.

(a)

(b)

(c)

(d)
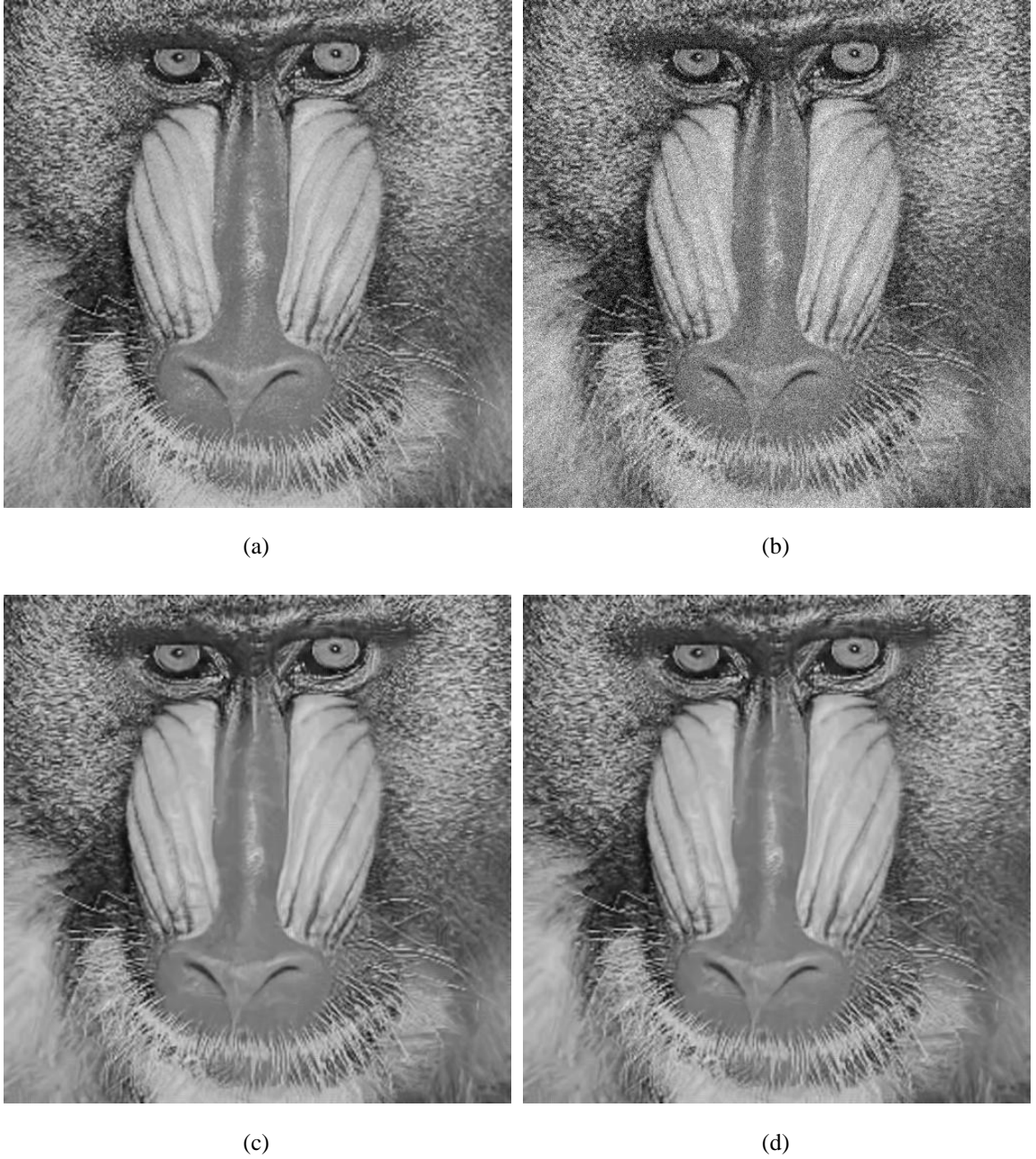
Figure 4-3: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Mandril at noise level $\sigma = 20$. (a) Noise free Mandril image. (b) Noisy image with $\sigma = 20$. (c) Denoised image using BM3D, PSNR = 29.1114, SSIM = 0.8495. (d) Denoised image using AEG-BM3D, PSNR = 29.1224, SSIM = 0.8505.

(a)   (b)

(c)   (d)
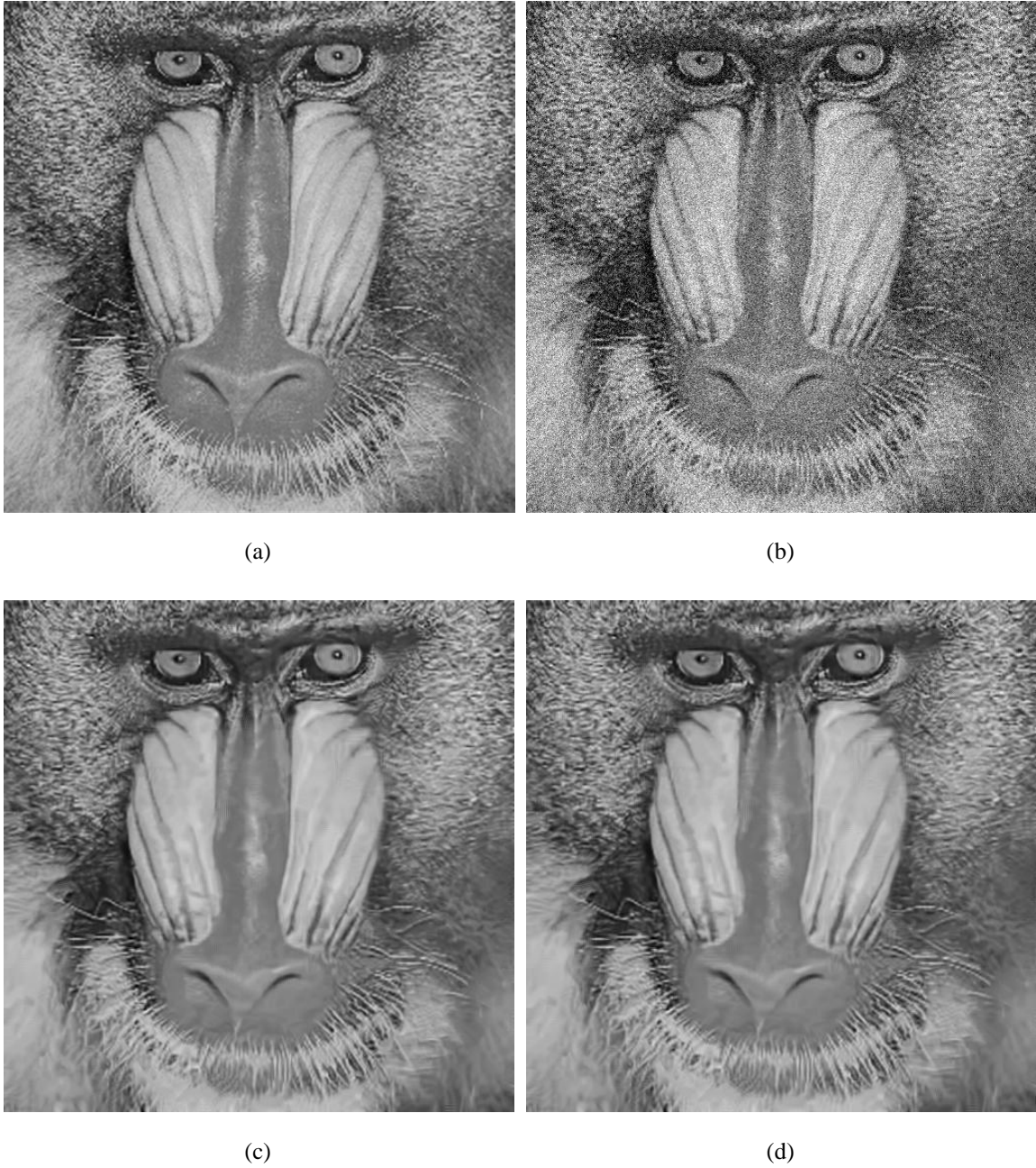
Figure 4-4: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Mandril at noise level $\sigma = 30$. (a) Noise free Mandril image. (b) Noisy image with $\sigma = 30$. (c) Denoised image using BM3D, PSNR = 26.8046, SSIM = 0.7705. (d) Denoised image using AEG-BM3D, PSNR = 26.8315, SSIM = 0.7725.

(a)   (b)

(c)   (d)
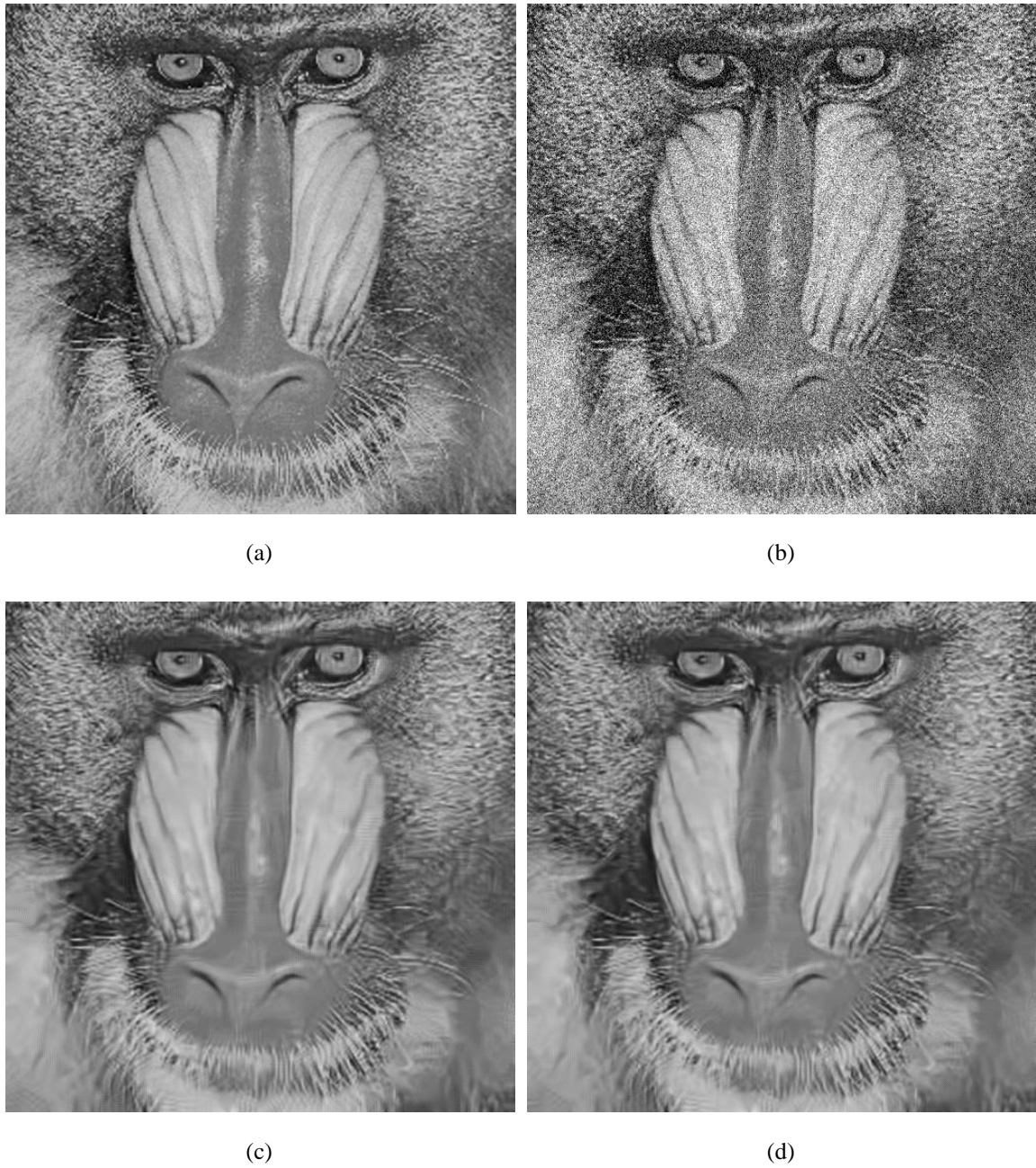
Figure 4-5: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Mandril at noise level $\sigma = 40$. (a) Noise free Mandril image. (b) Noisy image with $\sigma = 40$. (c) Denoised image using BM3D, PSNR = 25.3906, SSIM = 0.6988. (d) Denoised image using AEG-BM3D, PSNR = 25.3887, SSIM = 0.6989.

(a)   (b)

(c)   (d)
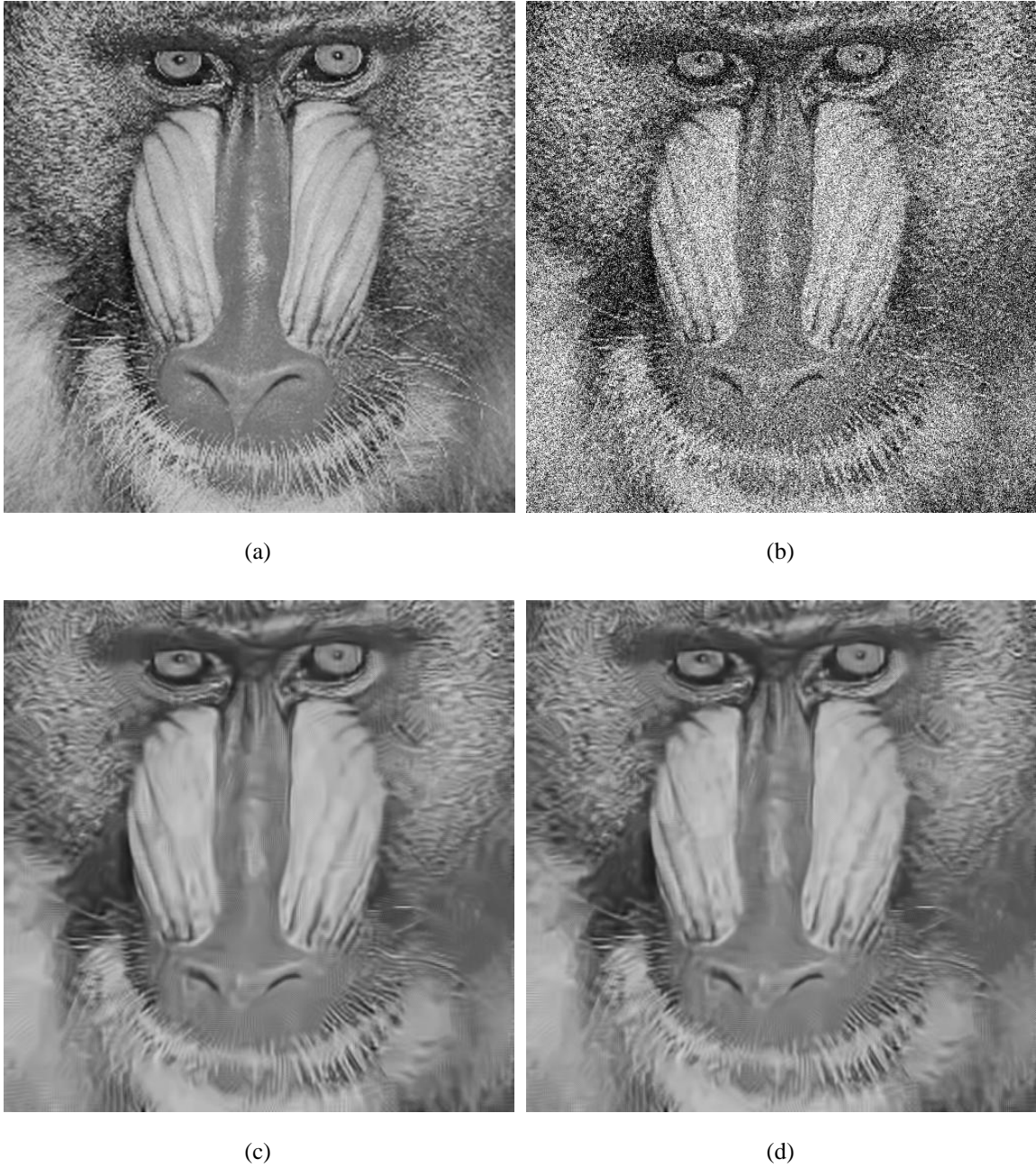
Figure 4-6: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Mandril at noise level $\sigma = 60$. (a) Noise free Mandril image. (b) Noisy image with $\sigma = 60$. (c) Denoised image using BM3D, PSNR = 23.3718, SSIM = 0.5680. (d) Denoised image using AEG-BM3D, PSNR = 23.4056, SSIM = 0.5732.

(a)   (b)

(c)   (d)
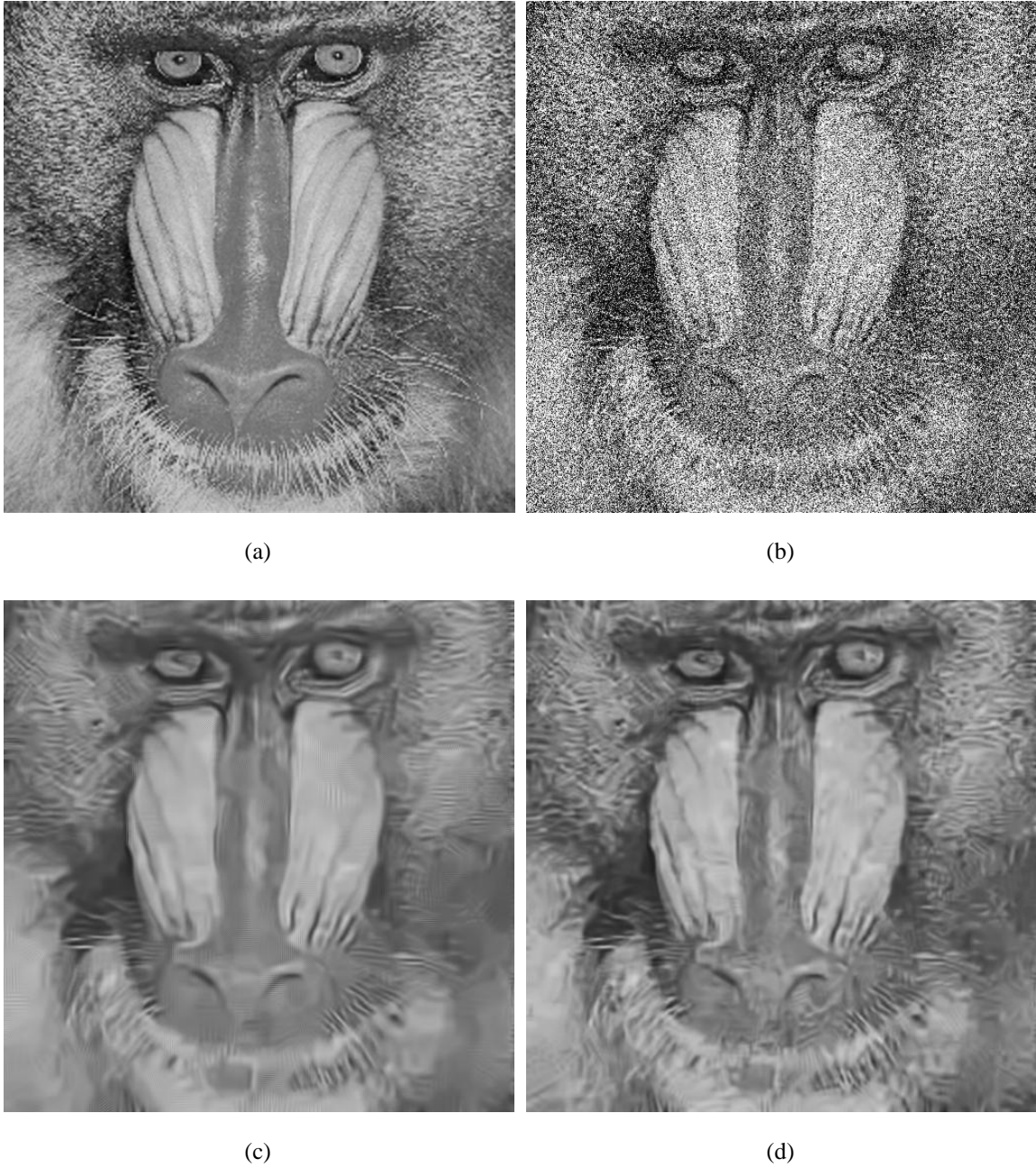
Figure 4-7: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Mandril at noise level $\sigma = 80$. (a) Noise free Mandril image. (b) Noisy image with $\sigma = 80$. (c) Denoised image using BM3D, PSNR = 21.9692, SSIM = 0.4600. (d) Denoised image using AEG-BM3D, PSNR = 22.1969, SSIM = 0.4978.

(a)   (b)

(c)   (d)
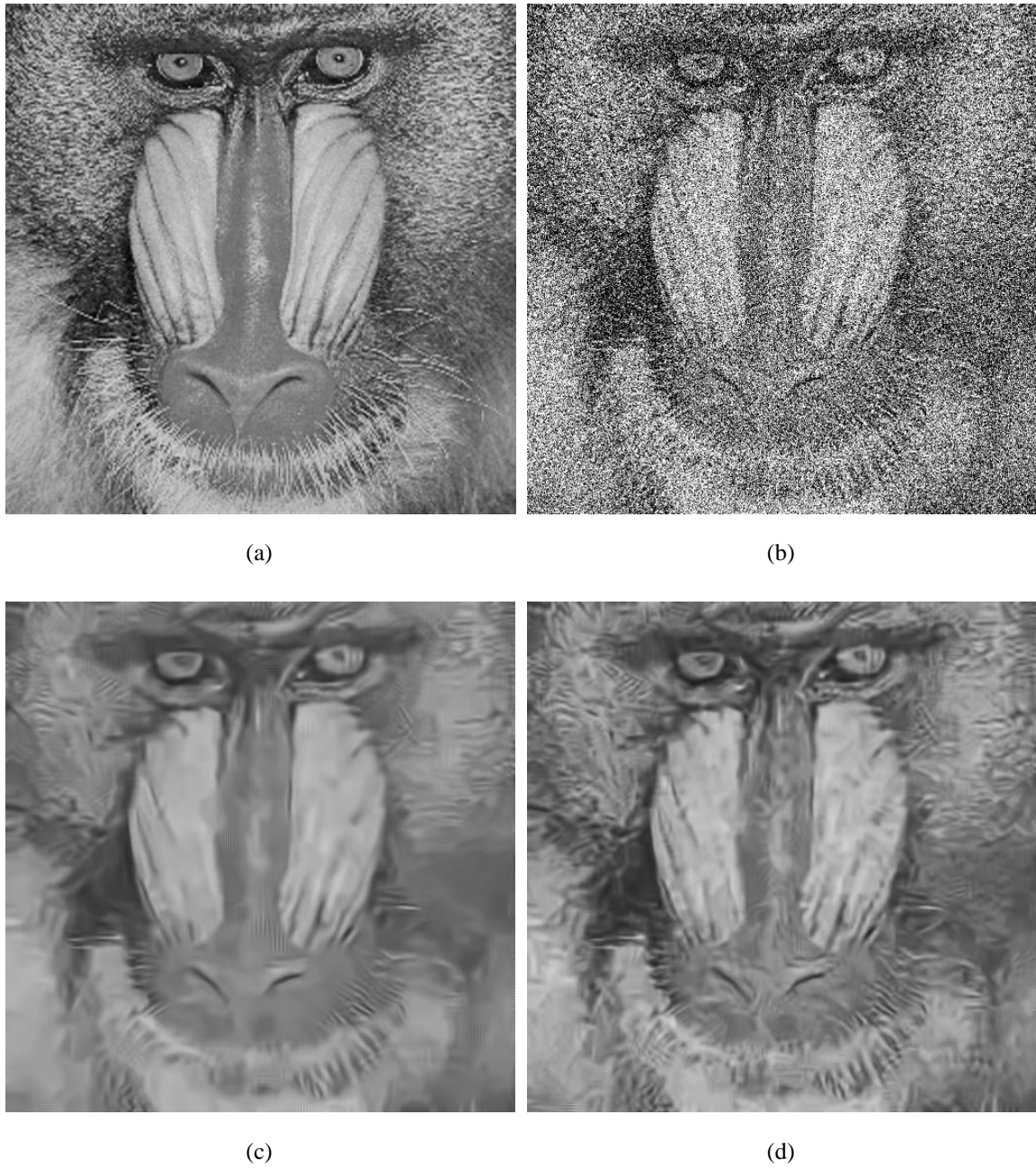
Figure 4-8: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Mandril at noise level $\sigma = 90$. (a) Noise free Mandril image. (b) Noisy image with $\sigma = 90$. (c) Denoised image using BM3D, PSNR = 21.4804, SSIM = 0.4235. (d) Denoised image using AEG-BM3D, PSNR = 21.8169, SSIM = 0.4634.

(a)   (b)

(c)   (d)
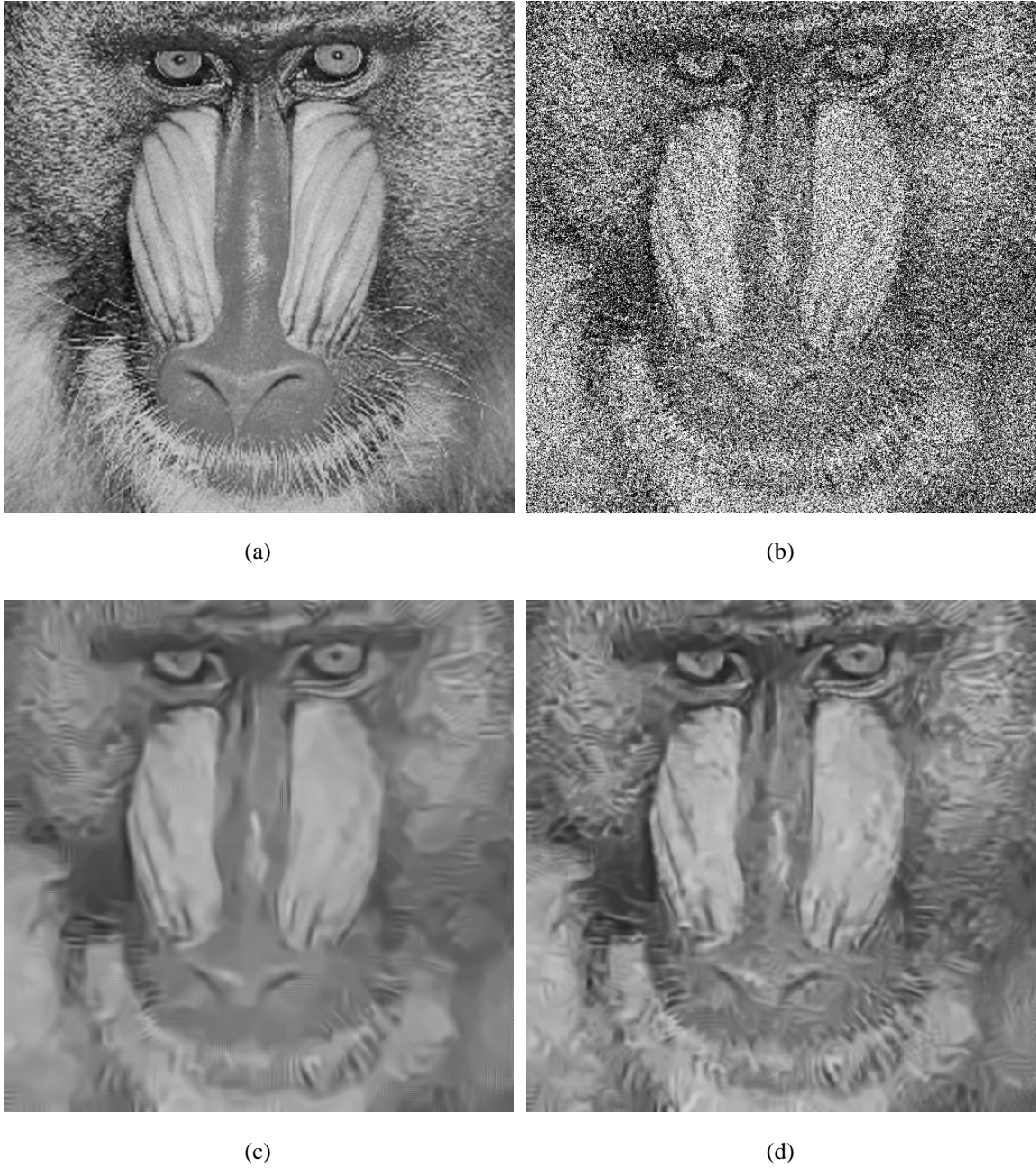
Figure 4-9: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Mandril at noise level $\sigma = 100$. (a) Noise free Mandril image. (b) Noisy image with $\sigma = 100$. (c) Denoised image using BM3D, PSNR = 21.0728, SSIM = 0.3991. (d) Denoised image using AEG-BM3D, PSNR = 21.4881, SSIM = 0.4416.
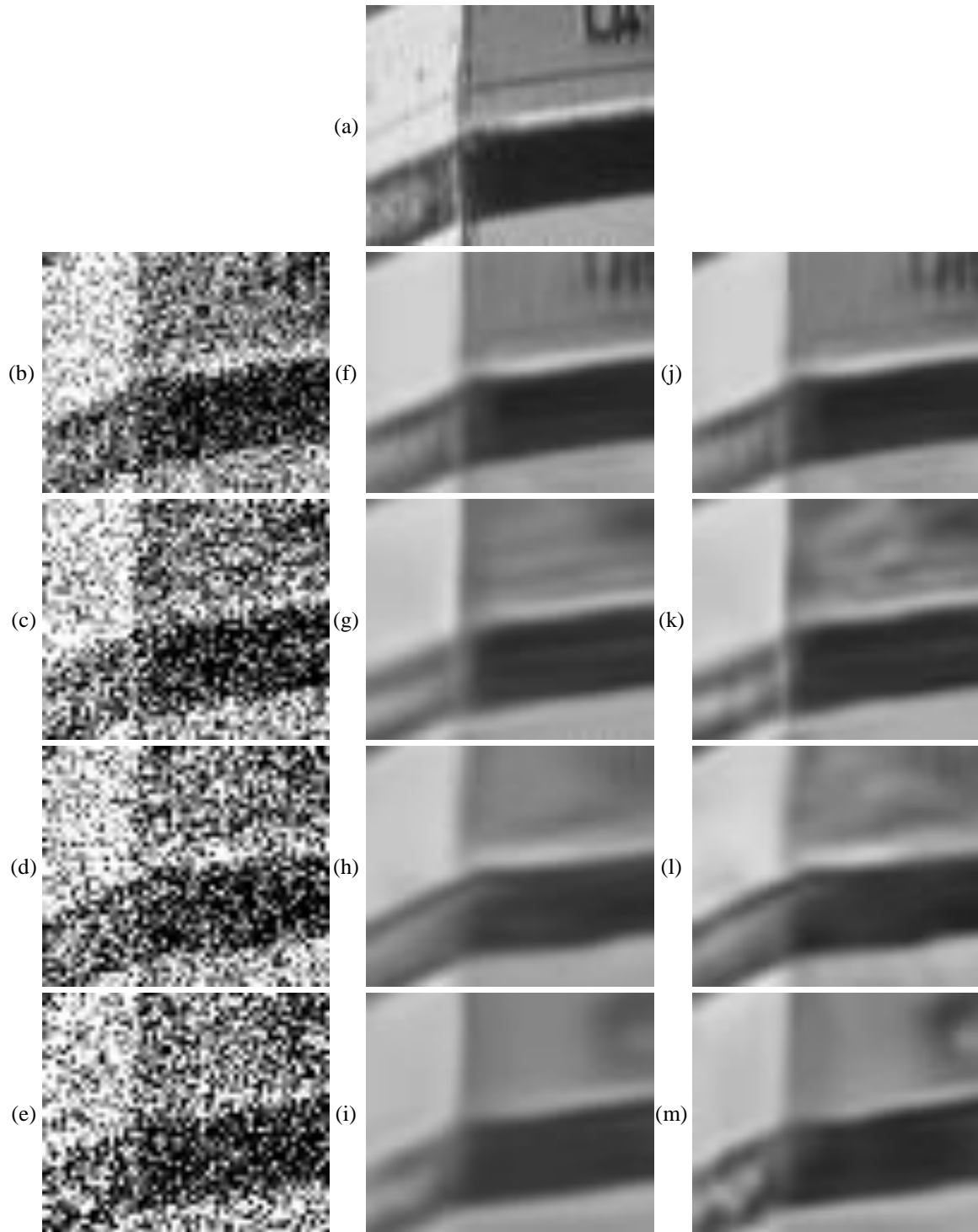
Figure 4-10: Comparison of edge and contrast preservation for zoomed fragment of Boat image. (a) Noise free fragment. (b)-(e) Noisy fragment with $\sigma = 60, 80, 90, and\ 100$ respectively. (f)-(i) Denoised fragment using BM3D (j)-(m) Denoised fragment using AEG-BM3D.
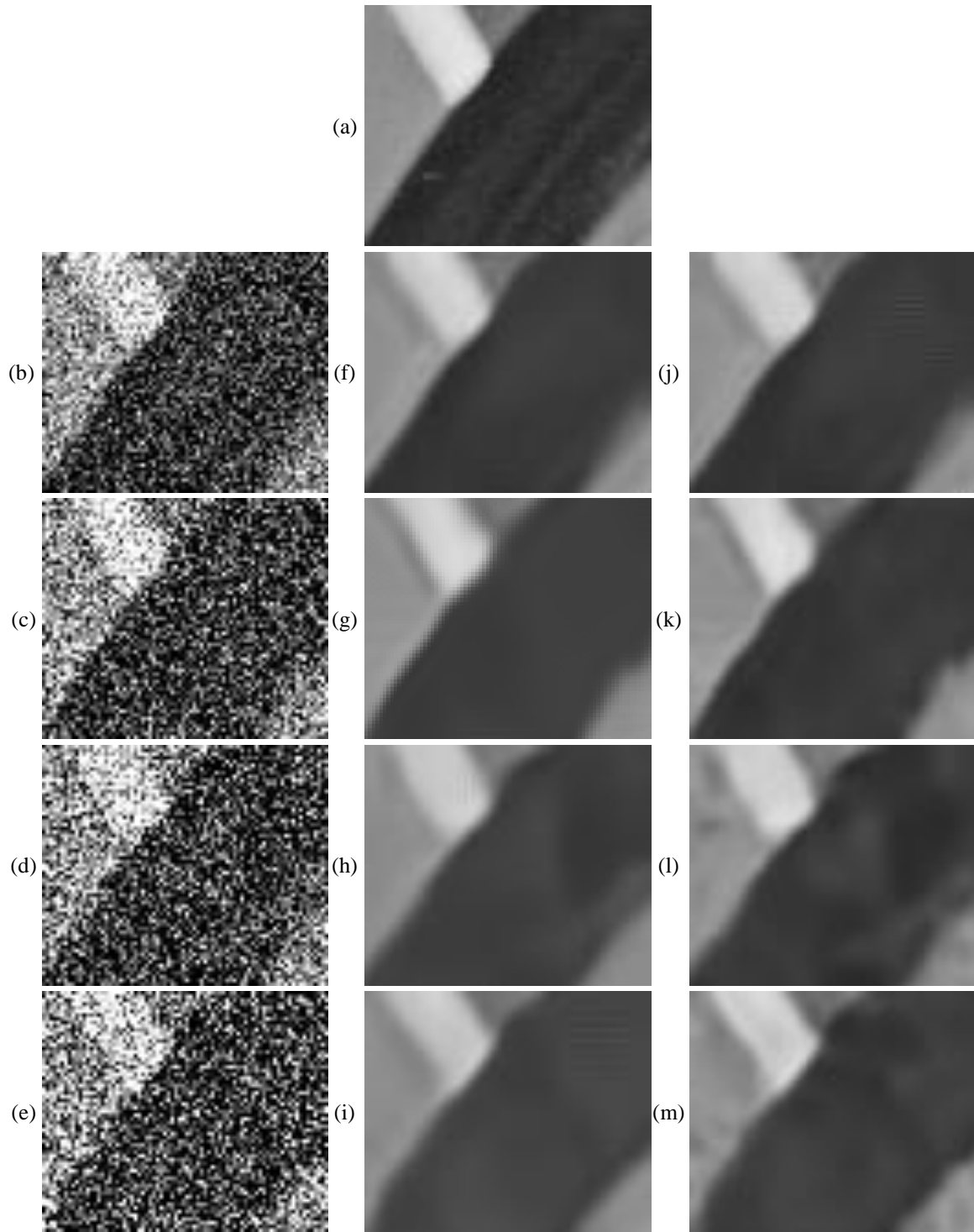
Figure 4-11: Comparison of edge and contrast preservation for zoomed fragment of Lena image. (a) Noise free fragment. (b)-(e) Noisy fragment with $\sigma = 60, 80, 90, and\ 100$ respectively. (f)-(i) Denoised fragment using BM3D (j)-(m) Denoised fragment using AEG-BM3D

From this visual comparison we can confirm that the edge guided pixel estimation step used in our proposed method is a feasible approach to better preserve the edges and micro-textured regions. It is also confirmed that AEG-BM3D produces denoised images with more contrast and details than the original BM3D.

The performance of our proposed method for edge preservation as compare to BM3D can also be justified by intensity profiling. Intensity profile is a graphical representation of an image line showing its intensity of each pixel along the line. We plot all the pixel numbers along x axis and the intensities of each pixel along y axis. This is a popular tool for visual observation of how much the denoised image is deviated from the original one. Here, we will present the intensity profile of a selected scan line of the standard House image along with their denoised version by BM3D and our proposed method.

Figure 4-13 shows an intensity profile of the popular house image at scan line 50, which is illustrated in Figure 4-12. In Figure 4-13, the x axis contains all the 256 pixels along the line 50. At each pixel, we plot its intensity along the y axis. The red curve denotes the true noise free image line while the blue one indicates its denoised version by BM3D with noise level equals 80. The point here is to consider how much the denoised version is deviated from the original one.

Figure 4-12: Row number 50 of the standard House image is chosen as the scan line (presented by a dark red straight line) to generate intensity profiles.

Now, let us consider Figure 4-14 where we plot the intensity profile of the original image with that of our proposed method. Clearly, for the pixels between 120-255, the intensity profile of proposed method is much closer than that of BM3D. Again, in this scan line, there are two sharp edges, for which the proposed method tends to be closed to the original image. Since BM3D does not treat edges differently than smooth areas, in Figure 4-13, we can observe that the intensity profile deviates in edge lines. Here we are presenting the intensity profile for higher noise levels only as it is difficult to perceive the difference between intensity profiles of BM3D and our proposed method when the noise level is very low.

In figures Figure 4-15 to Figure 4-18 we are presenting the same scan line of the same image for other noise levels. In every case, we can see that the proposed method tends to produce better result than original BM3D.

However, one can also observe that the denoised images produced by our method introduce some artifacts in smooth areas which impede the visual quality, both from the denoised images and intensity profiles presented in the discussion. This is due to the edge guided estimation approach that we employed in Section 3.1.2 and the reason can be given as follows: We estimated the pixel based on an edge guidance and selected the estimated pixel value from two different estimation available (i.e. $\hat{y}^{guide}$ and $\hat{y}^{est}$). In some cases, it is possible that we take the estimation for one pixel from one of this estimation, while taking its neighboring values from the other estimation, introducing the artifact that we have observed.
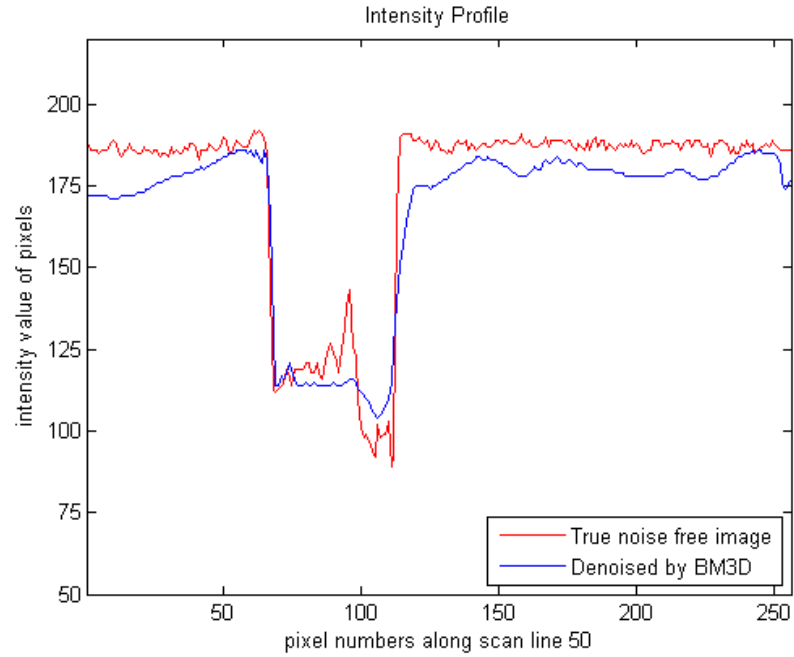
Figure 4-13: Intensity profile of House image and denoised image by BM3D at scan Line 50 (σ=80)
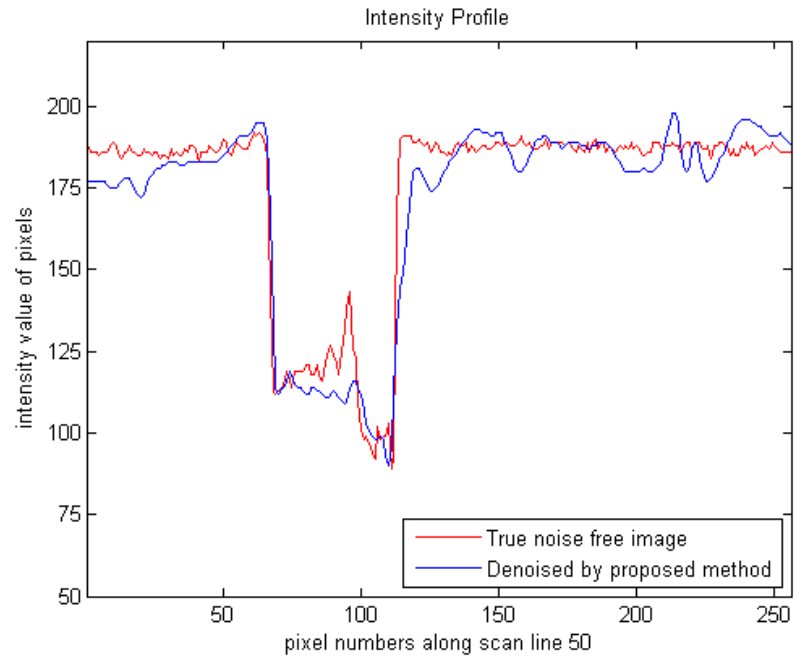
Figure 4-14: Intensity profile of House image and denoised image by proposed method at scan line 50 (σ=80)
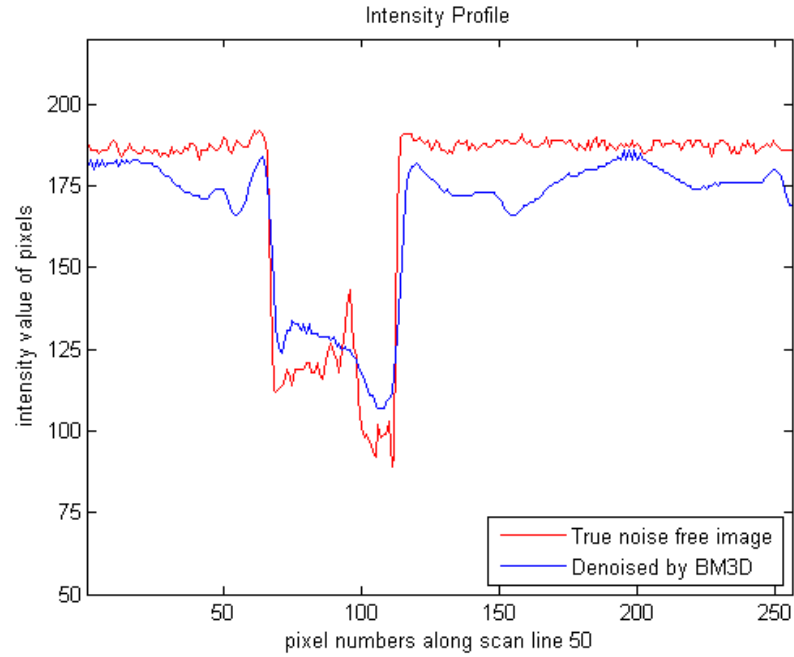
Figure 4-15: Intensity profile of House image and denoised image by BM3D at scan line 50 (σ=90)
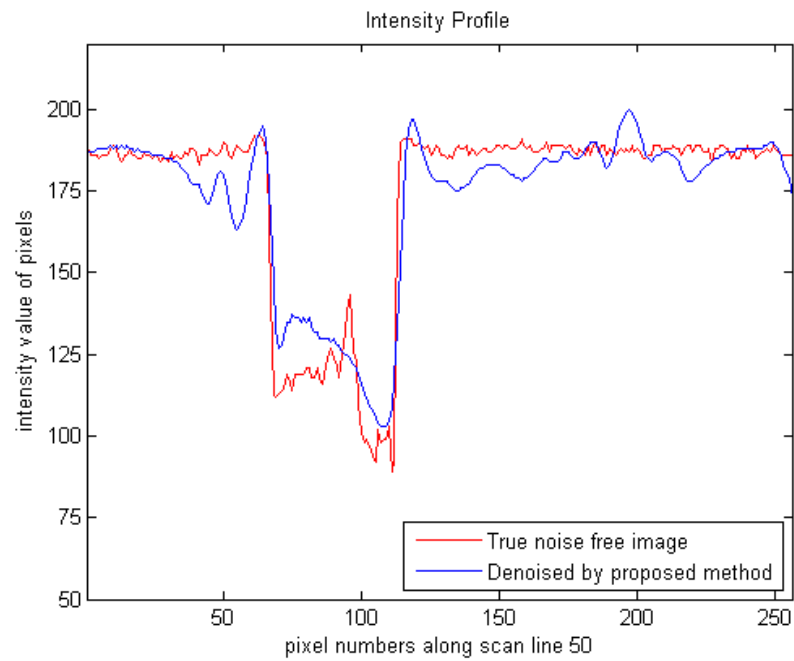
Figure 4-16: Intensity profile of House image and denoised image by proposed method at scan line 50 (σ=90)
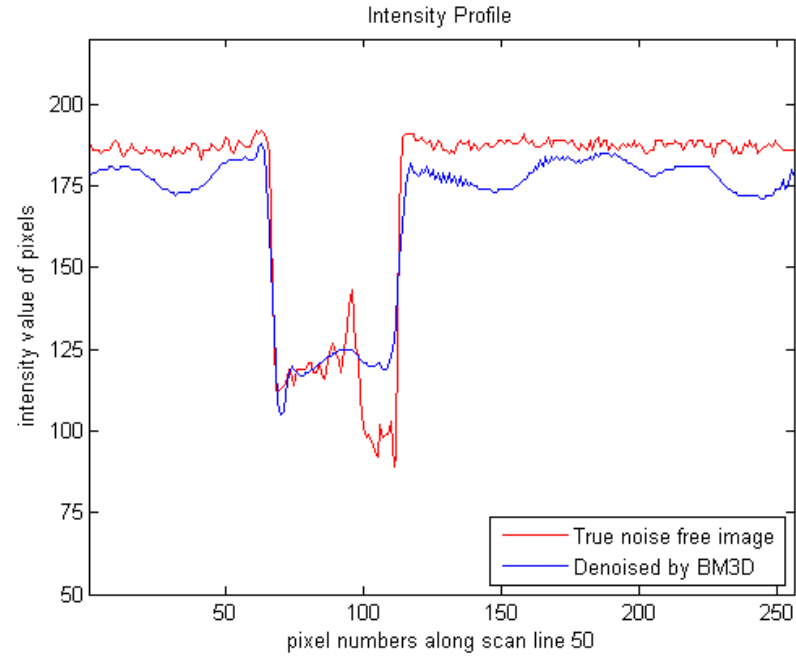
Figure 4-17: Intensity profile of House image and denoised image by BM3D at scan line 50 (σ=100)

Figure 4-18: Intensity profile of House image and denoised image by proposed method at scan line 50 (σ=100)

4.2.4   Average SSIM, PSNR, and Runtime Comparison

We have used the standard Lena image and Mandril image to compare the average SSIM, PSNR, and Runtime of our proposed method AEG-BM3D with BM3D. Table 4-7 and Table 4-8 shows the average SSIM, PSNR, and runtime for Lena and Mandril image respectively along with their standard deviations, denoted by STD. These results are calculated as the average over 10 runs measured in seconds on a 2.00 GHz core i7 processor with 8GB RAM running under LINUX based OS Ubuntu 12. Actual CPU times are counted instead of elapsed time.

Table 4-7: Average SSIM, PSNR, and Runtime performance comparison of our proposed method AEG-BM3D with BM3D for standard Lena image

Sigma		SSIM		PSNR		Runtime in Seconds	
		AEG-BM3D	BM3D	AEG-BM3D	BM3D	AEG-BM3D	BM3D
2	Average	0.977730	0.977620	43.554940	43.541750	11.603020	10.459029
	STD	0.000046	0.000075	0.010214	0.009731	1.229432	1.933546
5	Average	0.945950	0.943780	38.786510	38.709390	12.793990	11.153765
	STD	0.000175	0.000160	0.016017	0.014875	1.918378	2.043569
10	Average	0.915540	0.915460	35.883080	35.885450	14.715000	11.564860
	STD	0.000369	0.000383	0.015742	0.016321	1.035653	2.477389
20	Average	0.875310	0.875120	32.987370	32.984300	13.854470	11.032515
	STD	0.000658	0.000658	0.020894	0.022668	1.488480	2.480096
30	Average	0.841570	0.841170	31.201270	31.188090	15.883890	11.242479
	STD	0.001023	0.001043	0.023671	0.024972	2.010854	1.504644
40	Average	0.814720	0.811750	29.865570	29.849270	21.510700	20.989660
	STD	0.001795	0.001483	0.044928	0.043668	2.710620	3.175685
60	Average	0.768570	0.765650	27.648770	27.613410	22.214800	18.511230
	STD	0.002622	0.002730	0.070433	0.067009	2.656465	2.617268
80	Average	0.719710	0.735580	25.391570	25.521770	23.600770	22.564640
	STD	0.002773	0.003151	0.068865	0.066393	2.757263	3.769843
90	Average	0.703190	0.724070	24.463360	24.568130	24.204130	21.936040
	STD	0.002239	0.002037	0.047206	0.060949	3.170428	2.697791
100	Average	0.690010	0.712690	23.611330	23.679560	24.157920	22.746670
	STD	0.002315	0.001377	0.034317	0.036759	2.599486	4.524312

Table 4-8: Average SSIM, PSNR, and Runtime performance comparison of our proposed method AEG-BM3D with BM3D for standard Mandril image

Sigma		SSIM		PSNR		Runtime in Seconds	
		AEG-BM3D	BM3D	AEG-BM3D	BM3D	AEG-BM3D	BM3D
2	Average	0.993320	0.993610	44.294120	44.530330	12.679419	11.512850
	STD	0.000040	0.000030	0.015660	0.015077	2.567868	2.656986
5	Average	0.974460	0.974950	37.934750	38.095590	11.634940	8.998986
	STD	0.000150	0.000169	0.010087	0.010881	1.253602	1.646073
10	Average	0.934820	0.934680	33.451970	33.457060	14.118150	10.144318
	STD	0.000260	0.000271	0.018001	0.017437	1.029466	2.653537
20	Average	0.850800	0.849640	29.133600	29.118200	14.063640	9.759393
	STD	0.000587	0.000623	0.021254	0.021110	1.712800	2.183927
30	Average	0.771740	0.769700	26.853310	26.822990	16.030400	9.610290
	STD	0.001426	0.001374	0.020743	0.020767	1.723739	1.510655
40	Average	0.700500	0.700100	25.365130	25.367720	21.494560	17.989280
	STD	0.001521	0.001547	0.027196	0.028129	2.130412	2.879697
60	Average	0.571920	0.566960	23.382770	23.347020	21.340370	20.125880
	STD	0.002527	0.002581	0.020836	0.021354	2.591571	4.145865
80	Average	0.493050	0.462550	22.170720	21.991090	23.548420	20.838790
	STD	0.002816	0.002482	0.026201	0.029685	2.686709	3.413759
90	Average	0.460200	0.423350	21.692170	21.470440	25.892500	21.190950
	STD	0.001972	0.002286	0.028022	0.018580	4.092002	2.926935
100	Average	0.435410	0.395770	21.287050	21.044780	25.398420	20.164410
	STD	0.002234	0.002516	0.022995	0.022154	1.493795	2.042982

From the Table 4-7 and Table 4-8 we can see that the average runtime of our proposed method is greater than BM3D. This is expected as we have incorporated two new stages (i.e. the prefiltering stage and the edge guidance stage) along with the stages of BM3D. But the running time increment is linear with respect to the size of the input image as the new incorporated stages of our proposed method can be computed in linear time.

4.3   Overall results discussion

The experimental results presented in this chapter confirm the superiority of our proposed method, AEG-BM3D, over original BM3D, both numerically and visually. We have used 32 standard test images for our experiment and we showed that our proposed method performs well for these standard test images. Also we have conducted test with 10 different noise levels for AWGN.

To compare the denoising results quantitatively or objectively, we have evaluated both PSNR and SSIM for all the test images. In terms of PSNR our method shows significantly improved denoising results as compared to BM3D algorithm. Although for lower noise levels the improvement is not that significant, our method truly excelled when the image is highly corrupted with AWGN. The SSIM values presented also demonstrate the superiority of our proposed method in terms of structural similarity preservation.

When compared visually we can clearly see that our method can reduce more noise while preserving sharp edges, micro-textured regions, and fine details. We also observed better contrast preservation in denoised images along with some artifacts.

In summary, the denoised images of our proposed method, AEG-BM3D, shows:

- better reconstruction of sharp edges (e.g. borders of objects in Boat and Lena image),
- better reconstruction of micro-textured regions and repeating patterns (e.g. the texture of Lena's hair and face texture of Mandril),
- better preservation of contrast (this can be seen in all the images),
- PSNR results are better than BM3D for most of the noise levels,
- SSIM values are mixed,
- Few artifacts in smooth regions of denoised images.

<div align="center">**Chapter 5**</div>

5 Concluding Remarks and Future Work

5.1 Conclusion

In this study we have introduced an extension to the BM3D algorithm [1] by introducing edge guidance and selective prefiltering of noisy pixels. In Chapter 4 we have extensively tested and analyzed our proposed method in both visual quality measures and quantitative measures and verified its superiority over the current state of the art denoising algorithm, BM3D.

In our proposed method, AEG-BM3D, the improvement contributed by the pre-filtering and edge guided pixel estimation in step 1 and step 2 before the final step of Wiener filtering can be justified as follows. We tried to remove as much noise as possible from the basic estimate. We also estimated the edges better than BM3D in the hard-thresholding step due to the edge guidance process and Canny edge detection. Because more noise has already been attenuated, preserving the edges, both block matching and estimation of pixels in third step are more accurate when these processes are applied on the refined basic estimate. It results in sparser representation in transform domain. Moreover, Wiener filleting in the third step is more effective when it is applied by using a more reliable basic estimate as an oracle.

Experimental results confirm that our proposed extension shows better result than BM3D in most of the cases. The PSNR improvement is well for higher noise levels. Though one can argue that the SSIM improvement is modest, but during subjective evaluation we have shown that the images denoised by our proposed method preserves edges and details better along with superior contrast preservation when compared to state of the art BM3D.

5.2 Future work

The current state of the art denoising algorithms shows near optimal denoising performance [94]. However, we have demonstrated in our study that there is still room for perfection in some cases. Images rich in texture or images corrupted with higher level of noise are harder to denoise and some improvement can be possible. Some visual enhancement is also possible, as we have seen that some artifact is introduced by both our proposed method AEG-BM3D and BM3D.

In the prefiltering step we have used a simple neighborhood based technique for detection of noisy pixels and restored their values using median filter. Instead, in this step we can assume the noisy pixels to be restored as the pixels corrupted by impulse noise and use some current state of the art method for impulse noise detection and restore their values using more statistics from the neighboring healthy pixels. Also we have used the pre-filtered image as input to both the hard-thresholding step and the Wiener filtering step. It will be interesting to experiment with pre-filtered image as the input to the hard-thresholding step only, while feeding the original corrupted image as the input to the Wiener filtering step as it uses the information from basic estimate obtained in the first step.

We have applied the edge guidance in the hard-thresholding step only. It will be interesting to verify whether some improvement is possible by using the same procedure in the Wiener filtering step.

Visual evaluation shows some artifacts were introduced in images denoised by our proposed method, particularly for images corrupted with higher noise level. Future work shall address this problem by using more smooth edge guidance in the hard-thresholding step.

AEG-BM3D can easily be applied to color images with a small change during the prefiltering and edge guidance step. Besides AWGN, it can also be applied to other types of noise models, for example, non-Gaussian noise and Poison noise. Just like BM3D our

proposed method can also be extended for other image processing tasks such as image and video restoration, deblurring, and sharpening.

Bibliography

[1]   Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Image denoising by sparse 3-D transform-domain collaborative filtering. Image Processing, IEEE Transactions on, 16(8), 2080-2095.

[2]   Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2006, February). Image denoising with block-matching and 3D filtering. In Electronic Imaging 2006. International Society for Optics and Photonics.

[3]   Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2008, August). A nonlocal and shape-adaptive transform-domain collaborative filtering. In Proc. Int. Workshop on Local and Non-Local Approx. in Image Process., LNLA.

[4]   Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2009). BM3D image denoising with shape-adaptive principal component analysis. In SPARS'09-Signal Processing with Adaptive Sparse Structured Representations.

[5]   Katkovnik, V., Foi, A., Dabov, K., & Egiazarian, K. (2008, August). Spatially adaptive support as a leading model-selection tool for image filtering. In Proc. First Workshop Inf. Th. Methods Sci. Eng., WITMSE, Tampere.

[6]   Chen, Q., & Wu, D. (2010). Image denoising by bounded block matching and 3D filtering. Signal Processing, 90(9), 2778-2783.

[7]   Danielyan, A., Katkovnik, V., & Egiazarian, K. (2010, August). Image deblurring by augmented Lagrangian with BM3D frame prior. In Workshop on Information

Theoretic Methods in Science and Engineering (WITMSE), Tampere, Finland (pp. 16-18).

[8]   Danielyan, A., Katkovnik, V., & Egiazarian, K. (2012). BM3D frames and variational image deblurring. Image Processing, IEEE Transactions on, 21(4), 1715-1728.

[9]   Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007, September). Joint image sharpening and denoising by 3D transform-domain collaborative filtering. In Proc. 2007 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP (Vol. 2007).

[10]   Mäkitalo, M., & Foi, A. (2011, February). Spatially adaptive alpha-rooting in BM3D sharpening. In IS&T/SPIE Electronic Imaging (pp. 787012-787012). International Society for Optics and Photonics.

[11]   Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2008, February). Image restoration by sparse 3D transform-domain collaborative filtering. In Electronic Imaging 2008 (pp. 681207-681207). International Society for Optics and Photonics.

[12]   Dabov, K. (2010). Image and Video Restoration with Nonlocal Transform-Domain Filtering (Doctoral dissertation, Thesis for the degree of Doctor of Science in Technology, Tampere University of Technology, Tampere, Finland).

[13]   Dabov, K., Foi, A., & Egiazarian, K. (2007, September). Video denoising by sparse 3D transform-domain collaborative filtering. In Proc. 15th European Signal Processing Conference (Vol. 1, No. 2, p. 7).

[14]   Maggioni, M., Boracchi, G., Foi, A., & Egiazarian, K. (2011, February). Video denoising using separable 4D nonlocal spatiotemporal transforms. In IS&T/SPIE Electronic Imaging (pp. 787003-787003). International Society for Optics and Photonics.

[15]   Maggioni, M., Boracchi, G., Foi, A., & Egiazarian, K. (2012). Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms. Image Processing, IEEE Transactions on, 21(9), 3952-3966.

[16]   Pedada, R., Kugu, E., Li, J., Yue, Z., & Shen, Y. (2009, February). Parameter optimization for image denoising based on block matching and 3D collaborative filtering. In SPIE Medical Imaging (pp. 725925-725925). International Society for Optics and Photonics.

[17]   Li, X., Chen, Z., & Xing, Y. (2012, October). Multi-segment limited-angle CT reconstruction via a BM3D filter. In Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2012 IEEE (pp. 2390-2394). IEEE.

[18]   Kang, D., Slomka, P., Nakazato, R., Woo, J., Berman, D. S., Kuo, C. C. J., & Dey, D. (2013, March). Image denoising of low-radiation dose coronary CT angiography by an adaptive block-matching 3D algorithm. In SPIE Medical Imaging (pp. 86692G-86692G). International Society for Optics and Photonics.

[19]   Donoho, D. L., & Johnstone, J. M. (1994). Ideal spatial adaptation by wavelet shrinkage. Biometrika, 81(3), 425-455.

[20] Huang, J., Ma, J., Liu, N., Feng, Q., & Chen, W. (2011, March). Projection data restoration guided non-local means for low-dose computed tomography reconstruction. In Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on (pp. 1167-1170). IEEE.

[21] Feruglio, P. F., Vinegoni, C., Gros, J., Sbarbati, A., & Weissleder, R. (2010). Block matching 3D random noise filtering for absorption optical projection tomography. Physics in medicine and biology, 55(18), 5401.

[22] Wang, L., Meng, Z., Yao, X. S., Liu, T., Su, Y., & Qin, M. (2012). Adaptive Speckle Reduction in OCT Volume Data Based on Block-Matching and 3-D Filtering. Photonics Technology Letters, IEEE, 24(20), 1802-1804.

[23] Peltonen, S., Tuna, U., Sanchez-Monge, E., & Ruotsalainen, U. (2011, October). PET sinogram denoising by block-matching and 3D filtering. In Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2011 IEEE (pp. 3125-3129). IEEE.

[24] Lin, X., & Qiu, T. (2011, October). Denoise MRI images using sparse 3D transformation domain collaborative filtering. In Biomedical Engineering and Informatics (BMEI), 2011 4th International Conference on (Vol. 1, pp. 233-236). IEEE.

[25] Chong, B., & Zhu, Y. K. (2013). Speckle reduction in optical coherence tomography images of human finger skin by wavelet modified BM3D filter. Optics Communications, 291, 461-469.

[26]   Danielyan, A., Katkovnik, V., & Egiazarian, K. (2011, September). Deblurring of

Poissonian images using BM3D frames. In SPIE Optical Engineering+ Applications

(pp. 813812-813812). International Society for Optics and Photonics.

[27]   Begovic, B., Stankovic, V., & Stankovic, L. (2011, September). Contrast

enhancement and denoising of Poisson and Gaussian mixture noise for solar

images. In Image Processing (ICIP), 2011 18th IEEE International Conference on

(pp. 185-188). IEEE.

[28]   Gao, J., Chen, Q., & Blasch, E. (2012, July). Image denoising in the presence of

non-Gaussian, power-law noise. In Aerospace and Electronics Conference

(NAECON), 2012 IEEE National (pp. 103-108). IEEE.

[29]   Matrecano, M., Poggi, G., & Verdoliva, L. (2012). Improved BM3D for Correlated

Noise Removal. In VISAPP (1) (pp. 129-134).

[30]   Zhang, Y. S., Zhu, S. J., & Li, Y. J. (2012). BM3D Denoising Algorithm with Adaptive

Block-Match Thresholds. Applied Mechanics and Materials, 229, 1715-1720.

[31]   Poderico, M., Parrilli, S., Poggi, G., & Verdoliva, L. (2010, October). Sigmoid

shrinkage for BM3D denoising algorithm. In Multimedia Signal Processing

(MMSP), 2010 IEEE International Workshop on (pp. 423-426). IEEE.

[32]   Hou, Y., Chen, T., Yang, D., Zhu, L., & Yang, H. (2012, January). Image denoising by

block-matching and 1D filtering. In Fourth International Conference on Machine

Vision (ICMV 11) (pp. 83490A-83490A). International Society for Optics and

Photonics.

[33]   Dai, L., Zhang, Y., & Li, Y. (2013). Image Denoising Using BM3D Combining Tetrolet
       Prefiltering. Information Technology Journal, 12(10).

[34]   Elad, M., & Aharon, M. (2006, June). Image denoising via learned dictionaries and
       sparse representation. In Computer Vision and Pattern Recognition, 2006 IEEE
       Computer Society Conference on (Vol. 1, pp. 895-900). IEEE.

[35]   Wiener, N., Wiener, N., Mathematician, C., Wiener, N., & Wiener, N. (1964).
       Extrapolation, interpolation, and smoothing of stationary time series: with
       engineering applications (Vol. 8). MIT press.

[36]   Hou, Y., Zhao, C., Yang, D., & Cheng, Y. (2011). Comments on Image Denoising by
       Sparse 3-D Transform-Domain Collaborative Filtering. Image Processing, IEEE
       Transactions on, 20(1), 268-270.

[37]   Atto, A. M., Pastor, D., & Mercier, G. (2008). Smooth sigmoid wavelet shrinkage
       for non-parametric estimation. Acoustics, Speech and Signal Processing, ICASSP,
       IEEE International Conference on. IEEE, 2008.

[38]   Mittal, A., Moorthy, A. K., & Bovik, A. C. (2012, February). Automatic parameter
       prediction for image denoising algorithms using perceptual quality features. In
       SPIE Proceedings Human Vision and Electronic Imaging.

[39]   Makitalo, M., & Foi, A. (2011). Optimal inversion of the Anscombe
       transformation in low-count Poisson image denoising. Image Processing, IEEE
       Transactions on, 20(1), 99-109.

[40] Foi, A. (2011, March). Noise estimation and removal in mr imaging: The variance-stabilization approach. In Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on (pp. 1809-1814). IEEE.

[41] Tomasi, C., & Manduchi, R. (1998, January). Bilateral filtering for gray and color images. In Computer Vision, 1998. Sixth International Conference on (pp. 839-846). IEEE.

[42] Yaroslavsky, L. P., & Yaroslavskij, L. P. (1985). Digital picture processing. An introduction. Digital picture processing. An introduction.. LP Yaroslavsky (LP Yaroslavskij). Springer Seriesin Information Sciences, Vol. 9. Springer-Verlag, Berlin-Heidelberg-New York-Tokyo. 12+ 276 pp. Price DM 112.00 (1985). ISBN 3-540-11934-5 (FR Germany), ISBN 0-387-11934-5 (USA)., 1.

[43] Lee, J. S. (1981). Speckle analysis and smoothing of synthetic aperture radar images. Computer graphics and image processing, 17(1), 24-32.

[44] Rudin, L. I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena, 60(1), 259-268.

[45] Chambolle, A. (2004). An algorithm for total variation minimization and applications. Journal of Mathematical imaging and vision, 20(1-2), 89-97.

[46] Chambolle, A., & Lions, P. L. (1997). Image recovery via total variation minimization and related problems. Numerische Mathematik, 76(2), 167-188.

[47] Chan, T. F., Osher, S., & Shen, J. (2001). The digital TV filter and nonlinear denoising. Image Processing, IEEE Transactions on, 10(2), 231-241.

[48] Osher, S., Burger, M., Goldfarb, D., Xu, J., & Yin, W. (2004). Using geometry and iterated refinement for inverse problems (1): Total variation based image restoration. Department of Mathematics, UCLA, LA, CA, 90095, 04-13.

[49] Tadmor, E., Nezzar, S., & Vese, L. (2004). A multiscale image representation using hierarchical (BV, L 2) decompositions. Multiscale Modeling & Simulation, 2(4), 554-579.

[50] Hu, Y., & Jacob, M. (2012). Higher degree total variation (HDTV) regularization for image recovery. Image Processing, IEEE Transactions on, 21(5), 2559-2571.

[51] Rodrıguez, P., & Wohlberg, B. (2009). Efficient minimization method for a generalized total variation functional. IEEE Trans. Image Process, 18(2), 322-332.

[52] Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 12(7), 629-639.

[53] Sapiro, G. (2006). Geometric partial differential equations and image analysis. Cambridge university press.

[54] Chan, T. F., & Shen, J. J. (2005). Image processing and analysis: variational, PDE, wavelet, and stochastic methods. Siam.

[55] Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 12(7), 629-639.

[56] Guichard, F., Moisan, L., & Morel, J. M. (2002, March). A review of PDE models in image processing and image analysis. In Journal de Physique IV (Proceedings) (Vol. 12, No. 1, pp. 137-154). EDP sciences.

[57] Awate, S. P., & Whitaker, R. T. (2005, June). Higher-order image statistics for unsupervised, information-theoretic, adaptive, image filtering. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 2, pp. 44-51). IEEE.

[58] Buades, A., Coll, B., & Morel, J. M. (2005). A review of image denoising algorithms, with a new one. Multiscale Modeling & Simulation, 4(2), 490-530.

[59] Kervrann, C., & Boulanger, J. (2006). Optimal spatial adaptation for patch-based image denoising. Image Processing, IEEE Transactions on, 15(10), 2866-2878.

[60] Kindermann, S., Osher, S., & Jones, P. W. (2005). Deblurring and denoising of images by nonlocal functionals. Multiscale Modeling & Simulation, 4(4), 1091-1115.

[61] Gilboa, G., & Osher, S. (2008). Nonlocal operators with applications to image processing. Multiscale Modeling & Simulation, 7(3), 1005-1028.

[62] Gilboa, G., & Osher, S. (2007). Nonlocal linear image regularization and supervised segmentation. Multiscale Modeling & Simulation, 6(2), 595-630.

[63] Alexander, S. K., Vrscay, E. R., & Tsurumi, S. (2007). An examination of the statistical properties of domain-range block matching in fractal image coding. preprint.

[64] Burt, P. J., & Adelson, E. H. (1983). The Laplacian pyramid as a compact image code. Communications, IEEE Transactions on, 31(4), 532-540.

[65] Meyer, Y. (1993). Wavelets-algorithms and applications. Wavelets-Algorithms and applications Society for Industrial and Applied Mathematics Translation., 142 p., 1.

[66] Mallat, S. G. (1989). Multiresolution approximations and wavelet orthonormal bases of L^2 (R). Transactions of the American Mathematical Society, 315(1), 69-87.

[67] Mallat, S. G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 11(7), 674-693.

[68] Mallat, S. (2008). A wavelet tour of signal processing. Academic press., 3rd ed., Third Edition: The Sparse Way. Academic Press.

[69] Coifman, R. R., & Donoho, D. L. (1995). Translation-invariant de-noising (pp. 125-150). Springer New York.

[70] Simoncelli, E. P., Freeman, W. T., Adelson, E. H., & Heeger, D. J. (1992). Shiftable multiscale transforms. Information Theory, IEEE Transactions on, 38(2), 587-607.

[71] Kingsbury, N. (2001). Complex wavelets for shift invariant analysis and filtering of signals. Applied and computational harmonic analysis, 10(3), 234-253..

[72] Simoncelli, E. P., & Freeman, W. T. (1995, October). The steerable pyramid: A flexible architecture for multi-scale derivative computation. In Image Processing, International Conference on (Vol. 3, pp. 3444-3444). IEEE Computer Society.

[73] Greenspan, H., Belongie, S., Goodman, R., Perona, P., Rakshit, S., & Anderson, C. H. (1994, June). Overcomplete steerable pyramid filters and rotation invariance. In Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on (pp. 222-228). IEEE.

[74] Muresan, D. D., & Parks, T. W. (2003, September). Adaptive principal components and image denoising. In ICIP (1) (pp. 101-104).

[75] Hoyer, P. (1999). Independent component analysis in image denoising. Master's Thesis, Helsinki University of Technology.

[76] Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. Image Processing, IEEE Transactions on, 16(5), 1395-1411.

[77] Donoho, D. L., & Johnstone, J. M. (1994). Ideal spatial adaptation by wavelet shrinkage. Biometrika, 81(3), 425-455.

[78] Donoho, D. L., & Johnstone, I. M. (1995). Adapting to unknown smoothness via wavelet shrinkage. Journal of the american statistical association, 90(432), 1200-1224.

[79] Stein, C. M. (1981). Estimation of the mean of a multivariate normal distribution. The annals of Statistics, 1135-1151.

[80] Pizurica, A., & Philips, W. (2006). Estimating the probability of the presence of a signal of interest in multiresolution single-and multiband image denoising. Image Processing, IEEE Transactions on, 15(3), 654-665.

[81] Sendur, L., & Selesnick, I. W. (2002). Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency. Signal Processing, IEEE Transactions on, 50(11), 2744-2756.

[82] Sendur, L., & Selesnick, I. W. (2002). Bivariate shrinkage with local variance estimation. Signal Processing Letters, IEEE, 9(12), 438-441.

[83] Luisier, F., Blu, T., & Unser, M. (2007). A new SURE approach to image denoising: Interscale orthonormal wavelet thresholding. Image Processing, IEEE Transactions on, 16(3), 593-606.

[84] Yan, F., Cheng, L., & Peng, S. (2008). A new interscale and intrascale orthonormal wavelet thresholding for SURE-based image denoising. Signal Processing Letters, IEEE, 15, 139-142.

[85] Yu, H., Zhao, L., & Wang, H. (2009). Image denoising using trivariate shrinkage filter in the wavelet domain and joint bilateral filter in the spatial domain. Image Processing, IEEE Transactions on, 18(10), 2364-2369.

[86] Sikora, T., & Makai, B. (1995). Shape-adaptive DCT for generic coding of video. Circuits and Systems for Video Technology, IEEE Transactions on, 5(1), 59-62..

[87] Foi, A. (2007). Pointwise shape-adaptive DCT image filtering and signal-dependent noise estimation.

[88] Mallat, S. (2009). Geometrical grouplets. Applied and Computational Harmonic Analysis, 26(2), 161-180.

[89] Shen, J. (2003). Inpainting and the fundamental problem of image processing. SIAM news, 36(5), 1-4.

[90] Guleryuz, O. G. (2003, November). Weighted overcomplete denoising. In Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on (Vol. 2, pp. 1992-1996). IEEE.

[91] Canny, J. (1986). A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, (6), 679-698.

[92] Krommweh, J. (2010). Tetrolet transform: A new adaptive Haar wavelet algorithm for sparse image representation. Journal of Visual Communication and Image Representation, 21(4), 364-374.

[93] Smith, S. M., & Brady, J. M. (1997). SUSAN—a new approach to low level image processing. International journal of computer vision, 23(1), 45-78.

[94] Chatterjee, P., & Milanfar, P. (2010). Is denoising dead?. Image Processing, IEEE Transactions on, 19(4), 895-911.

[95] Lebrun, M. (2012). An analysis and implementation of the BM3D image denoising method. Image Processing On Line, (2012).

[96] CMU Image Data Base - http://vasc.ri.cmu.edu/idb/html/motion/

[97] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. Image Processing, IEEE Transactions on, 13(4), 600-612.

[98] Statistics   Basic   -   http://www4.nau.edu/microanalysis/microprobe/Statistics-Basic.html.

[99] Mittal, A., Moorthy, A. K., & Bovik, A. C. (2011, November). Blind/referenceless image spatial quality evaluator. In Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on (pp. 723-727). IEEE.

Appendices

Appendix A.
PSNR for selected standard images

Table A-1: PSNR (dB) comparison for standard Man and Boat images among BM3D, our four experimental schemes and final proposed method for different noise levels

σ	SCHEME1	SCHEME3	SCHEME3-1	SCHEME3-3	AEG-BM3D	BM3D
Man						
2	**43.5457**	43.4819	42.9735	43.4583	**43.5457**	43.485
5	**37.7604**	37.6678	37.5985	**37.7359**	**37.7604**	37.6735
10	**33.8015**	**33.798**	33.7419	**33.8318**	**33.798**	33.7916
20	**30.4474**	**30.4473**	30.2096	30.4085	**30.4473**	30.4378
30	**28.7515**	**28.7512**	28.3983	28.6523	**28.7512**	28.7461
40	27.458	**27.4667**	27.1113	27.3429	**27.4667**	27.3817
60	**25.6895**	**25.6975**	25.5663	**25.6992**	**25.6975**	25.6155
80	**24.1857**	**24.1921**	**24.4492**	**24.5661**	**24.5661**	24.0964
90	**23.2551**	**23.2613**	**23.9335**	**23.9307**	**23.9307**	23.1692
100	**22.6412**	**22.6455**	**23.4451**	**23.4189**	**23.4189**	22.5421
Boat						
2	43.1182	**43.1414**	42.5873	43.085	43.1182	43.1403
5	**37.3464**	**37.3243**	37.2138	**37.3464**	**37.3464**	37.3187
10	**33.8105**	**33.8111**	33.7615	**33.8644**	**33.8111**	33.8095
20	30.6796	30.6849	30.4418	30.6493	30.6849	30.6902
30	28.8458	28.851	28.3904	28.7303	28.851	28.8529
40	**27.4279**	**27.4322**	26.9959	27.2648	**27.4322**	27.3769
60	**25.3871**	**25.3935**	25.1499	25.3252	**25.3935**	25.355
80	**23.6918**	**23.7009**	**23.8009**	**23.9182**	**23.9182**	23.6621
90	**22.9302**	**22.9353**	**23.2529**	**23.3414**	**23.3414**	22.8584
100	**22.2424**	**22.2477**	**22.8864**	**22.8964**	**22.8964**	22.175

Table A-2: PSNR (dB) comparison for standard Cameraman and Couple images among BM3D, our four experimental schemes and final proposed method for different noise levels

σ	SCHEME1	SCHEME3	SCHEME3-1	SCHEME3-3	AEG-BM3D	BM3D
Cameraman						
2	**43.9597**	43.8287	39.981	41.8708	**43.9597**	43.8318
5	**38.1362**	38.0827	36.7492	37.5484	**38.1362**	38.0906
10	**33.8924**	**33.8872**	33.2674	33.6378	**33.8872**	33.8852
20	30.0635	30.0703	29.4298	29.8429	30.0703	30.1032
30	28.0241	28.0388	27.4873	27.8454	28.0388	28.0719
40	**26.1962**	**26.2102**	25.7641	26.0094	**26.2102**	26.1147
60	**23.6928**	**23.7073**	**23.7146**	**23.8123**	**23.7073**	23.5987
80	**21.7923**	**21.8012**	**22.2477**	**22.1974**	**22.1974**	21.7183
90	**20.9579**	**20.962**	**21.7302**	**21.6412**	**21.6412**	20.875
100	**20.1095**	**20.1175**	**21.1255**	**20.9588**	**20.9588**	20.0129
Couple						
2	**43.0916**	43.0826	42.689	**43.0916**	**43.0916**	43.0832
5	**37.4362**	37.3591	37.3316	**37.4291**	**37.4362**	37.3603
10	33.8451	33.8465	33.8075	**33.9079**	33.8465	33.851
20	**30.5545**	**30.5548**	30.306	30.5183	**30.5548**	30.5425
30	**28.6125**	**28.6096**	28.1849	28.4895	**28.6096**	28.5898
40	**27.2012**	**27.198**	26.7479	27.0327	**27.198**	27.1841
60	**25.1486**	**25.147**	24.8818	25.0359	**25.147**	25.1236
80	**23.5197**	**23.5234**	**23.5719**	**23.6797**	**23.6797**	23.4911
90	**22.8324**	**22.8356**	**22.993**	**23.1174**	**23.1174**	22.7918
100	**22.2022**	**22.2045**	**22.6833**	**22.7297**	**22.7297**	22.1428

Table A-3: PSNR (dB) comparison for standard Barbara and Fingerprint images among BM3D, our four experimental schemes and final proposed method for different noise levels.

σ	SCHEME1	SCHEME3	SCHEME3-1	SCHEME3-3	AEG-BM3D	BM3D
Barbara						
2	43.6505	**43.7073**	42.6158	43.6505	43.6505	43.7008
5	38.3514	**38.3975**	37.9514	38.3514	38.3514	38.3899
10	**34.9944**	34.9957	34.3711	34.784	**34.9957**	34.9887
20	**31.7272**	31.7224	31.0047	31.5994	**31.7224**	31.6955
30	**29.8447**	29.8315	28.7591	29.5514	**29.8315**	29.7868
40	28.3166	28.2989	26.992	27.8213	28.2989	28.3344
60	25.8156	25.7977	24.844	25.4039	25.7977	25.847
80	23.6435	23.6196	23.3027	**23.6832**	**23.6832**	23.6616
90	22.5812	22.5612	**22.6132**	22.8494	**22.8494**	22.6093
100	21.7061	21.6885	**22.0618**	22.1557	**22.1557**	21.7381
Fingerprint						
2	42.9371	**42.9501**	42.9284	42.9371	42.9371	42.9493
5	**36.7304**	36.7069	**36.7261**	36.7304	36.7304	36.7079
10	32.5512	32.5543	**32.5928**	32.6044	32.5543	32.5561
20	28.8283	**28.8342**	28.6839	28.8173	**28.8342**	28.834
30	26.7707	**26.7737**	26.4804	26.7044	**26.7737**	26.7713
40	**25.3958**	25.4031	25.0775	25.2826	**25.4031**	25.3486
60	**23.2935**	23.3044	23.248	23.3308	23.3044	23.2339
80	**21.6866**	21.6947	22.0816	22.0624	**22.0624**	21.6048
90	**20.9466**	20.9482	21.5656	21.5004	**21.5004**	20.8807
100	**20.2144**	20.2108	21.0785	20.9409	**20.9409**	20.1802

Table A-4: PSNR (dB) comparison for standard House and Shoe images among BM3D, our four experimental schemes and final proposed method for different noise levels.

σ	SCHEME1	SCHEME3	SCHEME3-1	SCHEME3-3	AEG-BM3D	BM3D
House						
2	**44.6179**	44.5737	**44.6179**	**44.6179**	**44.6179**	44.5753
5	39.7285	39.7297	39.7285	39.7285	39.7285	39.7339
10	36.5933	**36.6013**	36.3279	36.3452	**36.6013**	36.6006
20	**33.7019**	**33.7029**	33.4999	33.6897	**33.7029**	33.6938
30	**31.8981**	**31.8991**	31.3522	31.7783	**31.8991**	31.867
40	**30.5189**	**30.5348**	29.8659	30.3039	**30.5348**	30.4538
60	**28.0795**	**28.0808**	27.3526	27.7633	**28.0808**	28.0653
80	**25.8092**	25.8068	25.8157	**26.0211**	**26.0211**	25.7544
90	24.6542	24.664	25.1173	**25.1994**	**25.1994**	24.5936
100	**23.8497**	23.8509	24.5849	**24.6315**	**24.6315**	23.837
Shoe						
2	47.9662	**48.252**	47.9662	47.9662	47.9662	48.1903
5	43.2197	**43.9594**	43.2197	43.2197	43.2197	43.8087
10	**40.6527**	**40.561**	39.079	39.0776	**40.561**	40.1852
20	**36.7288**	36.5804	36.4531	**36.57**	36.5804	35.9948
30	**32.4314**	32.3589	32.2601	32.3353	32.3589	31.9577
40	29.1596	29.1399	29.4793	29.3402	29.1399	30.2805
60	24.8187	24.8188	**26.4451**	25.8354	24.8188	25.0346
80	**22.2223**	22.222	24.9628	**24.0441**	**24.0441**	22.1107
90	21.0928	21.0928	24.2811	**23.2556**	**23.2556**	20.9546
100	**20.0176**	**20.0176**	23.4891	**22.4062**	**22.4062**	19.8889

Appendix B.
SSIM for selected standard images

Table B-1: SSIM comparison for standard Man and Boat images among BM3D, our four experimental schemes and final proposed method for different noise levels

σ	SCHEME1	SCHEME3	SCHEME3-1	SCHEME3-3	AEG-BM3D	BM3D
			Man			
2	**0.9847**	**0.9845**	0.9842	**0.9847**	**0.9847**	0.9845
5	**0.9539**	**0.9527**	**0.9533**	**0.9539**	**0.9539**	0.9527
10	**0.9051**	**0.9049**	0.903	0.9039	**0.9049**	0.9045
20	**0.83**	**0.8297**	0.8236	0.8289	**0.8297**	0.829
30	**0.7746**	**0.7744**	0.7638	0.7716	**0.7744**	0.774
40	**0.729**	**0.7293**	0.7191	**0.7275**	**0.7293**	0.7229
60	**0.6659**	**0.6663**	0.6569	**0.6656**	**0.6663**	0.6606
80	**0.6262**	**0.6265**	0.6113	**0.6216**	**0.6216**	0.6215
90	**0.6018**	**0.6021**	0.5899	**0.599**	**0.599**	0.5982
100	**0.5864**	**0.5866**	0.5714	**0.582**	**0.582**	0.5818
			Boat			
2	0.9829	**0.9831**	0.9828	0.9829	0.9829	0.9831
5	**0.9416**	**0.9405**	**0.9414**	**0.9416**	**0.9416**	0.9404
10	**0.8881**	**0.888**	**0.8902**	**0.8909**	**0.888**	0.8876
20	**0.8218**	**0.8218**	0.8156	0.8209	**0.8218**	0.8215
30	**0.7737**	**0.7736**	0.7579	0.7694	**0.7736**	0.7731
40	**0.7282**	**0.7282**	0.7116	0.7229	**0.7282**	0.7256
60	**0.6605**	**0.6606**	0.6425	0.6533	**0.6606**	0.6583
80	**0.607**	**0.6075**	0.5902	0.6006	0.6006	0.6049
90	**0.5806**	**0.5808**	0.5622	0.5762	0.5762	0.5771
100	**0.5607**	**0.561**	0.5515	**0.5625**	**0.5625**	0.5563

Table B-2: SSIM comparison for standard Cameraman and Couple images among BM3D, our four experimental schemes and final proposed method for different noise levels

σ	SCHEME1	SCHEME3	SCHEME3-1	SCHEME3-3	AEG-BM3D	BM3D
Cameraman						
2	**0.9831**	**0.9824**	0.9811	**0.9826**	**0.9831**	0.9824
5	0.9596	**0.9601**	0.9576	0.9591	0.9596	0.96
10	**0.9283**	**0.9281**	0.9178	0.9204	**0.9281**	0.9275
20	**0.8674**	**0.8671**	0.8569	0.8648	**0.8671**	0.8667
30	**0.8198**	**0.8199**	0.808	0.8183	**0.8199**	0.8197
40	**0.7792**	**0.7794**	0.76	0.7713	**0.7794**	0.7731
60	**0.7049**	**0.7053**	0.6892	**0.7034**	**0.7053**	0.6997
80	**0.6519**	**0.6522**	0.6312	0.6458	0.6458	0.647
90	**0.6252**	**0.6253**	0.5959	0.6125	0.6125	0.6241
100	**0.6022**	**0.6025**	0.592	**0.5983**	0.5983	0.5978
Couple						
2	**0.9843**	**0.9843**	0.9841	**0.9843**	**0.9843**	0.9843
5	**0.9506**	**0.9493**	**0.9504**	**0.9505**	**0.9506**	0.9493
10	**0.9058**	**0.9058**	**0.9068**	**0.9075**	**0.9058**	0.9056
20	**0.8409**	**0.8407**	0.8349	**0.8401**	**0.8407**	0.84
30	**0.785**	**0.7847**	0.7698	0.7811	**0.7847**	0.7834
40	**0.7391**	**0.7386**	0.7198	0.7325	**0.7386**	0.7381
60	**0.6523**	**0.652**	0.6315	0.6416	**0.652**	0.6499
80	**0.5897**	**0.5898**	0.5698	0.5813	0.5813	0.5885
90	**0.5615**	**0.5616**	0.5355	0.55	0.55	0.56
100	**0.5399**	**0.5399**	0.5239	0.5362	0.5362	0.5367

Table B-3: SSIM comparison for standard Barbara and Fingerprint images among BM3D, our four experimental schemes and final proposed method for different noise levels

σ	SCHEME1	SCHEME3	SCHEME3-1	SCHEME3-3	AEG-BM3D	BM3D
			Barbara			
2	0.9861	**0.9863**	0.9859	0.9861	0.9861	0.9863
5	**0.9649**	**0.9646**	**0.9646**	**0.9649**	**0.9649**	0.9646
10	**0.9418**	**0.9417**	0.9356	0.9366	**0.9417**	0.9417
20	**0.9044**	**0.9042**	0.8967	0.9031	**0.9042**	0.9038
30	**0.868**	**0.8676**	0.8448	0.8623	**0.8676**	0.8663
40	0.8307	0.8296	0.7881	0.8161	0.8296	0.8309
60	0.7573	0.7561	0.6973	0.7282	0.7561	0.7612
80	0.69	0.6883	0.6275	0.6566	0.6566	0.6942
90	0.6551	0.6537	0.5945	0.6178	0.6178	0.6618
100	0.6208	0.6193	0.5676	0.5891	0.5891	0.628
			Fingerprint			
2	**0.997**	**0.997**	**0.997**	**0.997**	**0.997**	0.997
5	**0.988**	**0.9879**	**0.988**	**0.988**	**0.988**	0.9879
10	**0.9693**	**0.9693**	**0.9694**	**0.9696**	**0.9693**	0.9692
20	**0.9299**	**0.9298**	0.9268	**0.9295**	**0.9298**	0.9295
30	**0.8921**	**0.8921**	0.8843	0.89	**0.8921**	0.8917
40	**0.8579**	**0.8578**	0.8477	0.8542	**0.8578**	0.8567
60	**0.7902**	**0.7904**	**0.7911**	**0.7931**	**0.7904**	0.7882
80	**0.7172**	**0.7173**	**0.7426**	**0.7381**	**0.7381**	0.7127
90	**0.6808**	**0.6804**	**0.7195**	**0.7129**	**0.7129**	0.677
100	**0.6343**	**0.6332**	**0.6928**	**0.6807**	**0.6807**	0.6326

Table B-4: SSIM comparison for standard House and Shoe images among BM3D, our four experimental schemes and final proposed method for different noise levels

σ	SCHEME1	SCHEME3	SCHEME3-1	SCHEME3-3	AEG-BM3D	BM3D
House						
2	**0.9833**	**0.9831**	**0.9833**	**0.9833**	**0.9833**	0.9831
5	0.9553	**0.9562**	0.9553	0.9553	0.9553	0.9562
10	**0.9218**	**0.9218**	0.9119	0.912	**0.9218**	0.9217
20	**0.8747**	**0.8747**	0.8693	0.874	**0.8747**	0.8745
30	**0.8425**	**0.8424**	0.8311	0.8405	**0.8424**	0.8421
40	**0.8216**	**0.8218**	0.8008	0.8158	**0.8218**	0.8179
60	**0.7848**	**0.7845**	0.7483	0.7675	**0.7845**	0.7804
80	**0.7525**	**0.7525**	0.7023	0.7217	0.7217	0.7486
90	0.735	0.7353	0.6884	0.7073	0.7073	0.7358
100	0.7317	0.7316	0.6634	0.6924	0.6924	0.7346
Shoe						
2	0.9954	**0.9957**	0.9954	0.9954	0.9954	0.9956
5	0.988	**0.9898**	0.988	0.988	0.988	0.9896
10	**0.9793**	**0.9789**	0.9702	0.9701	**0.9789**	0.9775
20	**0.9509**	**0.9495**	**0.9477**	**0.9493**	**0.9495**	0.9434
30	**0.8767**	**0.8745**	**0.8651**	**0.871**	**0.8745**	0.8625
40	0.7554	0.7539	0.7402	0.7477	0.7539	0.8311
60	0.4638	0.4638	0.5044	0.4905	0.4638	0.5407
80	0.3988	0.3988	**0.4319**	**0.4212**	**0.4212**	0.4144
90	0.3753	0.3753	**0.4125**	**0.4028**	**0.4028**	0.3827
100	0.3612	0.3612	**0.4012**	**0.3911**	**0.3911**	0.3633

Appendix C.
Subjective comparison of Boat and Lena

(a)   (b)

(c)   (d)

Figure C-1: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Boat at noise level $\sigma = 20$. (a) Noise free Boat image. (b) Noisy image with $\sigma = 20$. (c) Denoised image using BM3D, PSNR = 30.6902, SSIM = 0.8215. (d) Denoised image using AEG-BM3D, PSNR = 30.6849, SSIM = 0.8218.

(a)   (b)

(c)   (d)

Figure C-2: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Boat at noise level $\sigma = 30$. (a) Noise free Boat image. (b) Noisy image with $\sigma = 30$. (c) Denoised image using BM3D, PSNR = 28.8529, SSIM = 0.7731. (d) Denoised image using AEG-BM3D, PSNR = 28.8510, SSIM = 0.7736.

(a)   (b)

(c)   (d)

Figure C-3: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Boat at noise level $\sigma = 40$. (a) Noise free Boat image. (b) Noisy image with $\sigma = 40$. (c) Denoised image using BM3D, PSNR = 27.3769, SSIM = 0.7256. (d) Denoised image using AEG-BM3D, PSNR = 27.4322, SSIM = 0.7282.

(a)   (b)

(c)   (d)

Figure C-4: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Boat at noise level $\sigma = 60$. (a) Noise free Boat image. (b) Noisy image with $\sigma = 60$. (c) Denoised image using BM3D, PSNR = 25.3550, SSIM = 0.6583. (d) Denoised image using AEG-BM3D, PSNR = 25.3935, SSIM = 0.6606.

(a)

(b)

(c)

(d)

Figure C-5: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Boat at noise level $\sigma = 80$. (a) Noise free Boat image. (b) Noisy image with $\sigma = 80$. (c) Denoised image using BM3D, PSNR = 23.6621, SSIM = 0.6049. (d) Denoised image using AEG-BM3D, PSNR = 23.9182, SSIM = 0.6006.

(a)

(b)

(c)

(d)

Figure C-6: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Boat at noise level $\boldsymbol{\sigma} = \mathbf{90}$. (a) Noise free Boat image. (b) Noisy image with $\boldsymbol{\sigma} = \mathbf{90}$. (c) Denoised image using BM3D, PSNR = 22.8584, SSIM = 0.5771. (d) Denoised image using AEG-BM3D, PSNR = 23.3414, SSIM = 0.5762.

(a)   (b)

(c)   (d)

Figure C-7: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Boat at noise level $\sigma = 100$. (a) Noise free Boat image. (b) Noisy image with $\sigma = 100$. (c) Denoised image using BM3D, PSNR = 22.175, SSIM = 0.5563. (d) Denoised image using AEG-BM3D, PSNR = 22.8964, SSIM = 0.5625.

(a)   (b)

(c)   (d)

Figure C-8: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Lena at noise level $\sigma = 20$. (a) Noise free Lena image. (b) Noisy image with $\sigma = 20$. (c) Denoised image using BM3D, PSNR = 32.9968, SSIM = 0.8749. (d) Denoised image using AEG-BM3D, PSNR = 33.0008, SSIM = 0.8752.

(a)

(b)

(c)

(d)

Figure C-9: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Lena at noise level $\sigma = 30$. (a) Noise free Lena image. (b) Noisy image with $\sigma = 30$. (c) Denoised image using BM3D, PSNR = 31.1574, SSIM = 0.8410. (d) Denoised image using AEG-BM3D, PSNR = 31.1758, SSIM = 0.8416.

(a)

(b)

(c)

(d)

Figure C-10: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Lena at noise level $\sigma = 40$. (a) Noise free Lena image. (b) Noisy image with $\sigma = 40$. (c) Denoised image using BM3D, PSNR = 29.7966, SSIM = 0.8083. (d) Denoised image using AEG-BM3D, PSNR = 29.8042, SSIM = 0.8117.

(a)   (b)

(c)   (d)

Figure C-11: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Lena at noise level $\sigma = 60$. (a) Noise free Lena image. (b) Noisy image with $\sigma = 60$. (c) Denoised image using BM3D, PSNR = 27.7087, SSIM = 0.7699. (d) Denoised image using AEG-BM3D, PSNR = 27.7289, SSIM = 0.7729.

Figure C-12: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Lena at noise level $\sigma = 80$. (a) Noise free Lena image. (b) Noisy image with $\sigma = 80$. (c) Denoised image using BM3D, PSNR = 25.5006, SSIM = 0.7344. (d) Denoised image using AEG-BM3D, PSNR = 25.9448, SSIM = 0.7036.

(a)

(b)

(c)

(d)

Figure C-13: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Lena at noise level $\boldsymbol{\sigma} = \mathbf{90}$. (a) Noise free Lena image. (b) Noisy image with $\boldsymbol{\sigma} = \mathbf{90}$. (c) Denoised image using BM3D, PSNR = 24.5864, SSIM = 0.7259. (d) Denoised image using AEG-BM3D, PSNR = 25.3037, SSIM = 0.6933.

(a)

(b)

(c)

(d)

Figure C-14: Subjective comparison of denoising performance between BM3D and AEG-BM3D for standard test image Lena at noise level $\sigma = 100$. (a) Noise free Lena image. (b) Noisy image with $\sigma = 100$. (c) Denoised image using BM3D, PSNR = 23.7521, SSIM = 0.7134. (d) Denoised image using AEG-BM3D, PSNR = 24.673, SSIM = 0.6758.

Curriculum Vitae

Name: Mohammad Mahedi Hasan

Post-secondary Bachelor of Science in Computer Science and Engineering
Education and University of Dhaka
Degrees: Dhaka, Bangladesh
2002-2006

Honors and Western University Graduate Research Scholarship
Awards: 2012-2013

Graduate Research Fellowship at Centre for Logic and Information,
Graduate Student Assistantship at St. Francis Xavier University
2011-2012

Related Work Teaching Assistant
Experience Western University
2012-2013

Teaching Assistant
St. Francis Xavier University
2011-2012