RECURSIVE NON-LOCAL MEANS FILTER FOR VIDEO DENOISING

Thesis

Submitted to

The School of Engineering of the

UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree of

Master of Science in Electrical Engineering

By

Redha Almahdi

UNIVERSITY OF DAYTON

Dayton, Ohio

December, 2016

RECURSIVE NON-LOCAL MEANS FILTER FOR VIDEO DENOISING

Name: Almahdi, Redha

APPROVED BY:

Russell C. Hardie, Ph.D.
Advisor Committee Chairman
School of Engineering, Department of
Electrical and Computer Engineering

Vijayan Asari, Ph.D.
Committee Member
School of Engineering, Department of
Electrical and Computer Engineering

John S. Loomis, Ph.D.
Committee Member
School of Engineering, Department of
Electrical and Computer Engineering

Robert J. Wilkens, Ph.D., P. E.
Associate Dean for Research and Innovation
Professor
School of Engineering

Eddy M. Rojas, Ph.D., M.A., P. E.
Dean, School of Engineering

ABSTRACT

RECURSIVE NON-LOCAL MEANS FILTER FOR VIDEO DENOISING

Name: Almahdi, Redha
University of Dayton

Advisor: Dr. Russell C. Hardie

In this thesis, we propose a computationally efficient algorithm for video denoising that exploits temporal and spatial redundancy. The proposed method is based on Non-Local Means (NLM). NLM methods have been applied successfully in various image denoising applications. In the single-frame NLM method, each output pixel is formed as a weighted sum of the center pixels of neighboring patched, within a given search window. The weights are based on the patch intensity vector distances. The process requires computing vector distances for all of the patches in the search window. Direct extension of this method from 2D to 3D, for video processing, can be computationally demanding. Note that the size of a 3D search window is the size of the 2D search window multiplied by the number of frames being used to form the output. Exploiting a large number of frames in this manner can be prohibitive for real-time video processing. Here we propose a novel Recursive NLM (RNLM) algorithm for video processing. Our RNLM method takes advantage of recursion for copmutationally savings, compared with the direct 3D NLM. However, like the 3D NLM, our method is still able to exploit both spatial and temporal redundancy for improved performance, compared with 2D NLM. In our approach, the first frame is processed with single-frame NLM. Subsequent frames are estimated using a weighted sum of pixels from the current-frame and a pixel from the previous

frame estimate. Only the single best matching patch from the previous estimate is incorporated into the current estimate. Several experimental results are presented here to demonstrate the efficacy of our proposed method in terms of quantitative and subjective image quality, as well as processing speed.

To my loving parents, my brothers and my sisters

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to my advisor Prof.Russell C. Hardie for his continuous support and willingness that allowed me to pursue my M.S. study and research on topics for which I am truly passionate. His tremendous academic support, and insightful guidance have helped me in all the time of algorithm development and writing this thesis document. I could not have imagined having better advisor, and mentors for my M.S. study.

I would also like to express my sincere appreciation and utmost gratitude to Dr. Abdussalam Al Mahjub, the College of Electronic Technology-Bani Walid and the Ministry of Higher Education of Libya for supporting me in my studies and research at the University of Dayton.

In addition, I would like to thank Dr. Khaled Mohamed for his assistance in the initial coding and testing of the RNLM method, and all of the professors who taught me during my studies at the University of Dayton. It was the pleasure to be in each of their classes and to learn from each of them.

Finally, I would like to thank my family: my parents, brothers, and sister for supporting me spiritually throughout writing this thesis and my life in general. A sincere gratitude also goes to everyone who has prayed for my success during my studies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 Objective of Study

During the last four decades, noise reduction for digital image and video sequences is a subject of extensive research, mainly for military and satellite application. Other applications include medical imaging, remote sensing and video broadcasting [1]. Digital image and video, acquired by still cameras, consumer camcorders, or even broadcast quality video cameras, are usually degraded by some amount of blur and noise [2]. Even the new advanced cameras for video acquisition may suffer from severe degradation. Furthermore, old video sequences are likely to have suffered degradation at the early stage of the acquisition mainly due to hard ware and technological limitations [1]. Noise in imaging sensors comes from a variety of sources, the two predominant random noise sources in digital image acquisition modalities are the stochastic nature of the photon-counting at the detectors (Poisson), and the intrinsic thermal and electronic fluctuations of the acquisition devices (Gaussian) [3]. However, the majority of the noise reduction algorithms developed are based on signal frame filtering for image and all 2D signals or three dimensions for video sequences considering both spatial and temporal correlation [1]. One of the main design issues in 3D filter, the size of a 3D search window is the size of the 2D search window multiplied by the number of frames being used to form the output. Typically, the larger the length of the filter, the more computational complexity

significantly increases and delays are introduced. On the other hand, employ a large number of frames to form the output in this manner can be prohibitive for real-time video processing.

1.2 Video Restoration

Digital videos are invariably corrupted by noise during acquisition. Digital video tends to have a lower signal-to-noise ratio (SNR) than static images, due to the short integration times needed to achieve desired frame rates [4]. Low light conditions and small camera apertures tend to worsen the problem. In the case of some medical imagery, like x-ray images and video, short integration times are essential to limit the x-ray dose to the patient. While digital video may suffer from lower SNR, it also provides 3D data that often has significant temporal redundancy [5]. Video denoising algorithms seek to reduce noise by exploiting the both spatial and temporal correlation in the signal [4]. The Non-Local Means (NLM) algorithm [6] for image denoising has received significant attention in the image processing community. This may be, in large part, because of generally good performance, and its intuitive and conceptually simple nature. The standard NLM algorithm is introduced by Buades et al. in [6]. The NLM method exploits self-similarity that appears in most natural images for noise reduction. In the single-frame NLM method, each output pixel is formed as a weighted sum of the center pixels of neighboring patches, within a given search window. The weights are based on similarity with respect to the reference patch. The similarity is measured by means of patch intensity vector distances. Pixels from patches with higher similarity (lower vector distances) are given more weight, using a negative exponential weighting. One or more tuning parameters are used to control the weighting.

Many variations of the NLM method have been proposed to either reduce the computational complexity, and/or improve the denoising performance. In the work of Mahmoudi and Sapiro [7], dissimilar neighborhoods are excluded from the weighted sum. Dissimilar blocks are identified

based on mean value and gradient. This may improve performance, and it reduces the computational cost. Wang et al. [8] proposed an efficient Summed Square Image scheme as another means to accelerate the patch similarity computations. A cluster tree arrangement has been used to group similar patches [9]. A method using preselection of the most similar voxels, multithreading, and blockwise implementation is presented in [10]. Furthermore, adaptive smoothing neighborhoods are presented in [11], and a kernel regression method is presented in [12]. The method in [13] uses a spatially-recursive moving-average-filter to compute the Euclidean distances. The estimation of the mean square error (MSE) from a noisy image is performed using an analytical expression basde on Steins unbiased risk [14]. Another speed enhancement, based on probabilistic early termination, is proposed in [15]. Karnati et al. [16] proposed a multi-resolution approach requiring fewer comparisons.

Many methods, originally proposed for single image denoising, have been adapted to video denoising. Among these are the NLM method, which has been applied successfully to image sequences in [17] and [18]. Han et al. [19] introduced the Dynamic Non-local means (DNLM) video denoising method, which is based on Kalman filtering theory. The basic idea of this filter is to use information from the past video frames to restore the current frame, combining the NLM and Kalman filtering algorithms. However, the computational complexity is still relatively high with this method. Another example of a 2D denoising method, later extended to 3D, is Block Matching and 3D (BM3D) filtering [20]. BM3D generally outperforms NLM in single image denoising, but has a higher computational complexity. The BM3D method, like NLM, uses vector distances between 2-D image blocks. The most similar blocks are stacked into a 3-D group, and then filtered through a transform-domain shrinkage operation. The BM3D filtering has been extended to video denoising (V-BM3D) in [21]. While there may be many variations of patch based image denoising algorithms. What they share in common is that they require computing vector distances between each reference

patch and neighboring patches. Direct extension of such methods from 2D to 3D, for video processing, can be computationally demanding. Note that the size of a 3D search window is the size of the 2D search window multiplied by the number of frames being used to form the output. Exploiting a large number of frames in this manner can be prohibitive for real-time video processing.

1.3 Significance of the Study

In this thesis, we propose a novel temporally-recursive NLM (RNLM) algorithm for video processing. Our RNLM method takes advantage of temporal recursion for computational savings, compared with the direct 3D NLM. However, like the 3D NLM, our method is still able to exploit both spatial and temporal redundancy for improved performance, compared with 2D NLM. In our approach, the first frame is processed with single-frame NLM. Subsequent frames are estimated using a weighted sum of pixels from the current-frame and a pixel from the previous frame estimate. Only the best matching patch from the previous estimate is incorporated into the current estimate. This is done to maximize the temporal correlation. Our approach shares its recursive nature with DNLM in [19]. However, here we have opted for a much simpler framework, in keeping with the simplicity of the original NLM. Several experimental results are presented here to demonstrate the efficacy of our proposed method in terms of quantitative and subjective image quality, as well as processing speed. We show that our approach offers a computationally simple approach to video denoising, but with a performance that rivals much more complex methods.

1.4 Organization of the Thesis

The remainder of the thesis is organized as follows. The observation model considered in the practical work of the study, the Overview of Non-Local Means algorithm itself , and the direct extension of the Non-Local Means algorithm from 2D to 3D for video processing presented in Chapter II. The recursive Non-Local means (RNLM) algorithm and computational complexity are

proposed and discussed in Chapter III. We show some experimental results in Chapter IV. Finally, conclusions are provided in Chapter V.

CHAPTER II

NON-LOCAL MEANS METHOD

In this chapter we introduce the Non-Local Means (NLM) method. We begin with the observation model for the proposed algorithm development in Chapter 2.1 and then We provide some of the key algorithm details and definition in chapter 2.2. Also, we provide the direct extension of this algorithm from 2D to 3D for video processing in Chapter 2.3.

2.1 Observation Model

In this section, we present the observation model and notation for our video restoration methods. Some of the key variables used in this paper are defined in Table 2.1. We use the standard degradation model, treating the noise as additive and signal-independent. This is expressed as

$$y_k(i) = x_k(i) + n_k(i), \tag{2.1}$$

for $i = 1, 2, ..., N$ and $k = 1, 2, ..., K$. Note that $y_k(i)$ represents a pixel in the observed frame in the video sequence. The index i refers to the specific pixel in the spatial domain, and the k denotes the temporal frame number in the image sequence. The variable $x_k(i)$ denotes the corresponding pixel in the ideal input frame. The noise is represented by $n_k(i) \sim \mathcal{N}(0, \sigma_n^2)$, which is assumed to be samples of a zero-mean independent and identically distributed Gaussian random variable, with variance σ_n^2.

Table 2.1: Variable definitions.

$x_k(i)$	Ideal pixel i in frame k
$y_k(i)$	Noisy pixel i in frame k
$\hat{x}_k(i)$	Estimated pixel i in frame k
$\mathbf{y}_k(i)$	Lexicographical patch about pixel i in frame k in $\{y_k(\cdot)\}$
N	Number of pixels in one frame
K	Number of frames in input sequence
L	Number of frames used by 3D NLM to generate one frame output
M_s	NLM search window dimension ($M_s \times M_s$)
M_p	NLM patch dimension ($M_p \times M_p$)
N_s	BMA search window dimension ($N_s \times N_s$)
N_b	BMA block dimension ($N_b \times N_b$)

2.2 Single Frame Non-Local Means Filter

The non-local means algorithm we shall now discuss is one of the most popular and attractive algorithms for image restoration. It performs denoising in the spatial domain and removes the noise while retaining the important image features such as for preserve edges and details. For this purpose, the NLM take advantage of the high degree of redundancy and self-similarity of the natural image structure. The denoising method based on assuming that every small patch in a natural image has many similar patches in the same image. Figure 2.1 illustrates these properties on a natural image. The image House with a chosen search window $M_s \times M_s$ marked in Black, and each colored squares corresponds to respective similarity patches. The reference patch $\mathbf{y_k(i)}$ is marked in blue, and the index i is a positional of the central pixel of the reference patch while the green squares $\mathbf{y_k(1)}$, $\mathbf{y_k(2)}$, $\mathbf{y_k(3)}$, and $\mathbf{y_k(4)}$ corresponds to a set of very similar patches to the reference patch. On the other side, the orange square $\mathbf{y_k(5)}$ is the patch that having a small similarity pixel values compared to the reference patch. The red square $\mathbf{y_k(6)}$ along with the patch that has high dissimilarity pixel values compared to the reference patch.

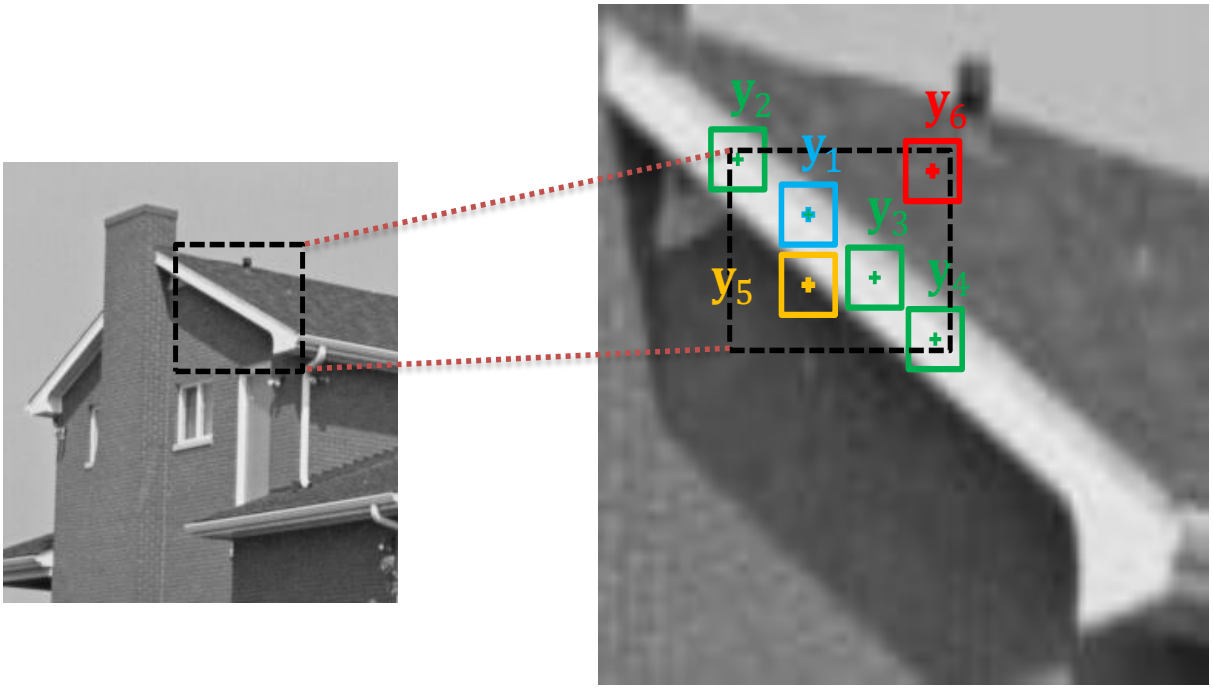


Figure 2.1: The Non-Local Means scheme for a schematic view of a chosen search window and its respective similarity patches related to reference patch.

The main idea of NLM filter is the weight computation depend on the similarity of the intensity grey-level vectors between the similar patches in the search region. The patches with a similar grey-level neighborhood to the reference patch $\mathbf{y_k(i)}$ will have larger weights on the average. For example, the central pixels of patches $\mathbf{y_k(1)}$, $\mathbf{y_k(2)}$, $\mathbf{y_k(3)}$, and $\mathbf{y_k(4)}$ will assign weights larger than the central pixel of patches $\mathbf{y_k(5)}$ and $\mathbf{y_k(6)}$ because they have high similarity compared to the reference patch $\mathbf{y_k(i)}$. The weight computation can be expressed as in Equation (2.3) and Equation (2.4).

We defining the single-frame NLM (SNLM) [6], upon which our method is built. Processing the frames from an image sequence individually, the SNLM output can be expressed as

$$\hat{x}_k(i) = \frac{1}{W_{k,i}} \sum_{j \in \varepsilon(i)} w_k(i,j) y_k(j), \tag{2.2}$$

where $\hat{x}_k(i)$ denotes the estimated image at pixel i in frame k. The set $\varepsilon(i)$ contains the indices of the pixels within an $M_s \times M_s$ search window centered about pixel i. The variable $w_k(i,j)$ is the weight applied to pixel j, when estimating pixel i in frame k. To normalize the weights, the variable $W_{k,i}$ is used, and this is simply the sum of the individual weights.

The SNLM weights are computed based on patch similarity and spatial proximity. In particular, $w_k(i,j)$ is computed as

$$w_k(i,j) = \exp\left\{-\frac{\|\mathbf{y}_k(i) - \mathbf{y}_k(j)\|^2}{2\sigma_y^2} - \frac{d(i,j)^2}{2\sigma_d^2}\right\}, \tag{2.3}$$

and $W_{k,i}$ can be expressed as

$$W_{k,i} = \sum_{j \in \varepsilon(i)} w_k(i,j). \tag{2.4}$$

Note that the variable $\mathbf{y}_k(i)$ is a vector in lexicographical form containing pixels from an $M_p \times M_p$ patch centered about pixel i in frame k from the sequence $\{y_k(\cdot)\}$. The variable $d(i,j)^2$ is the squared Euclidean distance between pixel i and j. The parameter σ_y^2 is a tuning parameter to control the decay of the exponential weight function with regard to patch similarity, and σ_d^2 is a tuning parameter controlling the decay with regard to spatial proximity between pixels i and j. It can seen from Equation (2.3) that the weight given to pixel $y_k(j)$ goes down as $\|\mathbf{y}_k(i) - \mathbf{y}_k(j)\|^2$ goes up. The weight also goes down with the spatial distance between pixel i and j.

From Equation (2.2) and (2.3) imply that NLM filter rely on tuning parameters. The NLM patch dimension $M_p \times M_p$, which is set to 5×5 or 7×7 for the most part to produce the best results. However, the patch size should be carefully chosen depending on the local scale of the image; a large patch size is more suitable for smooth areas in the image, and a small patch size is better for textured areas. In fact, employing a too large patch in textured areas prevents the algorithm from finding redundancies. The weight smoothing parameter σ_y^2, is often set close to the noise variance. Studies of filter parameter selection can be found in [6, 22, 23, 24]. The size of the search window $M_s \times M_s$, that has a large impact on the computation time and the overall complexity of

the algorithm. On the other hand, the large search window size helps to find more similar patches related to the reference patch. While, the small search window size leads to difficulty to find enough similar patches which also has an influence on the visual quality of the results.

2.3 3-D Non-Local Means Filter

In this section, we introduce a direct extension of 2D SNLM to 3D, to use as an additional performance benchmark. The 3D NLM uses a spatio-temporal search window to provide improved video denoising. In our approach, the patches remain 2D, but the search window is extended to 3D. This version of the 3D NLM is given by

$$\hat{x}_k(i) = \frac{1}{W_{k,i}} \sum_{j \in \varepsilon(i)} \sum_{m \in \psi(k)} w_{k,m}(i,j) y_m(j), \tag{2.5}$$

where $w_{k,m}(i,j)$ is the weight for pixel j of frame m, when estimating pixel i of frame k. The temporal search window is defined by $\psi(k)$, which is the set of frame indices used in the 3D search window for the estimation of frame k. The temporal window comprised of the most recent L frames, and this is represented as $\psi(k) = \{k, k-1, ..., k-L+1\}$.

The 3D NLM weights are computed as

$$w_{k,m}(i,j) = \exp\left\{ -\frac{\|\mathbf{y}_k(i) - \mathbf{y}_m(j)\|^2}{2\sigma_y^2} - \frac{d(i,j)^2}{2\sigma_d^2} - \frac{(k-m)^2}{2\sigma_t^2} \right\}, \tag{2.6}$$

where a new temporal proximity tuning parameter, σ_t^2 is introduced. This extra tuning parameter is a natural extension to the spatial proximity parameter σ_d^2. The weights are normalized using

$$W_{k,i} = \sum_{j \in \varepsilon(i)} \sum_{m \in \psi(k)} w_{k,m}(i,j). \tag{2.7}$$

Other similar extensions of the SNLM to 3D can be found in [25, 26].

CHAPTER III

METHOD

3.1 The Proposed RNLM Video Denoising Algorithm Definition

The goal of the proposed RNLM method is to effectively exploit spatio-tempotal information, as is done with the 3D NLM in Equation (2.5), but with a computational complexity more in line with the SNLM in Equation (2.2). To do so, RNLM estimate is formed as a weighted sum of pixels from the current frame, like Equation (2.2), but it also includes a previous frame pixel estimate. That is, the current input frame and the prior output frame are used to form the current output. This type of temporal recursive processing helps to exploit the temporal signal correlation, without significantly increasing the search window size or overall computational complexity.

Specifically, the estimate for the proposed RNLM is given by

$$\hat{x}_k(i) = \frac{1}{W_{k,i}} \left[w_{\hat{x},k}(i)\hat{x}_{k-1}(s_k(i)) + \sum_{j \in \varepsilon(i)} w_{y,k}(i,j) y_k(j) \right], \tag{3.1}$$

where $\hat{x}_{k-1}(s_k(i))$ is the previous frame estimate (i.e., frame $k-1$) at pixel $s_k(i) \in \{1, 2, ..., N\}$. Pixel $s_k(i)$ is selected from $\{\hat{x}_{k-1}(\cdot)\}$ based on block matching with respect to the block in input frame k centered about pixel i. For the selection of $s_k(i)$, we allow for a potentially different block and search size from that used for the within-frame processing. In particular, the block-matching block size is $N_b \times N_b$, with an $N_s \times N_s$ search window. As in Equation (2.2) for the SNLM, the

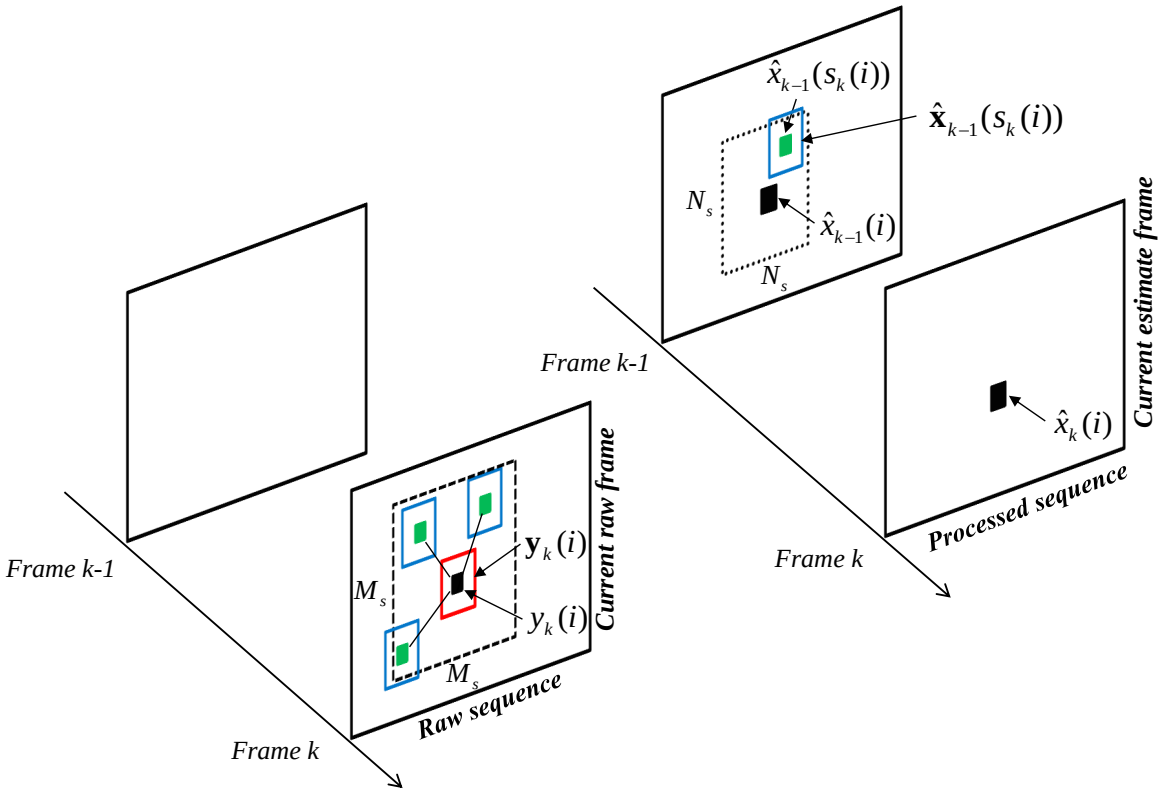


Figure 3.1: Block Diagram of Proposed Algorithm.

set $\varepsilon(i)$ in Equation (3.1) contains the indices of the pixels within an $M_s \times M_s$ search window centered about pixel i. The recursive weight in Equation (3.1) is $w_{x,k}(i)$, and the non-recursive weights, $w_{y,k}(i,j)$, are similar to that for the SNLM. We shall define and discuss all of the weights shortly. The relationship among the various pixels used in the RNLM estimation process is depicted in Figure 3.1. Shown are the raw unprocessed frames in parallel with the processed frames. Output $\hat{x}_k(i)$ is formed using a weighted sum of the input frames pixels, shown on the left, and the best matching previous processed output, shown on the back right.

Let us now define the weights. The non-recursive weights are defined in a manner similar to SNLM. Specifically, these are given by

$$w_{y,k}(i,j) = \exp\left\{-\frac{\|\mathbf{y}_k(i) - \mathbf{y}_k(j)\|^2}{h_{y_b}} - \frac{\sigma_n^2}{h_{y_n}}\right\}, \tag{3.2}$$

where h_{y_b} and h_{y_n} are tuning parameters. Here, we do not use the spatial distance weighting term of the SNLM. This could easily be added, but it did not provide improved performance in out experimental results. The recursive weights are given by

$$w_{\hat{x},k}(i) = \exp\left\{-\frac{\|\mathbf{y}_k(i) - \hat{\mathbf{x}}_{k-1}(s_k(i))\|^2}{h_{\hat{x}_b}} - \frac{\sigma^2_{\hat{x}_{k-1}(s_k(i))}}{h_{\hat{x}_n}}\right\}, \tag{3.3}$$

where $h_{\hat{x}_b}$ and $h_{\hat{x}_n}$ are tuning parameters, and $\sigma^2_{\hat{x}_{k-1}(s_k(i))}$ is the residual noise variance associated with $\hat{x}_{k-1}(s_k(i))$. The vector $\hat{\mathbf{x}}_{k-1}(s_k(i))$ is the $M_s \times M_s$ patch of pixels about pixel $\hat{x}_{k-1}(s_k(i))$ (shown in Figure 3.1) in lexicographical vector form. The weight normalization factor here is given by

$$W_{k,i} = w_{\hat{x},k}(i) + \sum_{j \in \varepsilon_y(i)} w_{y,k}(i,j). \tag{3.4}$$

Finally, assuming the noise is independent and identically distributed, the residual noise variance can be computed recursively as follows

$$\sigma^2_{\hat{x}_k}(i) = \frac{w^2_{\hat{x},k}(i)\sigma^2_{\hat{x}_{k-1}}(s_k(i)) + \sum_{j \in \varepsilon_y(i)} w^2_{y,k}(i,j)\sigma_n^2}{W^2_{k,i}}. \tag{3.5}$$

Note that Equation (3.5) is not the error variance. Rather it only accounts for the variance of the noise component of the error associated with the estimate $\hat{x}_{k-1}(s_k(i))$.

The RNLM weights in Equations (3.2) and (3.3) have a total of four tuning parameters, two to govern the non-recursive weights, and two to govern the recursive weights. In the non-recursive weights in Equation (3.2), the parameter h_{y_b} serves the same role as σ_y^2 in the SNLM in Equation

(2.3). We refer to this weight as the non-recursive bias error weight. We view $||\mathbf{y}_k(i) - \mathbf{y}_k(j)||^2$ as a measure of the bias error in $y_k(j)$ with respect to the true sample $x_k(i)$ that we are estimating. That is, underlying signal differences between the pixel i and j are being quantified by this term. The tuning parameter h_{y_b} controls the exponential decay of the weights as a function of this bias error. The noise error associated with $y_k(j)$ is given by the constant noise variance σ_n^2. The noise variance in Equation (3.2) is scaled by the h_{y_n}, to control the weight decay as a function of the noise variance. For the recursive weight, we have similar tuning parameters. The term $||\mathbf{y}_k(i) - \hat{\mathbf{x}}_{k-1}(s_k(i))||^2$ quantifies the bias error associated with $\hat{x}_{k-1}(s_k(i))$, and the residual noise variance associated with this estimate is $\sigma_{\hat{x}^2_{k-1}(s_k(i))}$, and may be computed using Equation(3.5). We give each of these error quantities a tuning parameter, to balance their impact on the resulting filter weights. The bias error for the recursive sample is scaled by $h_{\hat{x}_b}$, and the residual noise term is scaled by $h_{\hat{x}_n}$.

We acknowledge that the proposed RNLM does not have the optimal framework of the Kalman filter [19]. However, it can exploit temporal signal correlation effectively, as we shall show in Section 4.1. By choosing the tuning parameters to balance the bias and noise components of the recursive and non-recursive terms, very good performance can be achieved with a low computational complexity. We believe that the RNLM may provide useful solution for many video denoising applications, and we believe it is in keeping with the spirit and simplicity of the original NLM method.

3.2 Computational Complexity

In this section, we compare the computational complexity of SNLM, 3D NLM, and RNLM by counting the number of multiplications and additions required to compute one processed output pixel. Beginning with the SNLM, the number of floating point multiplies and adds to compute

Equation (2.2) is M_s^2 . Computing the weights based on Equation (2.3) requires M_s^2 vector distances, for vectors of size $M_p^2 \times 1$. This requires approximately $2M_s^2M_p^2$ additions/subtractions, and $M_s^2M_p^2$ multiplications. Another M_s^2 adds and multiplies is needed to compute and apply the weight normlaization in Equation (2.4).

The 3D NLM algorithm requires LM_s^2 floating point multiplies and adds to compute Equation (2.5). The weights from Equation (2.6) require LM_s^2 vector distances using $M_p^2 \times 1$ vectors. This requires approximately $2LM_s^2M_p^2$ additions/subtractions, and $LM_s^2M_p^2$ multiplications. Another LM_s^2 adds and multiplies is needed to compute and apply the weight normlaization in Equation (2.7). Thus, the complexity of the 3D NLM algorithm increases linearly with the number of frames, L.

The process of the RNLM filter is accomplished in several steps. The output in Equation (3.1) requires $M_s^2 + 1$ floating point multiplies and adds. The computation of the weights in Equations (3.2) and (3.3) requires $M_s^2 + 1$ vector distances using $M_p^2 \times 1$ vectors. This requires approximately $2(M_s^2 + 1)M_p^2$ additions/subtractions, and $2M_s^2M_p^2$ multiplications. The residual noise recursion in Equation (3.5) requires $M_s^2 + 1$ floating point multiplies and adds. Finally, if BMA is used to find $s_k(i)$ in Equation (3.1), this requires N_s^2 vector distances with $N_b^2 \times 1$ vectors. This requires approximately $2N_s^2N_b^2$ additions/subtractions, and $N_s^2N_b^2$ multiplications. The weight normalization operation is accomplished with $M_s^2 + 1$ floating point adds to compute Equation (3.4).

Note that if we let $N_s = M_s$ and $N_b = M_p$, then RNLM has a computational complexity comparable to 3DNLM for $L = 2$ frames. Also, if we simply let $s_k(i) = i$ (i.e., No BMA option), the RNLM has a computational complexity that is approximately the same a SNLM. However, we have found that the BMA matching significantly improves perfromance in video sequences with significant amounts of motion. Furthermore, we have found that it is generally advantageous to choose $N_b > M_p$. This helps to provide a better match for the the important sample $\hat{x}_{k-1}(s_k(i))$

. The size of the search window N_s maybe selected based on the temporal motion expected in the video sequence.

CHAPTER IV

EXPERIMENTAL RESULTS

4.1 Simulated Data and Discussion

In order to illustatrate the efficacy of the proposed RNLM algorithm, we present a number of experimental results. We compare our method to several state-of-the-art methods including SNLM[6], 3D NLM, BM3D [20], V-BM3D [21], and DNLM [19]. Our results make use of standard and publicly available video sequences [27], allowing for reproducable and compariative results.

These sequences present variations in scene content, lighting conditions, and scene motion. We artifically degrade the imagery with Gaussian noise and compare the restored images to the originals.

The metrics we shall use are the peak signal-to-noise ratio (PSNR), and the structural similarity (SSIM). The PSNR metric in units of deciBels (dB) is defined as

$$PSNR(k) = 10 \times \log_{10}\left(\frac{R^2}{MSE(k)}\right), \tag{4.1}$$

where R is the maximum limit of the dynamic range of the image. For our 8 bit images, $R = 255$. The variable, $MSE(k)$, is the mean squared error for frame k, given by

$$MSE(k) = \frac{1}{N}\sum_{i=1}^{N}(x_k(i) - \hat{x}_k(i))^2, \tag{4.2}$$

where $x_k(i)$ is the true pixel value, and $\hat{x}_k(i)$ is the estimated pixel.

The SSIM provides an aditional metric that some argue is more consistent with subjective perception than PSNR [28, 29, 30].

In Table 4.1, we provide PSNR results for 5 different image seqeunces, each with 8 different noise standard deviations. The proposed RNLM method results are reported for two variations. The results labled RNLM (BMA) refers to the proposed method where block matching is employed to find the best match for the recursive sample $\hat{x}_{k-1}(s_k(i))$. The (No BMA) version simply uses the estimate pixel position, such that $s_k(i) = i$. This version has a reduced computational complexity. However, as shown in Table 4.1, using BMA gives improved results, compared with the (No BMA) version.

The results in Table 4.1 show that the RNLM method consistently outperforms the SNLM. Thus, the recursive processing is providing a clear performance benefit. RNLM also outperforms single frame BM3D. V-BM3D does provide higher PSNR values in most cases, but the computational complexity of that method is far greater, and the processing of the video is non-causal. On the other hand, RNLM has a low computational complexity, and the processing is fully causal, allowing for real-time processing. SSIM results for the same sequences are shown in Table 4.2. In this table, several additional published results are reported for comparison. These include WRSTF [31], SEQWT [32], 3DWTF [33], IFSM [34], 3DSWDCT [34], VBM3D [21], ST-GSM [35], and DNLM [19]. The results for these methods are the reported results from the respective papers, for the same sequences and noise levels. The standard deviations of the additive Gaussian noise are 10, 15 and 20. It can be seen that the RNLM method gives higher SSIM than SNLM, WRSTF, SEQWT, 3DWTF, IFSM, and DNLM, with average SSIM gains of 0.070, 0.018, 0.059, 0.034, 0.063, and 0.0091, respectively. The SSIM of RNLM is competitive with 3DSWDCT and ST-GSM. However, V-BM3D gives the best results here.

Table 4.1: PSNR comparison with competitive denoising algorithm.

σ_n	Video: / Res.: / Frames:	Salesman 288 x 352 50	Tennis 240 X 352 150	Fl. Gard. 240 X 352 150	Miss Am. 288 X 360 150	Foreman 288 X 352 300	Overall PSNR
5	SNLM	37.14	36.49	36.50	42.08	38.43	38.13
	BM3D	38.32	37.54	37.02	43.09	38.83	38.96
	RNLM (No BMA)	39.36	37.80	36.46	42.21	39.95	39.16
	RNLM (BMA)	39.59	38.61	37.09	42.74	40.55	39.71
	V-BM3D	41.35	40.03	37.40	43.98	39.91	40.53
10	SNLM	33.28	32.66	31.03	39.23	35.17	34.27
	BM3D	34.51	33.63	31.85	40.39	35.21	35.12
	RNLM (No BMA)	36.14	34.04	31.05	39.66	36.13	35.41
	RNLM (BMA)	36.31	34.40	32.08	40.22	36.72	35.95
	V-BM3D	38.33	36.37	32.58	42.17	36.67	37.22
15	SNLM	30.92	30.57	28.16	37.22	33.16	32.01
	BM3D	32.43	31.65	28.99	38.61	33.26	32.99
	RNLM (No BMA)	33.16	31.96	28.50	37.66	34.20	33.08
	RNLM (BMA)	34.30	32.49	29.47	38.53	34.85	33.93
	V-BM3D	36.55	34.35	29.93	40.93	34.87	35.33
20	SNLM	29.38	29.35	26.29	35.69	31.70	30.48
	BM3D	31.02	30.35	27.04	37.22	31.90	31.51
	RNLM (No BMA)	31.44	30.65	26.62	36.61	32.68	31.59
	RNLM (BMA)	32.79	30.93	27.74	37.24	33.37	32.42
	V-BM3D	35.07	32.92	28.06	39.93	33.54	33.91
25	SNLM	28.20	28.52	24.91	34.51	30.48	29.32
	BM3D	29.94	29.38	25.57	36.06	30.85	30.36
	RNLM (No BMA)	30.16	29.26	25.30	35.57	31.08	30.27
	RNLM (BMA)	31.61	29.92	26.57	36.40	32.09	31.32
	V-BM3D	33.69	31.83	26.55	39.04	32.42	32.71
30	SNLM	27.27	27.69	23.83	33.54	29.41	28.35
	BM3D	29.06	28.58	24.39	34.19	29.99	29.24
	RNLM (No BMA)	29.22	28.28	24.23	34.79	30.08	29.32
	RNLM (BMA)	30.64	28.80	25.55	35.70	31.10	30.36
	V-BM3D	32.45	30.96	25.21	37.40	31.42	31.49
35	SNLM	26.53	26.85	22.96	32.72	28.47	27.51
	BM3D	28.28	27.85	23.41	34.19	29.23	28.59
	RNLM (No BMA)	28.48	27.53	23.35	34.05	29.53	28.58
	RNLM (BMA)	29.82	27.99	24.68	35.11	30.27	29.57
	V-BM3D	31.34	30.24	24.28	37.40	30.59	30.77
40	SNLM	25.93	26.07	22.23	32.02	27.65	26.78
	BM3D	27.38	27.04	22.60	33.26	28.30	27.72
	RNLM (No BMA)	27.96	25.74	22.55	33.55	28.83	27.72
	RNLM (BMA)	29.16	27.16	23.96	34.58	29.54	28.87
	V-BM3D	30.32	29.60	23.28	36.33	29.74	29.85

Table 4.2: SSIM comparison with competitive denoising algorithm.

σ_n	Video: Res.: Frames:	Salesman 288 x 352 50	Tennis 240 X 352 150	Fl. Gard. 240 X 352 150	Miss Am. 288 X 360 150	Foreman 288 X 352 300	Overall SSIM
10	SNLM	0.887	0.853	0.934	0.890	0.907	0.894
	BM3D	0.917	0.869	0.963	0.959	0.918	0.925
	WRSTF [31]	0.932	0.897	0.953	0.908	0.927	0.923
	SEQWT [32]	0.900	0.842	0.941	NA	NA	0.894
	3DWTF [33]	0.923	0.856	0.909	NA	NA	0.896
	IFSM [34]	0.904	0.855	0.927	0.904	0.886	0.895
	3DSWDCT [34]	0.955	0.894	0.959	0.946	0.932	0.937
	V-BM3D	0.959	0.916	0.963	0.967	0.934	0.948
	ST-GSM [35]	0.960	0.894	0.950	0.952	0.937	0.939
	DNLM [19]	0.931	0.856	0.947	0.964	0.946	0.928
	RNLM(No BMA)	0.937	0.881	0.953	0.954	0.923	0.930
	RNLM(BMA)	0.939	0.895	0.960	0.960	0.925	0.936
15	SNLM	0.825	0.759	0.886	0.829	0.870	0.834
	BM3D	0.878	0.817	0.937	0.948	0.888	0.893
	WRSTF [31]	0.901	0.839	0.922	0.877	0.877	0.883
	SEQWT [32]	0.846	0.722	0.893	NA	NA	0.820
	3DWTF [33]	0.903	0.793	0.872	NA	NA	0.856
	IFSM [34]	0.851	0.776	0.882	0.857	0.836	0.840
	3DSWDCT [34]	0.930	0.834	0.931	0.928	0.907	0.906
	V-BM3D	0.943	0.874	0.940	0.961	0.911	0.926
	ST-GSM [35]	0.941	0.841	0.925	0.943	0.917	0.913
	DNLM [19]	0.889	0.795	0.906	0.951	0.929	0.894
	RNLM(No BMA)	0.883	0.811	0.900	0.941	0.887	0.884
	RNLM(BMA)	0.902	0.829	0.933	0.945	0.901	0.902
20	SNLM	0.768	0.679	0.836	0.761	0.833	0.775
	BM3D	0.843	0.780	0.909	0.936	0.861	0.866
	WRSTF [31]	0.868	0.790	0.889	0.846	0.873	0.853
	SEQWT [32]	0.796	0.716	0.842	NA	NA	0.785
	3DWTF [33]	0.882	0.740	0.840	NA	NA	0.821
	IFSM [34]	0.801	0.709	0.837	0.812	0.793	0.790
	3DSWDCT [34]	0.905	0.790	0.900	0.909	0.884	0.878
	V-BM3D	0.923	0.836	0.918	0.956	0.891	0.905
	ST-GSM [35]	0.923	0.797	0.900	0.936	0.901	0.891
	DNLM [19]	0.849	0.758	0.865	0.939	0.913	0.865
	RNLM(No BMA)	0.881	0.779	0.881	0.930	0.855	0.865
	RNLM(BMA)	0.886	0.783	0.905	0.937	0.876	0.877

Additional PSNR results are provided in Table 4.3 for two noise levels. The noisy input image PSNRs here are 24 dB and 28 dB. Comparison methods here include STA [26], K-SVD [36], ST-GSM [35], 3D-patch [25], VBM3D [21], DNLM [19], and SNLM. The NLM Patch size is $M_p = 7$. The NLM Search window $M_s = 11$. The BMA search and block-matching size used $N_s = 3$, $N_b = 29$, respectively. As one can see, RNLM (BMA) outperforms STA, DNLM, RNLM (No BMA), and SNLM with the average PSNR gains of 1.35, 0.95, 0.68, and 2.96, respectively.

Figures 4.2, 4.1, and 4.3 show the restored PSNR for individual frames 1 to 150 for the sequences Salesman,Tennis,Miss America, and Flower Garden, respectively. The noise standard deviation for these results is $\sigma_n = 40$. The methods shown are V-BM3D, BM3D, SNLM, RNLM (No BMA), and RNLM (BMA). The proposed RNLM (BMA) provides the best results in the Flower Garden video sequence, and provides the second best performance on the other sequence.

Besides traditional PSNR criteria, we carry out experiments based on SSIM. Figures 4.4, 4.5, and 4.6 show the restored SSIM for individual frames 1 to 150 for the sequences Salesman, Tennis, Miss America, and Flower Garden respectively. The noise standard deviation for these results is $\sigma_n = 15$. The methods shown are V-BM3D, BM3D, SNLM, RNLM (No BMA), and RNLM (BMA). The proposed RNLM (BMA) has competitive performance with the video denoising algorithm V-BM3D in the Flower Garde video sequence and provides the second best performance on the Salesman sequence.

In Fig. 4.9, we offer a visual comparison of denoised Frame 92 from the salesman sequence with Gaussian noise at $\sigma_n = 40$. Figure 4.9(a) shows the original frame. The noisy frame is shown in Figure. 4.9(b). Figure 4.9(c) is the denoised image using SNLM. This method struggles to effectively handle the high levels of noise. However, with decreased σ_n, it tends to exhibit much better subjective visual performance. Figure 4.9(d) is the BM3D denoised frame. While it successfully reduces the noise, here it produces an over-smooth result and detail such as the mouth,

eyes, and texture of the original frame is lost. Figure 4.9(e) shows the V-BM3D denoised frame. This result does a good job with noise reduction and some detail preservation. However, it also tends to over-smooth texture in this image, such as the plant leaves. Finally, Fig. 4.9(f) shows the output of the proposed RNLM (BMA) algorithm. We believe that it produces a visually pleasing result here, with a good balance of noise reduction and detail/texture preservation.

Moreover, In Fig. 4.8, we offer another visual comparison of denoised Frame 48 from the Flower Garden sequence with Gaussian noise at $\sigma_n = 30$. Figure 4.9(a) shows the original frame. The noisy frame is shown in Fig. 4.9(b). Figure 4.9(c) and Figure 4.9(d) are the denoised image using SNLM, and the BM3D denoised frame, respectively. These methods successfully removed the noise, but produces over-smooth results which resulted lose some visual features and details. Figure 4.9(e) shows the V-BM3D denoised frame. This result successfully removed noise even in highly noise level, and produces fine image details. However, it smooth regions produces a few annoying artifacts and lose in texture regions. Finally, Fig. 4.9(f) shows the output of the proposed RNLM (BMA) algorithm. It produces a visually pleasing result which maintains richer texture in the scene and good sharp details. Figures4.10 and Figures4.11 show sample of denoised frame 114 from the Tennis sequence and frame 573 Foreman sequence, respectively. Figure 4.10 (a) and 4.11(a) show the noisy frame. Figure. 4.10(b) and 4.11 (b) show the output of the proposed RNLM (BMA) algorithm. Figure. 4.10(c) and 4.11 (c) show the SSIM index maps of the noisy frame where the darkness indicates the value of the local pixel difference. Figure. 4.10(d) and 4.11 (d) show the SSIM index maps of the output of the proposed RNLM (BMA) algorithm where the brightness indicates the magnitude of the local SSIM index. The brighter means, the better quality regarding underlying image structures measure. The absolute error maps and the SSIM index have been improved so the RNLM (BMA)obtain a better quality regarding image structures measure.

Finally, the current implementation of RNLM is significantly faster than reported processing speeds for DNLM, STA, and 3-D patch Algorithms. The average computational time of RNLM (for Intel 2.2 GHz CPU with 4GB Ram) is 0.065 sec/QCIF frame (176×144 images) for both 5×5 and 3×3 patch size on an Intel 2×2 GHz CPU with BMA motion compensation. The average computational time reported for DNLM is about 200 sec/QCIF frame [19]. In addition, The average computational time for the STA with no motion compensation is 60 sec/frame (384×228 images) [26], whereas RNLM(BMA) takes 0.319 sec/CIF frame at this resolution. Moreover, The 3-D patch algorithm is reported to potentially require days to run a non-optimized MATLAB implementation on a CIF-resolution video [25].The average computational time reported for SNLM 0.287 sec/CIF frame. The BM3D algorithm is slower than SNLM because the full operation of the BM3D algorithm is highly complex and computationally demanding.

Table 4.3: PSNR performances (dB) comparisons with various methods of video denoising algorithms.

Video	Salesman		Fl. Gard.	Miss Am.	Suzie		Foreman	
Input PSNR	28	24	28	28	28	24	28	24
STA [26]	35.13	32.60	31.33	39.39	37.07	35.11	34.94	32.90
K-SVD [36]	37.91	35.59	32.13	40.49	37.96	35.95	37.86	35.86
ST-GSM [35]	37.93	35.17	NA	41.43	38.36	36.21	36.85	34.37
3D-Patch [25]	39.26	36.35	NA	42.23	38.40	36.32	36.88	34.55
VBM3D [21]	38.79	36.07	32.51	41.64	38.16	36.24	37.27	35.19
SNLM [19]	32.97	30.02	30.33	38.47	34.33	31.90	34.92	32.65
DNLM [19]	35.22	32.73	31.28	39.70	37.22	35.25	36.19	34.06
RNLM (No BMA)	36.19	32.78	30.94	39.60	38.43	35.83	36.17	33.87
RNLM (BMA)	36.36	34.03	32.01	40.07	39.36	36.55	36.46	34.37

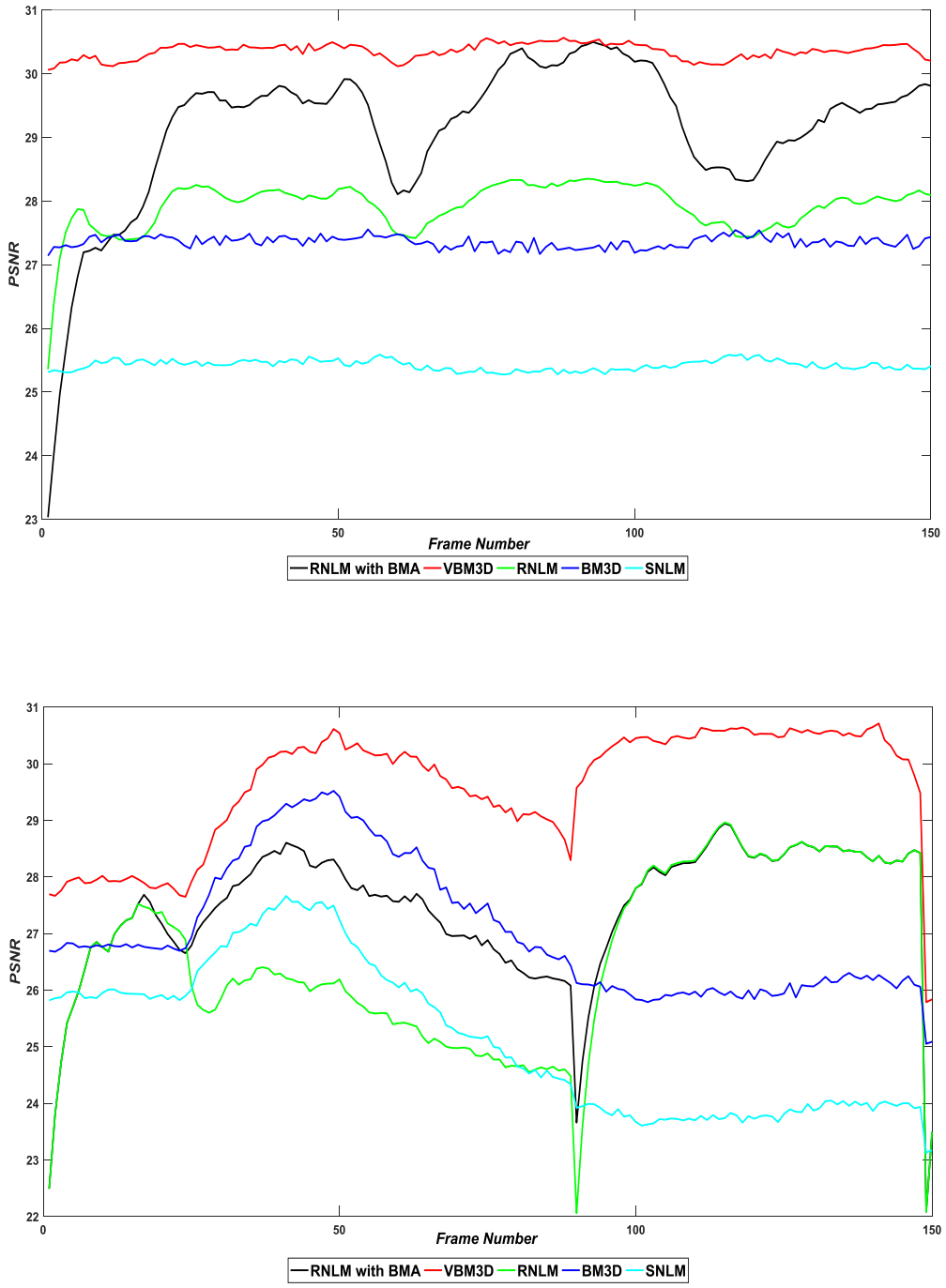


Figure 4.1: Shows the Comparison of PSNR output for two video sequences Salesman(top) and Tennis (bottom) corrupted by Gaussian noise with zero mean and standard deviation = 40 denoised by RNLM,RNLM(No BMA),VBM3D,BM3D, and SNLM

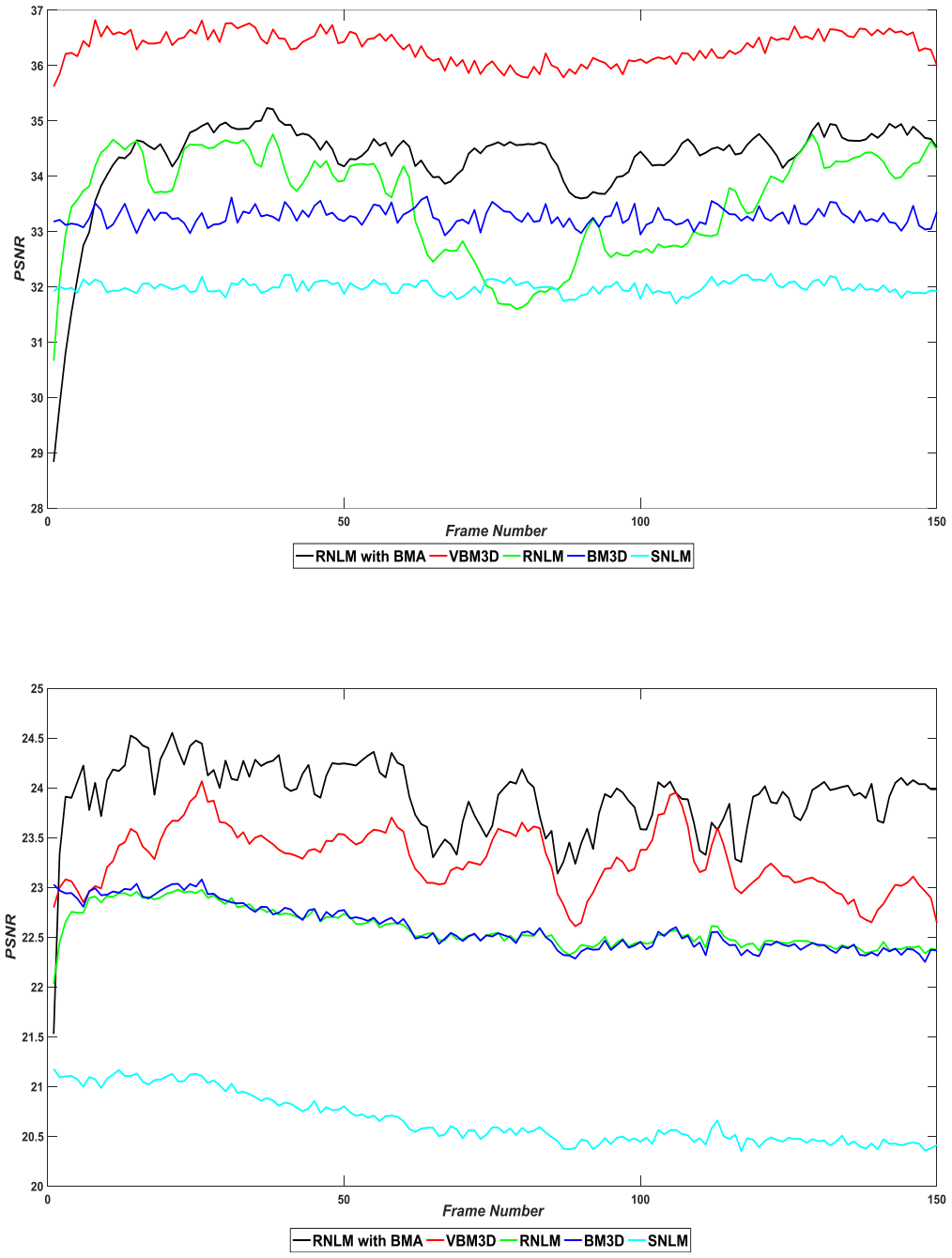


Figure 4.2: Shows the Comparison of PSNR output for two video sequences Miss Am.(top) and Fl. Grad (bottom) corrupted by Gaussian noise with zero mean and standard deviation = 40 denoised by RNLM,RNLM(No BMA),VBM3D,BM3D, and SNLM

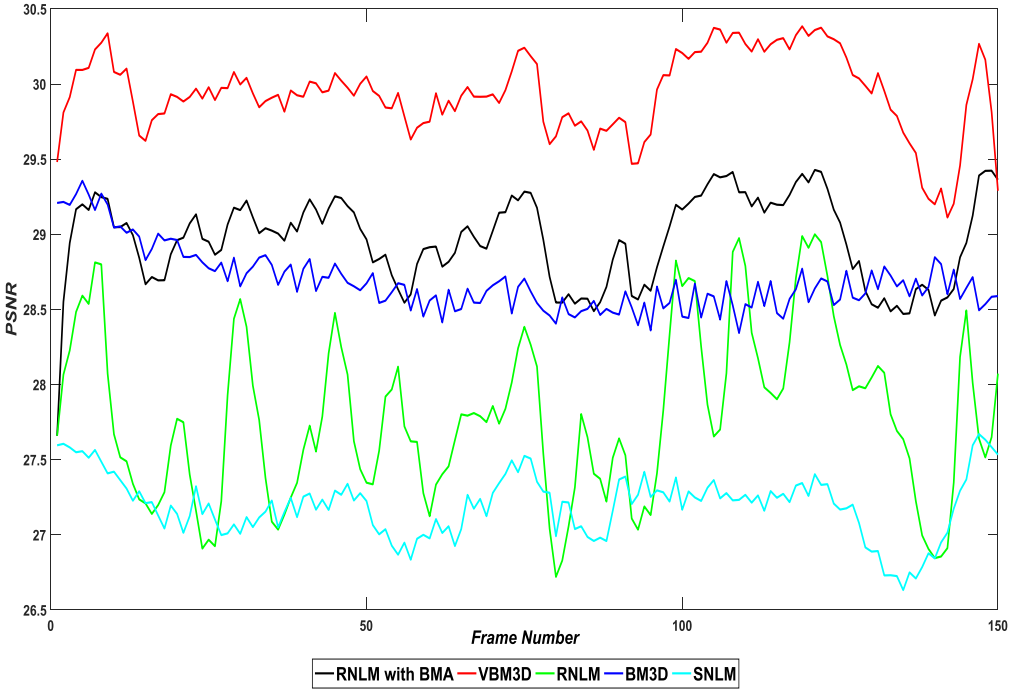


Figure 4.3: PSNR comparison for Foreman sequence corrupted with a noise level of $\sigma = 40$.

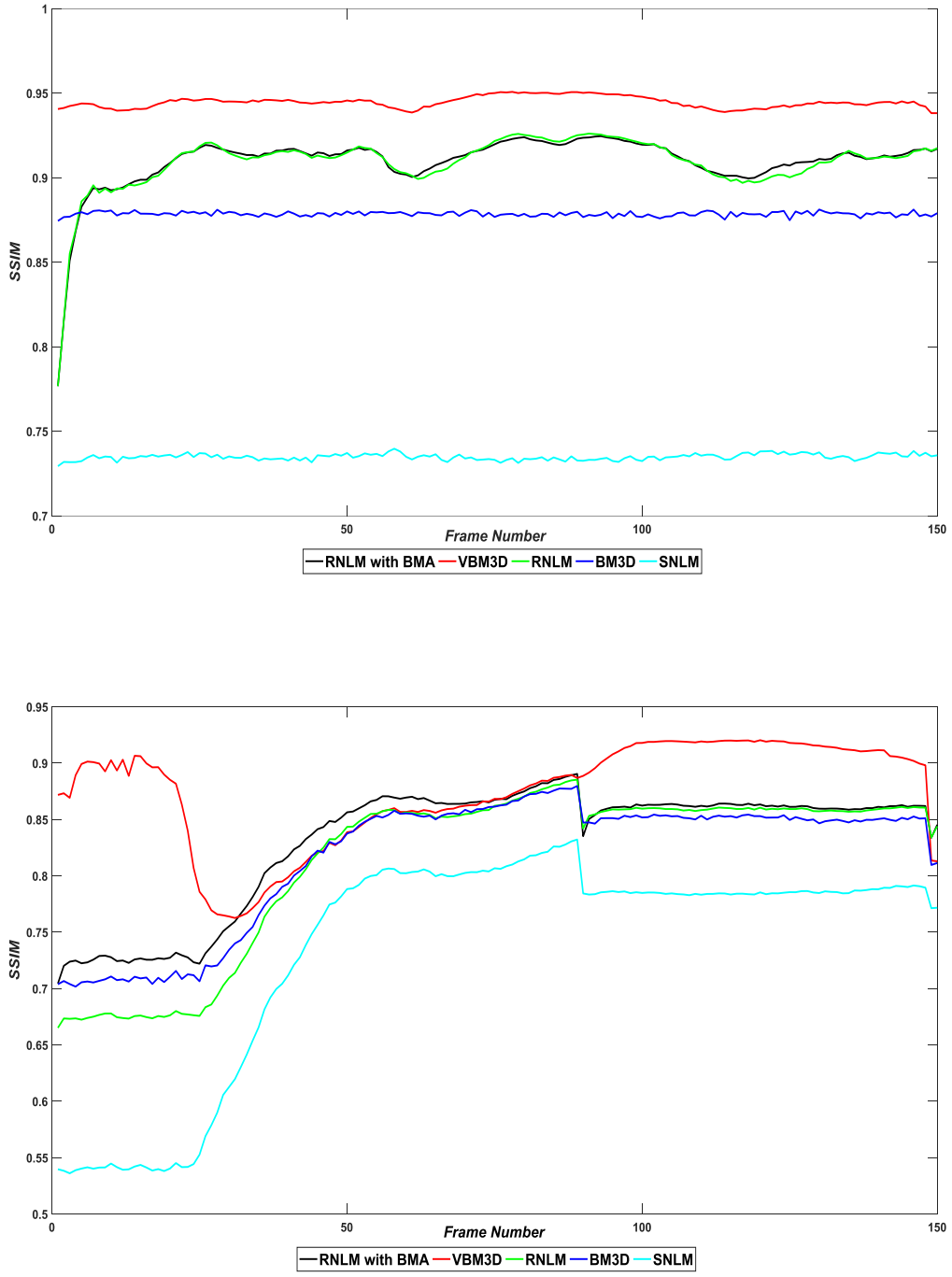
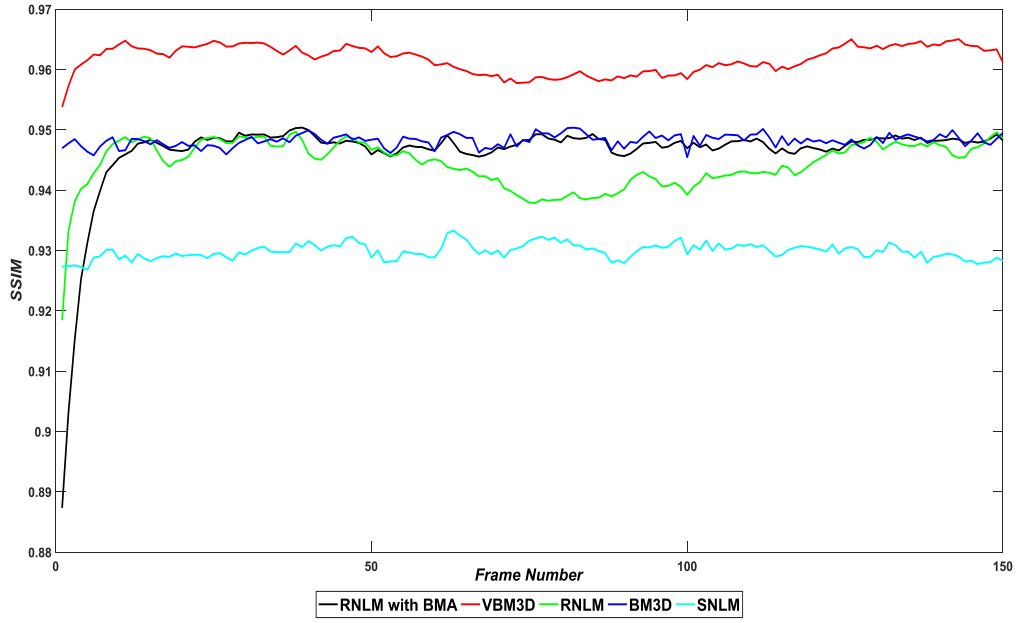


Figure 4.4: Shows the Comparison of SSIM output for two video sequences Salesman(top) and Tennis (bottom) corrupted by Gaussian noise with zero mean and standard deviation = 40 denoised by RNLM,RNLM(No BMA),VBM3D,BM3D, and SNLM



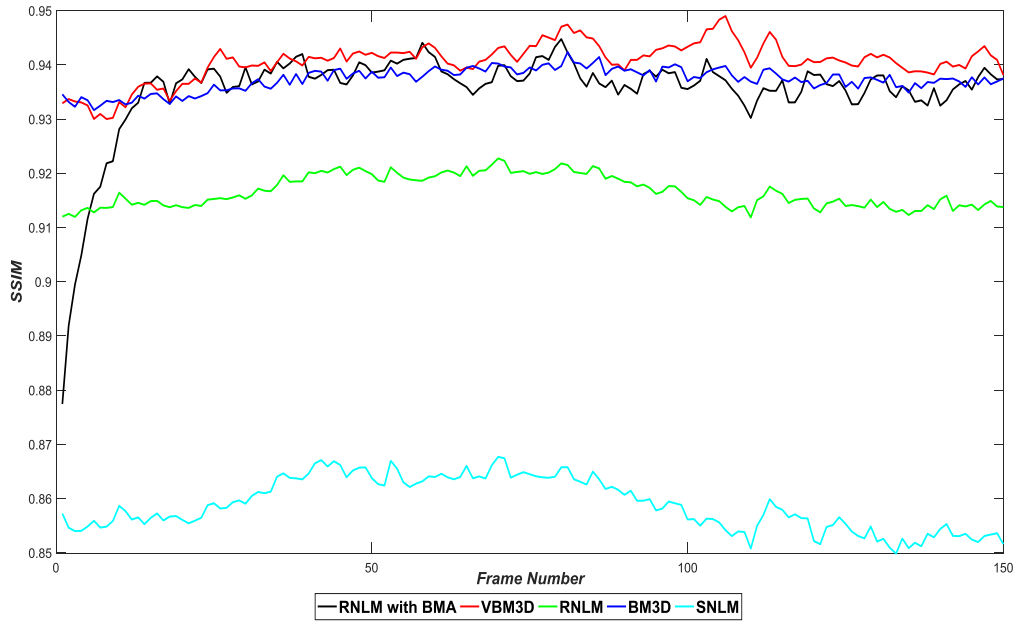


Figure 4.5: Shows the Comparison of SSIM output for two video sequences Miss Am.(top) and Fl. Grad (bottom) corrupted by Gaussian noise with zero mean and standard deviation = 40 denoised by RNLM,RNLM(No BMA),VBM3D,BM3D, and SNLM

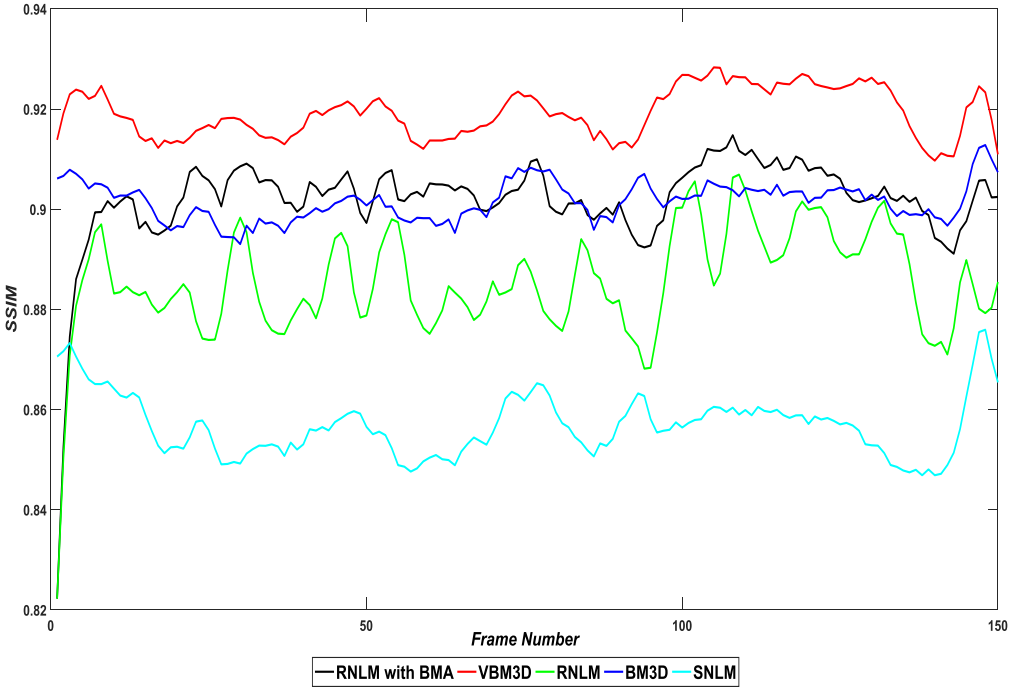


Figure 4.6: SSIM comparison for Foreman sequence corrupted with a noise level of $\sigma = 40$.

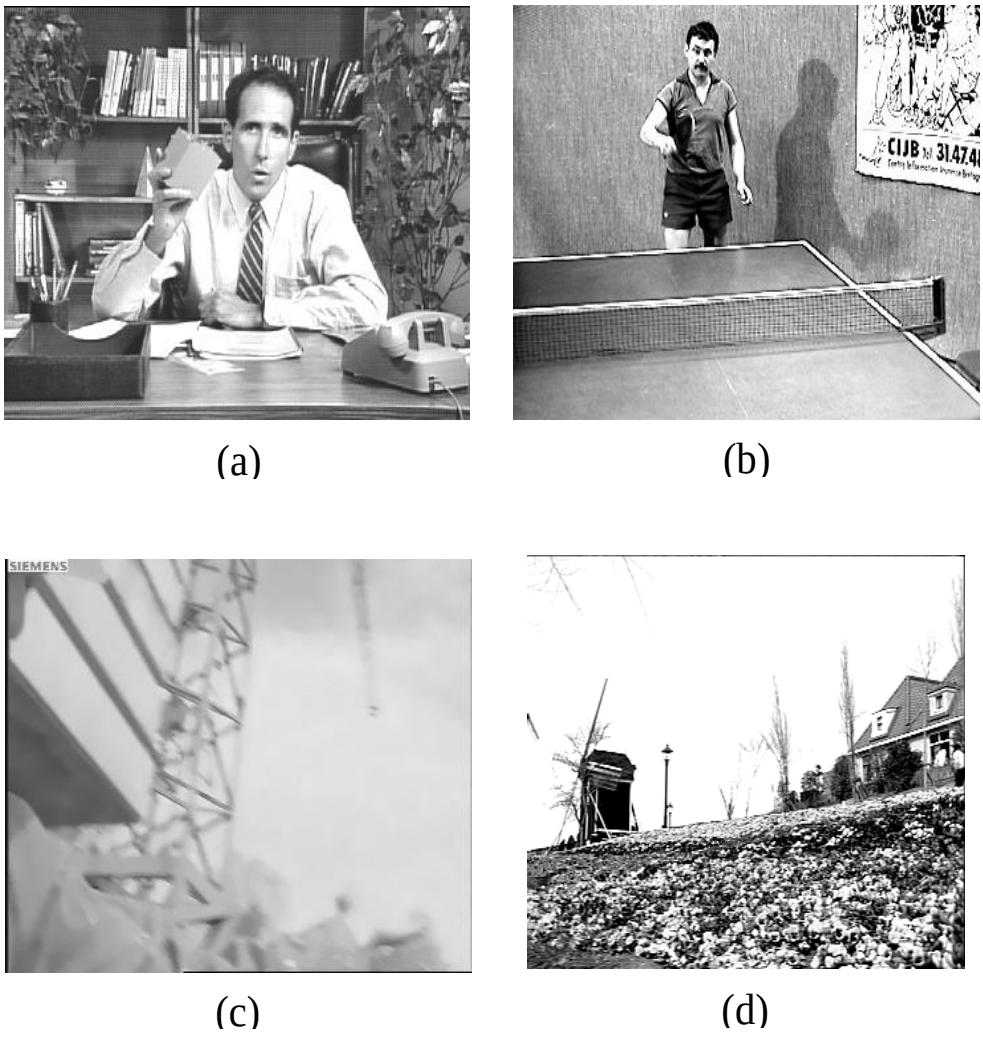


Figure 4.7: All truth sequence frame used in experimental results. (a) Salesman (288×352); (b) Tennis (288×352); (c) Miss Am.(288×352); (d) Fl. Gard (288×352); and (e) Foreman (288×352).

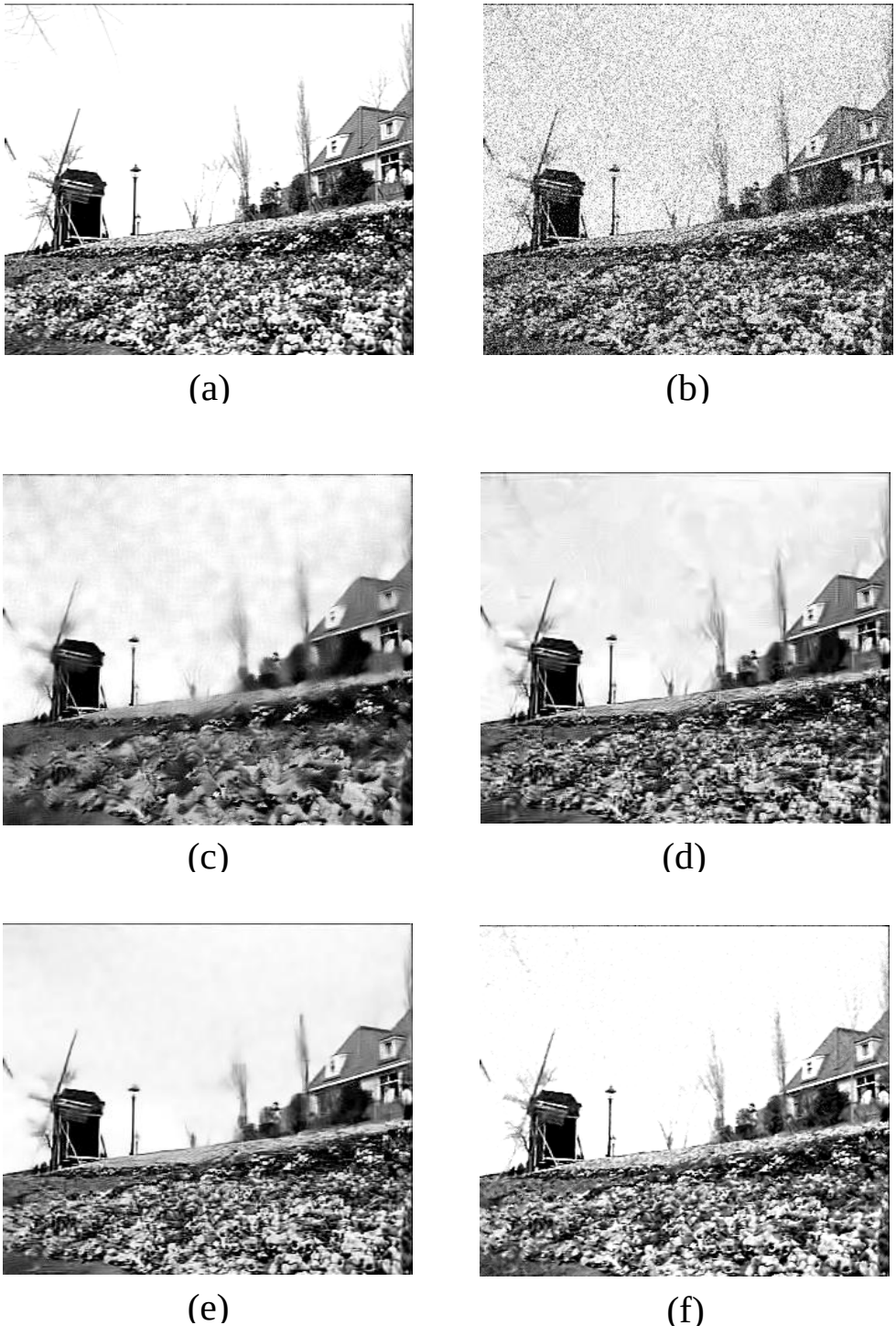


Figure 4.8: Comparison of denoised Frame 93 from the Flower Garden sequence with $\sigma_n = 30$. (a) Original frame, (b) corrupted frame, (c) SNLM, (d) BM3D, (e) V-BM3D, (f) RNLM (BMA).

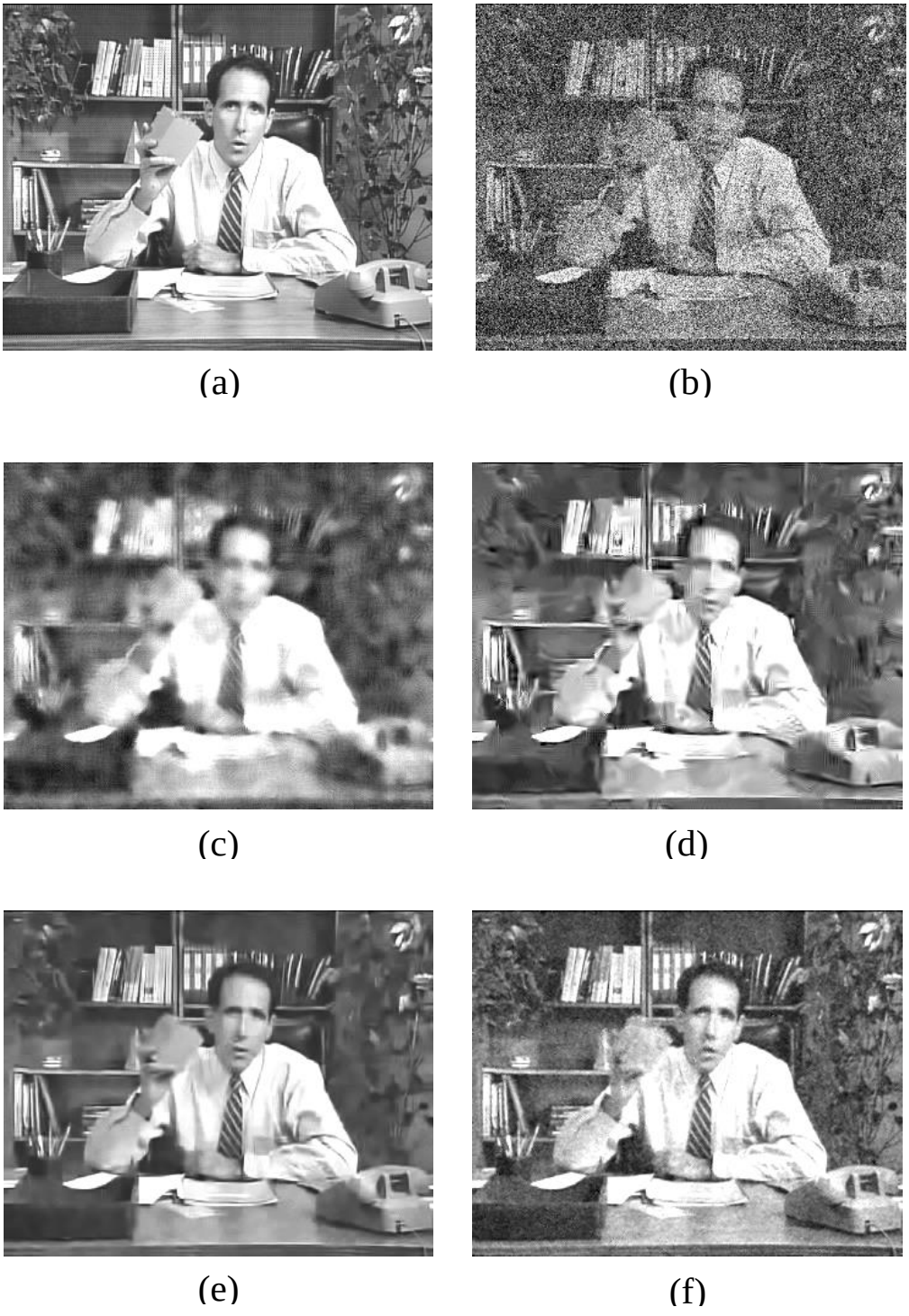


Figure 4.9: Comparison of denoised Frame 93 from the Salesman sequence with $\sigma_n = 40$. (a) Original frame, (b) corrupted frame, (c) SNLM, (d) BM3D, (e) V-BM3D, (f) RNLM (BMA).

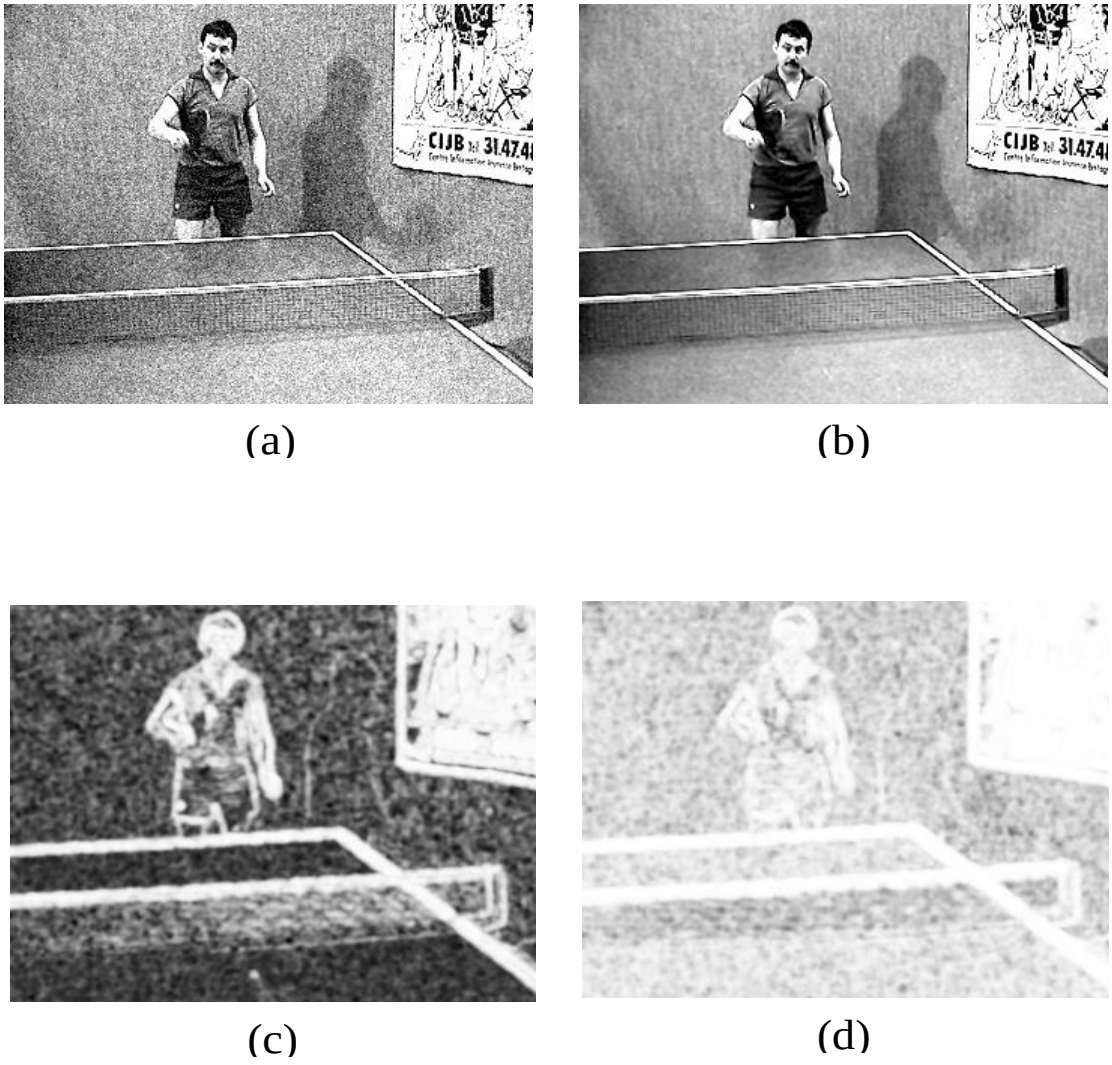


Figure 4.10: Denoising results of frame 114 in Tennis sequence with $\sigma_n = 20$. (a) Original frame, (b) corrupted frame, (c) SSIM quality map for corrupted frame (d) SSIM quality map for RNLM (BMA) frame

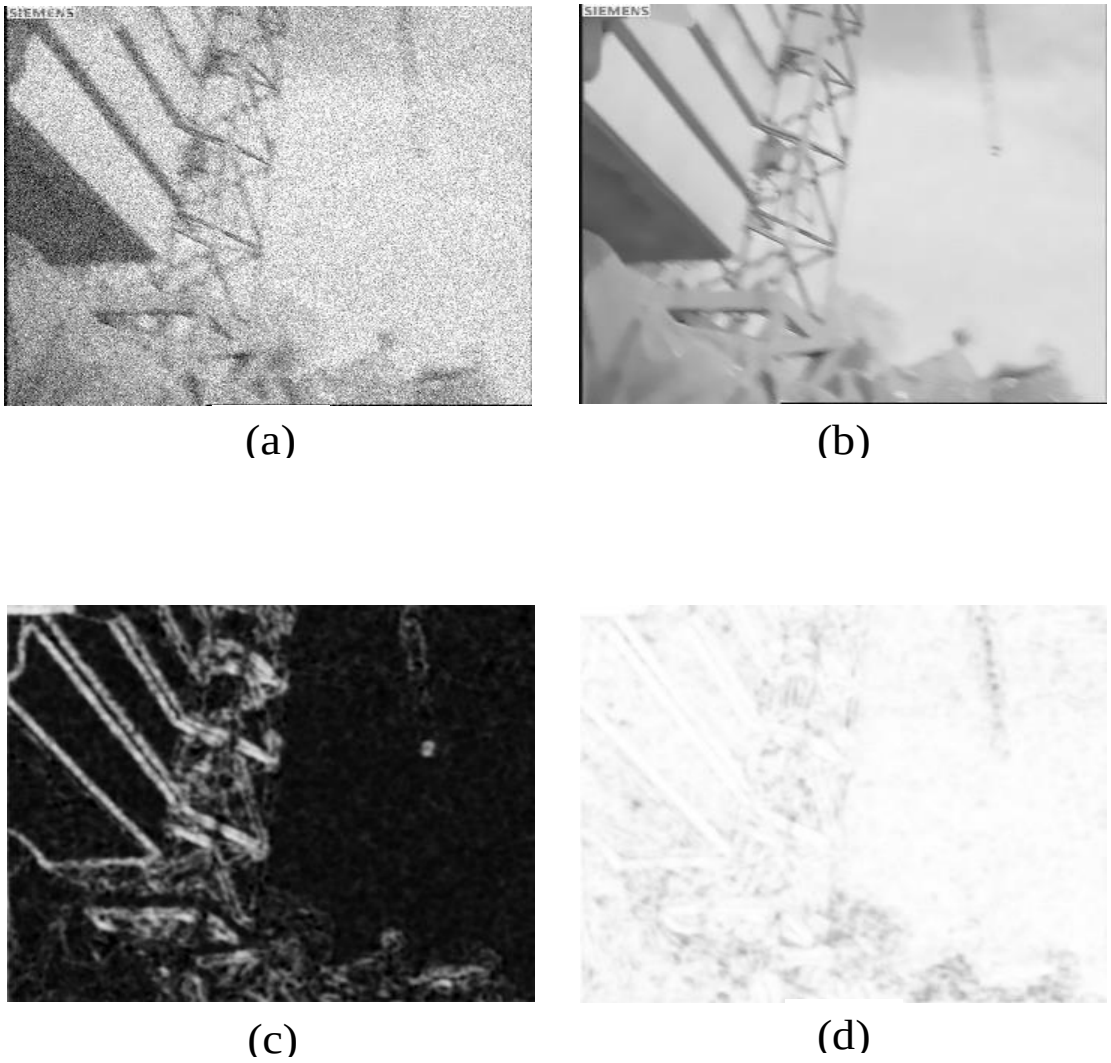


Figure 4.11: Denoising results of frame 573 in Foreman sequence with $\sigma_n = 10$. (a) Original frame, (b) corrupted frame, (c) SSIM quality map for corrupted frame (d) SSIM quality map for RNLM (BMA) frame

CHAPTER V

CONCLUSION

In this thesis, we have presented a new temporally-recursive NLM algorithm for video denoising. The output of the new RNLM method is a weighted sum of pixels from the current noisy frame, and of a selected pixel from the prior processed frame. This type of recursive processing allows us to exploit temporal correlation with little additional computational cost, compared with a SNLM. We have explored two versions of the RNLM method here, RNLM (BMA) that uses BMA for motion compensation, and RNLM (No BMA) that simply uses the previous pixel at the same location to be included in the weighted sum. The results also show that RNLM (BMA) consistently outperformed the RNLM (No BMA). The computational complexity of the RNLM (BMA) method is approximately the same as 3D-NLM with just two frames. Both versions of RNLM method consistently outperform SNLM. Furthermore, our results show that RNLM (BMA) is competitive with much more computationally complex algorithms, such as BM3D and V-BM3D. We believe the proposed method offers a simple and practical video denoising solution, capable of balancing noise reduction and detail preservation. Because of its low computational cost, we believe it is well suited for real-time implementation.

BIBLIOGRAPHY

[1] V. Argyriou, J. M. Del Rincon, B. Villarini, and A. Roche, *Image, video and 3D data registration: medical, satellite and video processing applications with quality metrics*. John Wiley & Sons, 2015.

[2] V. Madisetti, *Video, Speech, and Audio Signal Processing and Associated Standards*. CRC Press, 2009.

[3] F. Luisier, T. Blu, and M. Unser, "Image denoising in mixed poisson–gaussian noise," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 696–708, 2011.

[4] Z. Lin, X. Li, and Z. Sun, "The summary of video denoising method," 2015.

[5] G. Chen, J. Zhang, D. Li, and H. Chen, "Robust kronecker product video denoising based on fractional-order total variation model," *Signal Processing*, vol. 119, pp. 1–20, 2016.

[6] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[7] M. Mahmoudi and G. Sapiro, "Fast image and video denoising via nonlocal means of similar neighborhoods," *Signal Processing Letters, IEEE*, vol. 12, no. 12, pp. 839–842, 2005.

[8] J. Wang, Y. Guo, Y. Ying, Y. Liu, and Q. Peng, "Fast non-local algorithm for image denoising," in *Image Processing, 2006 IEEE International Conference on*. IEEE, 2006, pp. 1429–1432.

[9] T. Brox, O. Kleinschmidt, and D. Cremers, "Efficient nonlocal means for denoising of textural patterns," *Image Processing, IEEE Transactions on*, vol. 17, no. 7, pp. 1083–1092, 2008.

[10] P. Coupé, P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot, "An optimized blockwise nonlocal means denoising filter for 3-d magnetic resonance images," *Medical Imaging, IEEE Transactions on*, vol. 27, no. 4, pp. 425–441, 2008.

[11] C. Kervrann and J. Boulanger, "Optimal spatial adaptation for patch-based image denoising," *Image Processing, IEEE Transactions on*, vol. 15, no. 10, pp. 2866–2878, 2006.

[12] P. Chatterjee and P. Milanfar, "A generalization of non-local means via kernel regression," in *Electronic Imaging 2008*. International Society for Optics and Photonics, 2008, pp. 68 140P–68 140P.

[13] B. Goossens, Q. Luong, A. Pizurica, and W. Philips, "An improved non-local denoising algorithm," in *2008 International Workshop on Local and Non-Local Approximation in Image Processing (LNLA 2008)*, 2008, pp. 143–156.

[14] D. Van De Ville and M. Kocher, "Sure-based non-local means," *Signal Processing Letters, IEEE*, vol. 16, no. 11, pp. 973–976, 2009.

[15] R. Vignesh, B. T. Oh, and C.-C. J. Kuo, "Fast non-local means (nlm) computation with probabilistic early termination," *Signal Processing Letters, IEEE*, vol. 17, no. 3, pp. 277–280, 2010.

[16] V. Karnati, M. Uliyar, and S. Dey, "Fast non-local algorithm for image denoising," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 3873–3876.

[17] A. Buades, B. Coll, and J.-M. Morel, "Denoising image sequences does not require motion estimation," in *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*. IEEE, 2005, pp. 70–74.

[18] ——, "Nonlocal image and movie denoising," *International journal of computer vision*, vol. 76, no. 2, pp. 123–139, 2008.

[19] Y. Han and R. Chen, "Efficient video denoising based on dynamic nonlocal means," *Image and Vision Computing*, vol. 30, no. 2, pp. 78–85, 2012.

[20] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3d filtering," in *Electronic Imaging 2006*. International Society for Optics and Photonics, 2006, pp. 606 414–606 414.

[21] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3d transform-domain collaborative filtering," in *Signal Processing Conference, 2007 15th European*. IEEE, 2007, pp. 145–149.

[22] Z. Wang and L. Zhang, "An adaptive fast non-local image denoising algorithm," *Journal of Image and Graphics*, vol. 14, no. 4, pp. 669–675, 2009.

[23] J. V. Manjón, J. Carbonell-Caballero, J. J. Lull, G. García-Martí, L. Martí-Bonmatí, and M. Robles, "Mri denoising using non-local means," *Medical image analysis*, vol. 12, no. 4, pp. 514–523, 2008.

[24] H. Li and C. Y. Suen, "A novel non-local means image denoising method based on grey theory," *Pattern Recognition*, vol. 49, pp. 237–248, 2016.

[25] X. Li and Y. Zheng, "Patch-based video processing: a variational bayesian approach," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, no. 1, pp. 27–40, 2009.

[26] J. Boulanger, C. Kervrann, and P. Bouthemy, "Space-time adaptation for patch-based image sequence restoration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 1096–1102, 2007.

[27] "Video trace library." [Online]. Available: http://trace.eas.asu.edu/index.html

[28] Z. Wang and A. C. Bovik, "A universal image quality index," *Signal Processing Letters, IEEE*, vol. 9, no. 3, pp. 81–84, 2002.

[29] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.

[30] K. M. Mohamed and R. C. Hardie, "A collaborative adaptive wiener filter for image restoration using a spatial-domain multi-patch correlation model," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 1, pp. 1–23, 2015.

[31] V. Zlokolica, A. Pižurica, and W. Philips, "Wavelet-domain video denoising based on reliability measures," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 16, no. 8, pp. 993–1007, 2006.

[32] A. Pizurica, V. Zlokolica, and W. Philips, "Combined wavelet domain and temporal video denoising," in *Advanced Video and Signal Based Surveillance, 2003. Proceedings. IEEE Conference on*. IEEE, 2003, pp. 334–341.

[33] I. W. Selesnick and K. Y. Li, "Video denoising using 2d and 3d dual-tree complex wavelet transforms," in *Optical Science and Technology, SPIE's 48th Annual Meeting*. International Society for Optics and Photonics, 2003, pp. 607–618.

[34] S. M. Rahman, M. O. Ahmad, and M. Swamy, "Video denoising based on inter-frame statistical modeling of wavelet coefficients," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 2, pp. 187–198, 2007.

[35] G. Varghese and Z. Wang, "Video denoising based on a spatiotemporal gaussian scale mixture model," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 7, pp. 1032–1040, 2010.

[36] M. Protter and M. Elad, "Image sequence denoising via sparse and redundant representations," *Image Processing, IEEE Transactions on*, vol. 18, no. 1, pp. 27–35, 2009.