Structure-Oriented Directional Approaches to Video Noise Estimation and Reduction

Mohammed Asaad Ghazal

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

May, 2006

Your file Votre référence
ISBN: 978-0-494-20746-8
Our file Notre référence
ISBN: 978-0-494-20746-8

ABSTRACT

Structure-Oriented Directional Approaches to Video Noise Estimation and Reduction

Mohammed Asaad Ghazal

Video has become increasingly used in television broadcast, Internet, and surveillance applications. The presence of noise in video signals is not only visually unacceptable, but also hinders the performance of video processing applications. Thus, the interest in researching methods for fast, automated, and robust techniques to estimate and reduce image and video noise has grown over the years.

This thesis proposes approaches to estimate and reduce additive white Gaussian noise (AWGN) in video signals that are adaptive to frame structure and noise level. First, a spatio-temporal method for estimating the variance of AWGN is proposed. The method divides the video signal into cubes. Cube homogeneity is measured using Laplacian of Gaussian operators. The variances of homogeneous cubes calculated along homogenuous plains are used to estimate the noise variance. The Least Median of Squares (LMS) robust estimator is utilized to reject outliers and produce the domain-wise noise variance estimate. The domain-wise estimates are averaged to obtain the frame-wise estimate. The proposed algorithm works well for video sequences with high structure and motion activity with a maximum estimation error of 1.7 dB.

The thesis then proposes a framework for spatial adaptive multi-directional filtering of AWGN in video frames and adaptive multi-directional Sigma and Wiener filters. The proposed multi-directional Sigma filter achieves gains in the Peak Signal to Noise Ratio (PSNR) of up to 4.8 dB in real-time. The proposed multi-directional Wiener filter achieves gains in PSNR of up to 5.6 dB and is well suited for offline applications. The structure preservation capabilities of the proposed filters are studied using the Modulation Transfer Function.

Acknowledgments

بسم الله الرحمن الرحيم

«وَمَا تَوفِيقِي إِلاَّ بِاللهِ عَلَيهِ تَوَكَّلتُ وَإِلَيهِ أُنِيبُ»

صدق الله العظيم

I would like to begin by expressing my deepest appreciations and regards to my supervisors Dr. Aishy Amer and Dr. Ali Ghrayeb for their tremendous support, patience and confidence. Thanks to them, my interest in Video Processing was sparked and will continue to grow for years to come. I am the researcher and the Engineer that I am because of what they taught me. My gratitude goes as well to all my professors at *Concordia University* and Dr. Mohammed Al Rousan.

I would like to dedicate the success of this modest effort to my dear parents Asaad and Amal Ghazal. Where would I be without your love and support?! It is in your eyes that I saw the motivation to work through the day and study through the night. Making you proud has always been my objective all along. I pray and work hard in the hope that one day I will able to return some of the kindness and warmth that you overwhelmed me with. My gratitude goes to my wonderful family, my brother Mahmoud, my sisters Linda, Lamia and Lubna, my nieces Laila, Hala, Sama and Amal. You are the world to me. Last but not least, a big thank you to my beloved wife Sara. You are the ornament of my life. My journey with you has been wondeful.

Thanks to all my colleagues at VidPro; El Helali, Firas, Francois, Hanif, Chang Su, Kumara, Julius, Bin, Ken and Dr. Carlos Vazquez for their support. Thanks to my friends Abdullah El-Shazly and Mohammed Abu Khousa for being there for me.

Contents

List of Figures

List of Tables

List of Notations

General Notations

AWGN	Additive White Gaussian Noise
DVD	Digital Versatile Disc
PAL	Phase Alternate Line
NTSC	National Television Standards Committee
CIF	Common Intermediate Format
2D	Two Dimensional
3D	Three Dimensional
σ_η^2	Variance of AWGN noise process
(i, j, n)	Spatio-temporal pixel coordinates
V	Noise-free digital video signal
η	Noise component added to V
V_η	Noisy digital video signal
PSNR_η	Peak Signal to Noise Ratio of V_η
SNR	Signal to Noise Ratio
HVS	Human Visual System

Spatio-Temporal Video Noise Estimation

C_{klm}	3D portion of the video signal (or cube) with coordinates (k,l,m)
W	Window Size
Ψ_{klm}	Cubic window with spatial index k,l and m
D	A variable to represent a particular domain (S, T, ST, HT or VT)
ζ_D	Homogeneity (intensity invariance) in the D domain
LoG	Laplacian of Gaussian
$h(x,y)$	2D impulse response
$X(0)$	Random variable corresponding to noise-free center pixel i
$X(i)$	Random variable corresponding to noise-free neighboring pixel i
$N(0)$	Random variable corresponding to noise added to center pixel i
$N(i)$	Random variable corresponding to noise added to neighboring pixel i
$G(0)$	Random variable corresponding to noisy center pixel i
$G(i)$	Random variable corresponding to noisy neighboring pixel i
Y	Random variable corresponding to the response of the Laplacian operator
U_C	Set of all selected homogeneous cubes
U_D	Set of selected homogeneous cubes in domain D
$U_D^{\sigma^2}$	Set of variances of selected homogeneous cubes in domain D
L	Number of homogeneous cubes considered
L_{max}	Maximum possible number of selected homogeneous cubes
σ^2_{init}	Initial estimate of noise variance
PSNR$_{init}$	Initial estimate of noise PSNR
β	Scaling factor
μ_D	Signal mean calculated in domain D
σ^2_D	Signal variance calculated in domain D
μ_{D_ρ}	Signal mean calculated in domain D along plain ρ
$\sigma^2_{D_\rho}$	Signal variance calculated in domain D along plain ρ
α	Variance index
$\widehat{\sigma}^2_D$	Domain-wise noise variance estimate
R	Search range for variances used by the Least Median of Squares estimator
σ^2_p	A variance in R
Q	Search step size
$\widehat{\sigma}^2_\eta$	Frame-wise noise estimate
μ_E	Mean of estimation error
σ^2_E	Variance of estimation error

Spatial Adaptive Multi-Directional Noise Reduction

F	Noise-free video frame or image
η	Noise frame added to video frame or image
F_η	Noisy video frame or image
$\hat{F}$	Noise reduced frame or image
(i, j)	Spatial coordinate (i,j)
W	Window size
z	Homogeneity analyzer mask index also a direction index
ζ_z	Homogeneity along direction z or response of applying mask z to F
S	Set of mask responses $\{\zeta_z\}$
z_{min_1}	Index of most homogeneous direction
z_{min_n}	Index of n^{th} most homogeneous direction
D	Ordered sequence of mask indexes based on ζ_z
D_n	Number of directions used
$G_W(z)$	Spatial mask of size W used to isolate non-center pixels in direction z
$\omega(a, b)$	Weighting for pixels at (a,b)
C_η	Center pixel weight
C'_η	Non-center pixel weight
R	Noise reduction gain
$\sigma^2_{\eta_o}$	Noise variance after noise reduction (noise variance for $\hat{F}$)
B_l	The range of PSNR$_\eta$ corresponding to level l (window size or number of direction)
b_l, b_{l-1}	Boundary values for B_l
g_l	Window size level or number of directions level
$[A]$	Autocorrelation matrix of the set of non-center pixels
r	Correlation vector between the center pixel and all non-center pixels
t_η	Threshold in PSNR$_\eta$
κ	Number of directions in the directional Sigma filter
$F_\eta(k, l)$	Neighboring pixel in a selected direction z
K_x	Number of local pixels with gray level value z
K	Number of local pixels
$H(x)$	Normalized local histogram
$P(x)$	Estimation of the local cumulative distribution function of the image or frame
CDF	Cumulative Distribution Function
MTF	Modulation Transfer Function
ESF	Edge-Spread Function
σ^2_F	Variance of noise-free signal
μ_F	Mean of noise-free signal
$\sigma^2_{F_\eta}$	Variance noisy signal
$\hat{\sigma^2_F}$	Estimated variance of noise-free signal
μ_z	Mean along direction z
σ^2_z	Variance along direction z

Chapter 1

Introduction

1.1 Motivation

The demand for visual information is increasing on daily basis. Statistics Canada revealed that Canadians spent 21.6 hours per week watching television in 2002 [6]. Watching television at home is not the only form of daily interaction we have with video. For example, we record and watch video on the move with camcorders, mobile phones, portable televisions, iPods, and portable DVD players.

The rapidly increasing interest in visual information (or more specifically video) triggered a move from the analog to the digital domain spawning in the way many new digital video processing opportunities. Academia and the industry are able to realize many of these opportunities in the form of a wide range of video-related services and technologies. For example, consumers expect to watch high-quality television and record their favorite shows on the smallest amounts of storage or they want to be able to watch PAL DVDs on NTSC television sets and vice versa. Internet users want to search databases of entertainment videos for specific shows or scenes of interest. Security and law enforcement agencies require extracting significant portions of surveillance videos and querying them by events. Film makers need to restore and preserve motion pictures and video tapes recorded over the last century for their historic, cultural and artistic values.

In all mentioned video processing applications, noise is considered to be an undesirable phenomena. For example, video display devices with visible noise are unacceptable. Moreover, uncorrelated noise consumes unnecessary bandwidth in video compression. Video motion estimation algorithms suffer from noise interfering with the estimation process. Similarly, Video object segmentation algorithms may misclassify backgrounds and objects due to noise. Since noise is inevitable, the presence of noise must be accounted for through adaptation, preprocessing or post-processing.

Video noise estimation and reduction are used countermeasures against noise. They stem from a branch of video processing referred to as *Video Quality Enhancement*. As can be seen from Fig. 1.1, video noise estimation and reduction fit into the preprocessing and/or the post-processing stages of typical video processing systems. Information about the noise process (e.g., noise variance) is estimated in the video noise estimation stage and relayed to the video noise reduction stage which may utilize this information to tune the noise reduction process for better performance.

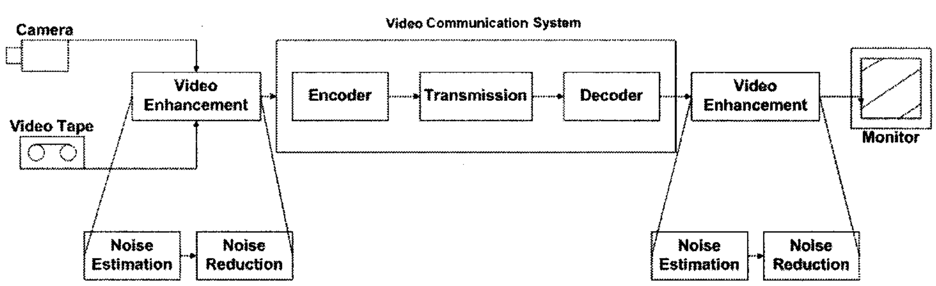


Figure 1.1: Video noise estimation and reduction in a typical video communication system.

The need for fast, accurate and robust video noise estimation algorithms rises from the fact that many video processing algorithms such as compression, deinterlacing, motion estimation and file format conversion require a priori knowledge of the noise present in the signal in order to adapt their parameters and improve performance. For example, noise reduction filters can be tuned to perform better with a priori knowledge of the noise. Noise reduction filters themselves have become integral components added to video applications to improve the overall performance. Different applications bring forth different requirements for noise reduction filters. For example, with the emergence of new image and video products and services, such as in mobile devices, digital cameras and iPods, rises the need for real-time noise reduction filters with high algorithmic speed, low memory consumption, and ability to handle variable noise levels. Offline applications are less interested in algorithmic speed or memory consumption and more interested in producing high gain in quality (e.g., PSNR) while still preserving high frequency image or video content such as fine details and structures.

This thesis proposes noise estimation and reduction algorithms tailored for the requirement and characteristics of digital image and video signals. The noise estimation algorithm utilizes spatio-temporal information present in video signals to estimate the noise level. The noise reduction algorithm utilizes information about the noise level to adapt the filtering process and achieve high gain in video quality.

1.2 Background

This Section first introduces the common types of noise to affect video signals. It then moves to describe the adopted video noise model and quality measures. Finally, it gives an overview of the basic methods for video noise reduction.

1.2.1 Types of Noise

Noise in this thesis refers to unwanted stochastic variations as opposed to deterministic distortions such as shading or lack of focus. Image and video signals are affected by noise regardless of the precision of the recording equipment. They may be corrupted with noise from various sources such as camera noise, shot noise from electronic hardware and the storage on magnetic tape, thermal noise and granular noise on film [7]. Impulse noise is also added through bit errors during signal transmission. When the signal is transmitted through analog channels, Gaussian noise is another form of noise to corrupt the signal. Image and video compression standards operate on the block-level and introduce signal or motion discontinuity along block boundaries or blocking artifacts [8].

1.2.2 Video Noise Model

The noise signal can be added to the video signal (i.e., additive noise) or multiplied with the video signal (i.e., multiplicative noise). Noise can also be signal dependent or signal independent. Moreover, The noise signal is classified as white or color noise based on its spectral properties. Usually, the noise component results from a mixture of contributions from various noise sources. In practice, the aggregate effect of noise is modeled as an additive white Gaussian noise (AWGN) process [7,9,10] with zero mean and variance σ_η^2 that is independent from the ideal uncorrelated video V. Accordingly, the corrupted digital noisy video signal is given by

$$V_\eta = V + \eta, \tag{1.1}$$

where η is the added noise component. The noise estimation problem now reduces to estimating the variance of the noise, σ_η^2, which is sufficient to characterize the noise process. The noise reduction problem reduces to obtaining the best possible estimate $\hat{V}$ of the original signal V given only V_η.

1.2.3 Video Quality Measurement

To assess the performance or a noise estimation algorithm or measure the gain by a noise reduction algorithm, a measure of the quality of a video signal must be defined first. Image and video quality assessment is not a trivial task and is still an ongoing research process. The most commonly used measure in video processing literature is the Peak Signal to Noise Ratio (PSNR) [11] given by

$$\mathrm{PSNR} = 10 \cdot \log \frac{(255)^2}{\sum\limits_{i,j,n} \left(V_\eta(i,j,n) - V(i,j,n)\right)^2}, \tag{1.2}$$

where (i,j,n) is the spatio-temporal coordinate of the signal element or pixel. The PSNR gives a measure of the improvement in Signal-to-Noise (dB). Contrary to the Signal-to-Noise Ratio (SNR), it is independent of the signal. Despite this, the PSNR is still unweighted with respect to visual perception. In other words, while the PSNR reflects the improvement in the quality of a video signal it does not necessarily reflect human subjective perception.

We can obtain the noise PSNR (i.e., PSNR_η) from the noise variance σ_η^2 using

$$\mathrm{PSNR}_\eta = 10 \cdot \log \frac{(255)^2}{\sigma_\eta^2} \tag{1.3}$$

and the noise variance σ_η^2 from the PSNR_η using

$$\sigma_\eta^2 = \frac{(255)^2}{10^{\frac{\mathrm{PSNR}_\eta}{10}}} \tag{1.4}$$

The Modulation Transfer Function (MTF) measures the degradation of object contrast due to blurring as a function of the spatial frequency. It will be used in this thesis to assess the structure preservation capabilities of noise reduction filters. The MTF can be calculated as the Fourier transform of the first derivative of the Edge

Spread Function (ESF) which is defined as the response of the system to an ideal edge. The ESF can be approximated from the Cumulative Distribution Function (CDF) of the signal and the fact that the locality from which the CDF was estimated represents a hard edge.

1.3 Summary of Proposed Methods

This thesis proposes approaches for estimating and reducing AWGN in video signals. First, a method for AWGN variance estimation is proposed which utilizes spatial, temporal and spatio-temporal information independently. The algorithm operates on units of 3D portions of the signal or cubes. The video is divided into cubes and the spatial, temporal and spatio-temporal homogeneity of the cubes are measured using 3D Laplacian of Gaussian operators. The variance of a selected number of homogeneous cubes calculated along homogeneous plains is recorded. A Least Median of Squares (LMS) robust estimator is applied to select the domain-wise (spatial, temporal and spatio-temporal domains) estimates of the noise variance. The domain-wise noise variance estimates are averaged to produce the frame-wise final noise variance estimate.

The thesis then proceeds to propose a framework for adaptive multi-directional filtering and multi-directional Sigma and Wiener filters. In the proposed framework, filtering is performed along homogeneous directions and not across edges to combat blurring. Window size, shape and pixel weighting are adapted to the image content and the noise level to optimize the filtering process. First, an adaptive multi-directional Sigma filter suitable for real-time image or video noise reduction is proposed. It achieves gains of up to 4.8 dB PSNR and is capable of preserving image content. Second, an adaptive multi-directional Wiener filter that achieves gains of up to 5.6 dB PSNR. The proposed directional Wiener filter is well suited for offline video noise reduction. To show the efficacy of the proposed framework, we use, besides the PSNR, the Modulation Transfer Function to measure the degradation in object contrast due to blurring as a function of the spatial frequency.

1.4 Overview of Contributions

The following list states which parts of this thesis are original to the knowledge of the author at the time the proposed methods of this thesis were developed:

- Spatio-temporal Video Noise Estimation
  - Division and treatment of the video signal as a set of 3D cubes
  - Development of 3D Laplacian of Guassian operators to measure spatial, temporal, and spatio-temporal homogeneity of video cubes.
  - Utilization of the Least Median of Squares (LMS) robust estimator to calculate the domain-wise (spatial, temporal and spatio-temporal) noise variance estimate.
- Spatial Adaptive Directional Noise Reduction
  - Generalization of the directional filtering of [3].
  - Introduction of noise level adaptation in directional filtering toward development of adaptive directional filtering.
  - Theoretical study of the benefits of adaptive directional filtering.
  - Development of a procedure to find the optimal window size, number of directions and directional filter coefficients.
  - Development of structure-oriented multi-directional Sigma and Wiener filters.

- Utilization of the Modulation Transfer Function to measure the structure preservation capabilities of the proposed spatial filters.

In addition, various referenced methods for real-time video noise estimation and reduction were studied, implemented, and their performance analyzed and compared with the proposed methods.

1.5 Thesis Outline

The remainder of the thesis is organized as follows. Chapter 2 presents the proposed spatio-temporal video noise estimation method. A review of related work is given in Section 2.2 and is followed by a description of the proposed approach in Section 2.3. Simulation results are provided in Section 2.4 for a representative set of video sequences with different levels of texture and types of motion activity. Chapter 3 follows and presents the proposed framework for spatial adaptive multi-directional filtering of noise in video frames. It starts with an overview of the related work in Section 3.2 followed by analysis of generalized multi-directional filtering and a study of the benefits of adaptive multi-directional filtering in Section 3.3. Section 3.3.3 presents the proposed multi-directional Sigma and Wiener filters. Subjective and Objective simulation results are provided for a representative set of images and videos corrupted by typical levels of noise in Section 3.6. Chapter 4 concludes the thesis and suggests future work.

Chapter 2

Spatio-Temporal Video Noise Estimation

2.1 Introduction

This chapter proposes a low-complexity algorithm that uses both spatial (intra-frame) and temporal (inter-frame) information to yield a stable and robust estimate of the noise variance. The proposed method divides the video signal into cubes and measures their homogeneity. The noise variance is then estimated from a set of selected cubes along the homogeneous plains only. The Least Median of Squares robust estimator is used to estimate the dimension-wise (spatial, temporal and spatio-temporal) noise variance. The dimension-wise estimates are averaged to produce the frame-wise noise variance.

The remainder of the chapter is as follows. Section 2.3 presents the proposed approach theoretically and gives an interpretation of its good performance. Objective simulation results are presented and discussed in Section 2.4. Finally, Section 2.5 summarizes the chapter.

2.2 Related Work

Proposed algorithms for estimating the variance σ_η^2 of the AWGN are either inter-frame or intra-frame based. There exist few methods for inter-frame noise estimation [2, 12]. These methods are challenged by the presence of object or global motion. Motion detection or motion compensation are commonly used countermeasures. Hence, methods in this area such as the one in [12] tend to be computationally expensive. The method in [2] attempts to utilize temporal adaptation to stabilize the spatially estimated noise variance.

Many methods for intra-frame noise estimation have been presented. Difficulties with these methods rise from frames with very high or very low noise levels as well as highly structured frames. The problem lies in determining whether intensity variations are due to noise or frame details. Intra-frame methods are categorized into smoothing-based, wavelet-based and block-based methods. Smoothing based algorithms such as the one in [13] estimate noise from the difference of the noisy frame and its smoothed version. The assumption is that the difference frame represents an approximation of the noise signal. These approaches are computationally expensive and tend to overestimate the noise variance.

The authors in [14] use the wavelet domain to decompose the frame into sub-bands. The coefficients of the diagonal details or the *HH* (*H*igh-*H*igh) band are used to estimate the noise variance. Wavelet decomposition isolates the high frequency noise in the *HH* band. Methods that use the wavelet domain are similar to the smoothing-based methods in overestimating the noise variance because the *HH* band has also high frequency frame information. Moreover, it is computationally demanding to transform every frame in the video sequence into the wavelet domain.

Block-based methods in [1] and [11] are less computationally demanding. These methods attempt to locate regions with the least amount of signal information. The intensity variations in these regions is assumed to be due to noise. The algorithm in [1] uses the variance to measure block homogeneity. The problem with this approach is that the variance is not always a reliable measure of homogeneity. The algorithm in [11] proposes a homogeneity test in which a number of high-pass operators are applied directionally. The variance of the noise is estimated from the local variances of the blocks selected to be the most homogeneous. The algorithm in [11], however, does not exploit the temporal information present in the video signal in the estimation process.

2.3 Proposed Approach

The proposed method attempts to estimate the global variance of the noise from the local variances of selected cubes in the video signal. The selected cubes have the common characteristic of being intensity homogeneous in the 2D or 3D space. Cube inhomogeneity is due to fine details and structures in the spatial domain, motion in the temporal domain or noise. The algorithm starts by dividing the 3D space defined by the video signal into cubic subspaces in an interpretation different from the one in [2] treating the video signal as a sequence of 2D images.

2.3.1 Local Homogeneity Measurement

Recalling (1.1) where we defined a noisy digital video signal V_η using

$$V_\eta = V + \eta. \tag{2.1}$$

A pixel in V_η is denoted by $V_\eta(i, j, n)$ where i and j are the spatial coordinates and n is the temporal coordinate. $\eta(i, j, n)$ is the amount of noise added to $V(i, j, n)$. Since the algorithm is designed to be context-free, there are no restrictions on the original signal V. The division of V_η into cubes C_{klm} with spatial indecies k and l and temporal index m is done using

$$\begin{aligned} C_{klm} &= \{V_\eta(i,j,n)|(i,j,n) \in \Psi_{klm}\}; \\ \Psi_{klm} &= \{(i,j,n)|k - \tfrac{W-1}{2} \le i \le k + \tfrac{W-1}{2}, \\ &\quad l - \tfrac{W-1}{2} \le j \le l + \tfrac{W-1}{2}, \\ &\quad m - \tfrac{W-1}{2} \le n \le m + \tfrac{W-1}{2}\} \end{aligned} , \tag{2.2}$$

where Ψ_{klm} is a cubic window of size W^3 ($W \in$ odd $\mathbb{Z}^+$) centered around the 3D point $(k,l,m) \in V_\eta$. To locate the homogeneous cubes in the video signal, we define a set of low-complexity homogeneity measures with (2.3). Theoretically, these measures represent the quantities in (2.4)-(2.8).

$$\{\zeta_D\}, D \in \{ST, T, S, VT, HT\}; \tag{2.3}$$

$$\zeta_{ST} = |\frac{\partial^2 V_\eta}{\partial i^2} + \frac{\partial^2 V_\eta}{\partial j^2} + \frac{\partial^2 V_\eta}{\partial n^2}|; \tag{2.4}$$

$$\zeta_T = |\frac{\partial^2 V_\eta}{\partial n^2}|; \tag{2.5}$$

$$\zeta_S = |\frac{\partial^2 V_\eta}{\partial i^2} + \frac{\partial^2 V_\eta}{\partial j^2}|; \tag{2.6}$$

$$\zeta_{VT} = |\frac{\partial^2 V_\eta}{\partial j^2} + \frac{\partial^2 V_\eta}{\partial n^2}|; \tag{2.7}$$

$$\zeta_{ZT} = |\frac{\partial^2 V_\eta}{\partial i^2} + \frac{\partial^2 V_\eta}{\partial n^2}|. \tag{2.8}$$

The proposed homogeneity measures are the magnitudes of 3 dimensional Laplacian operators. For (2.4)-(2.8) to be useful for digital video, they must be expressed in discrete form. For this purpose, we define the 3D masks in Fig. 2.1.

Fig. 2.1(a) is a 3D Laplacian operator used to measure spatio-temporal homo-

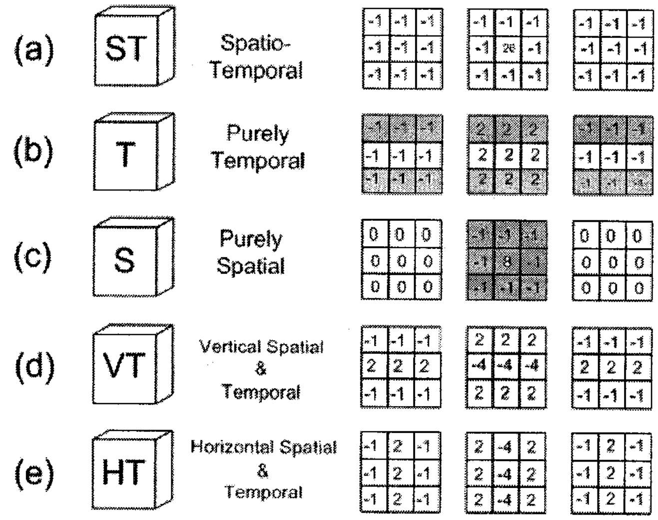


Figure 2.1: Homogeneity analyzer cubical masks where pixels in the same gray level belong to one plain.

geneity or ζ_{ST} in (2.4). The central coefficient of the mask (mask's 3D midpoint) can be calculated using $W^3 - 1$. The central coefficient accumulates to this value as a result of combining the 2^{nd} derivatives in all directions. The mask in Fig. 2.1(b) evaluates homogeneity along the temporal direction or ζ_T in (2.5). It acts as a local low-complexity motion detector. The mask in Fig. 2.1(c) is the spatial domain Laplacian operator. It measures purely spatial homogeneity or ζ_S defined in (2.6). This mask's response is an approximation of the sum of directional responses of the masks defined in [11]. The mask in Fig. 2.1(d) measures both the homogeneity along the spatial vertical direction and the temporal direction or ζ_{VT} in (2.7). Similarly, the mask in Fig. 2.1(e) measures the homogeneity along the spatial horizontal direction and the temporal direction or ζ_{HT} in (2.8).

To understand the motivation behind the proposed homogeneity analyzers, we review a number of commonly used structure (or edge) detectors in image and video segmentation literature. Generally, these detectors approximate the first or second derivatives. Some of these detectors are shown in Fig. 2.2.

The masks in Fig. 2.2(a) and (b) approximate the first derivative or the gradi-

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

(a) Prewitt

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

(b) Sobel

-1	-1	-1
-1	8	-1
-1	-1	-1

0	-1	0
-1	4	-1
0	-1	0

(c) Laplacian

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	2	-1	0
0	0	-1	0	0

(d) Laplacian of Gaussian

1	2	1
2	12	2
1	2	1

-1	-1	-1
-1	8	-1
-1	-1	-1

(e) Approximation of
Laplacian of Gaussian

Figure 2.2: Different discrete structure detecting masks.

ent. The main problem with these masks is that they are inherently directional. They respond to vertical or horizontal edges only which does not account for complex structures or motion patterns. The masks in Fig. 2.2(c) are Laplacian operators. They form the basis to the proposed homogeneity analyzers. These masks have a degree of rotational invariance to account for unpredictable object shapes or movements which means that they respond to change in different orientations. For example, the mask in Fig. 2.1(c) is 45^o rotation invariant (or isotropic) in the spatial domain.

This means that the masks response is invariant to 45^o rotations. Moreover, these masks are more sensitive to fine details and structures than the Sobel or Perwitt operators in Fig. 2.2(a) and (b). On the other hand, the disadvantages of using the Laplacian are threefold: 1) sensitivity to noise, 2) producing double edges and 3) inability to determine edge directions. Since the objective is to locate homogeneous areas, edge direction is of no interest. Similarly producing double edges is not an issue as no segmentation is needed. The most important problem facing the proposed masks is the sensitivity to noise. To overcome such sensitivity, the proposed method uses the Laplacian of Gaussian (LoG) filter which is the result of convolving a Gaussian smoothing filter with the Laplacian filter to produce the LoG filter with the continuous-time impulse response

$$h(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right]\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \tag{2.9}$$

$h(x,y)$ is sampled to produce the discrete-time mask in Fig. 2.2(d) which combines both Gaussian smoothing and Laplacian structure detection. Unfortunately, the smallest mask size that can be used to approximate the Laplacian of Gaussian is 5×5. Extending this mask to the 3^{rd} temporal dimension requires 5 frames delay which is a negative point in practical video processing systems. As an alternative approach, the masks in Fig. 2.2(e) are used to approximate the Laplacian of Gaussian. To study the effect of using these masks over the Laplacian, an expression of the Laplacian's sensitivity to noise is developed using Fig. 2.3 which shows a spatial 3×3 window of the original signal.

The response Y of applying the spatial Laplacian mask in Fig. 2.1(c) can be calculated with

$$\begin{bmatrix} G(1) & G(2) & G(3) \\ G(4) & G(0) & G(5) \\ G(6) & G(7) & G(8) \end{bmatrix} = \begin{bmatrix} X(1) & X(2) & X(3) \\ X(4) & X(0) & X(5) \\ X(6) & X(7) & X(8) \end{bmatrix} + \begin{bmatrix} N(1) & N(2) & N(3) \\ N(4) & N(0) & N(5) \\ N(6) & N(7) & N(8) \end{bmatrix}$$

Figure 2.3: Evaluation of sensitivity to noise in Laplacian operators

$$Y = 8G(0) - \sum_{i=1}^{8} G(i) \tag{2.10}$$

$$= 8(X(0) + N(0)) - \sum_{i=1}^{8} (X(i) + N(i)) \tag{2.11}$$

$$= \left[8X(0) - \sum_{i=1}^{8} X(i) \right] + (\sum_{i=1}^{8} N(i) \approx 0) + 8N(0). \tag{2.12}$$

$\left[8X(0) - \sum_{i=1}^{8} X(i) \right]$ measures the homogeneity of the window as $(\sum_{i=1}^{8} N(i) \approx 0)$ because the noise model is AWGN. We can see that the term $8N(0)$ can impair the Laplacian's response.

By applying the Gaussian mask in Fig. 2.2(e) before the Laplacian mask, $G(0) \approx X(0)$ and the effect of noise is reduced. At a later stage, robust estimation of the variance will be used to exclude cubes that pass the homogeneity test due to noise or other sources of outliers.

2.3.2 Homogeneous Cubes Selection

The quantities $\{\zeta_D\}$ in (2.4)-(2.8) are calculated for every cube C_{klm} by applying the 3D extensions of the masks in Fig. 2.2(e) to the video signal. Let U_D be the set of all selected homogeneous cubes based on ζ_D or

$$U_D = \left\{ C_{klm} | \min_{klm}^{L}(\zeta_D) \right\}, D \in \{ST, T, S, VT, HT\}, U_C = \bigcup_D U_D. \tag{2.13}$$

Note that (2.13) indicates that we are considering the set of the $L \in \mathbb{Z}$ most homogeneous cubes selected independently based on each ζ_D (i.e., ζ_S, ζ_T, ζ_{ST}, ζ_{HT} and ζ_{VT}). Since the five masks in Fig. 2.1 are used, the cardinality of the set of all selected cubes U_C equals to $5L$. L was fixed to 10% of the total number of blocks in [13] and [11]. In the proposed algorithm, L is variable and is computed by

$$L = L_{max} - \frac{\text{PSNR}_{init}}{\beta}, \tag{2.14}$$

where PSNR_{init} is the initial estimate of the PSNR_η calculated from the median of the variances of the 3 most homogeneous cubes over all ζ_D. Moreover, let σ^2_{init} be the corresponding initial estimate of the noise variance calculated from PSNR_{init} using (1.4). L_{max} is the maximum number of cubes to be used and β is a scaling factor. The choice of L_{max} is arbitrary between 5 and 30. In our simulations, L_{max} was set to 15 and $\beta = 5$. The function L can be replaced by any monochromically decreasing positive function of PSNR_{init} and is used to ensure the inclusion of more cubes in case of noisy video sequences and less cubes in case of less noisy ones. Using less cubes in case of less noisy videos results in a more reliable estimate. Homogeneity measures of a cube are not combined because a cube that is highly homogeneous temporally (low ζ_T) can be spatially non-homogeneous (high ζ_S).

After homogeneous cubes are selected, we calculate their sample mean and variance along the plains (pixels with the same gray level in Fig. 2.1) found to be most homogeneous. For all cubes in U_S, we use

$$\begin{aligned} \mu_S &= \frac{\sum_{(i,j)\in\Psi_{kl}} V_\eta(i,j)}{W^2}; \\ \sigma_S^2 &= \frac{\sum_{(i,j)\in\Psi_{kl}} (V_\eta(i,j)-\mu_S)^2}{W^2-1} \end{aligned}, \tag{2.15}$$

where Ψ_{kl} indicates that we use only pixels along the middle spatial plain of the cube (pixels in the same gray level in Fig. 2.1(c)). The set of all local variances calculated spatially from cubes in U_S is denoted $U_S^{\sigma^2}$. For cubes found to be temporally homogeneous we use

$$\begin{aligned} \mu_{T_\rho} &= \frac{\sum_{(i,n)\in\Psi_{km}} V_\eta(i,n)}{W^2}; \\ \sigma_{T_\rho}^2 &= \frac{\sum_{(i,n)\in\Psi_{km}} (V_\eta(i,n)-\mu_{T_\rho})^2}{W^2-1} \end{aligned}, \tag{2.16}$$

where Ψ_{km} indicates that we use only pixels along temporal plains (pixels in the same gray level in Fig. 2.1(b)). Using (2.16), we calculate the sample mean μ_{T_ρ} and variance $\sigma_{T_\rho}^2$ along each plain $\rho = \{1, .., W\}$ and then compute the average over the W plains. It is important that the noise variance is estimated using only plains found to be homogeneous as we have no information about the homogeneity along other plains. Following the same notation, the set of all local variances calculated temporally from cubes in U_T is denoted $U_T^{\sigma^2}$. For cubes that are chosen to be spatio-temporally most homogeneous (i.e., $U_{ST} \bigcup U_{HT} \bigcup U_{VT}$), the sample mean and variance are calculated over all pixels in the cube using

$$\begin{aligned} \mu_{ST,VT,HT} &= \frac{\sum_{(i,j,n)\in\Psi_{klm}} V_\eta(i,j,n)}{W^2}; \\ \sigma_{ST,VT,HT}^2 &= \frac{\sum_{(i,j,n)\in\Psi_{klm}} (V_\eta(i,j,n)-\mu_{ST,VT,HT})^2}{W^3-1} \end{aligned}. \tag{2.17}$$

The corresponding sets of variances calculated spatio-temporally are denoted $U_{ST}^{\sigma^2}$, $U_{HT}^{\sigma^2}$ and $U_{VT}^{\sigma^2}$

2.3.3 Robust Estimation using the Least Median of Squares

The dimension-based (i.e., spatial, temporal and spatio-temporal) noise variances are robustly estimated from $U_D^{\sigma^2}$. Robustness is defined in [15] as the ability to deal with the possible consequences of deviations from the assumed statistical model. In computer vision literature, M-*Estimators* and Least Median of Squares (LMS) are the most commonly used robust estimators [16]. The overall noise variance estimate from a set $U_D^{\sigma^2}$, $\widehat{\sigma}_D^2$, is calculated using the LMS robust estimator as

$$\widehat{\sigma}_D^2 = \underset{\sigma_p^2 \in R}{\operatorname{argmin}} \underset{\sigma_{D\alpha}^2 \in U_D^{\sigma^2}}{\operatorname{median}} |\sigma_p^2 - \sigma_{D\alpha}^2| \tag{2.18}$$

where R is given by

$$R = \left[\sigma_{init}^2 - \frac{\sigma_{th}^2}{2} : \frac{\sigma_{th}^2}{Q} : \sigma_{init}^2 + \frac{\sigma_{th}^2}{2}\right] \tag{2.19}$$

where σ_p^2 is a variance in R and $\sigma_{D_\alpha}^2$ is a variance in $U_D^{\sigma^2}$. Eq. (2.19) states that the values of σ_p^2 in (2.18) are varied between $\sigma_{init}^2 - \sigma_{th}^2/2$ and $\sigma_{init}^2 + \sigma_{th}^2/2$ in steps of $\frac{\sigma_{th}^2}{Q}$ as illustrated in Fig. 2.4. Q is the search step size and can be varied between 5 and 15. It controls the accuracy versus the complexity or the number of search steps. Larger Q means more computations but more accurate estimation and vice versa. σ_{th}^2 is set to correspond to a PSNR value of 2.75dB. The breakdown point is defined as the maximum percentage of outliers that can be injected into the assumed model before it fails (deviates largely from the expected behavior). The breakdown point measures the robustness to outliers of an estimator. The LMS is used because it has the highest breakdown point of 0.5. The mean and eventually any least squares based estimator has a breakdown point of 0. Which means that a single outlier can impair the estimation result as opposed to 50% outliers in case of the LMS.

To speed up the median calculation process, the proposed method uses an algorithm that calculates the median without resorting to sorting. When calculating the

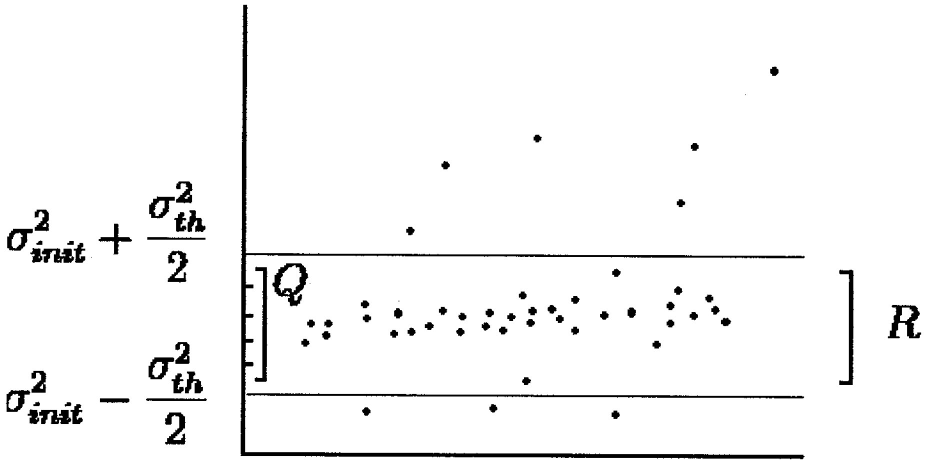


Figure 2.4: Deployment of robust Least Median of Squares estimator in the proposed algorithm

median, the fact that it has less than half the data smaller than it, and less than half the data larger than it is utilized. The procedure starts by taking the first value of the data and counting the number of elements Ns in the rest of the set that are smaller than the first value and the number of elements Nb that are larger. If $Ns \neq Nb$, the first value is not the median. If $Ns < Nb$, then we do not need to consider any value less than the first value because we already know that the median is larger than the first value. On the other hand, if $Ns > Nb$, then we do not need to consider any value larger than the first value because we are sure the median is smaller than the first value. The search proceeds until the median is found when $Ns = Nb$ for a specific value. This procedure is similar in nature and complexity to linear search. This means we reduce the complexity (number of search elements) of median calculation from in the best case scenario to $O(n)$.

The efficiency of median calculation increases with larger L. For smaller L, the median can be used instead which can be expressed by

$$\widehat{\sigma}_D^2 = median(\sigma_{D\alpha}^2), \quad \sigma_{D\alpha}^2 \in U_D^{\sigma^2}. \tag{2.20}$$

Using (2.20), the quantities $\widehat{\sigma}^2_S$, $\widehat{\sigma}^2_T$, $\widehat{\sigma}^2_{ST}$, $\widehat{\sigma}^2_{HT}$ and $\widehat{\sigma}^2_{VT}$ are calculated.

The frame-wise noise variance is then estimated using the domain-wise noise variances using

$$\widehat{\sigma}^2_\eta = \frac{1}{N_D} \sum_D \hat{\sigma}^2_D, \tag{2.21}$$

where N_D is the number of domain-wise estimates used. We only include in the averaging process the domain-wise estimates that do not exceed σ^2_{init} by more than σ^2_{th} to account for the case of complete estimation failure in a given domain.

2.4 Simulations

To evaluate the performance of the algorithm, estimation error defined to be the absolute difference between the true value of the variance of noise σ^2_η and the estimated value $\widehat{\sigma^2}_\eta$, or $E = |\sigma^2_\eta - \widehat{\sigma^2}_\eta|$, is used. The estimation error average μ_E and variance σ^2_E are computed using (2.22)

$$\mu_E = \frac{\sum_{i=1}^{N} E(i)}{N}; \quad \sigma^2_E = \frac{\sum_{i=1}^{N} (E(i) - \mu_E)^2}{N-1}, \tag{2.22}$$

where N is the total number of test frames used. While μ_E measures the performance of a noise estimation algorithm, σ^2_E measures the reliability of that performance. The standard video sequences *Prlcar, Tennis, Train, Football, Car and Flowergarden* were corrupted with 20, 30 and 40 dB AWGN. Simulation was run on the first 50 frames of each sequence using $W = 3$ cubic windows. Average time needed for the proposed and referenced algorithms was measured and the Time Ratio (TR) between them was calculated accordingly. Implementation was using C++ under an Intel(R) Xeon(TM) CPU 2.40GHz machine running Linux. The proposed method was found to be faster than all referenced methods except [11].

Table 2.1 shows that the proposed algorithm has the most reliable performance for different noise levels. Figs. 2.5, 2.6, 2.7, 2.8, 2.9 and 2.10 show the individual estimation error for the test sequences used for the proposed and referenced methods at different noise levels. The proposed method produces less error for all sequences and noise levels.

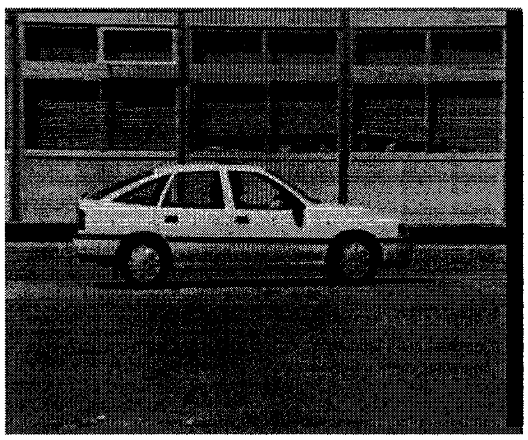
Fig. 2.11 shows the average estimation error over time, μ_E, and estimation error standard deviation, σ_E^2, averaged over all test sequences for every noise level. As can be seen from Fig. 2.11, the proposed method gives a lower average estimation error than referenced methods and is temporally stable. It also shows that the reliability of the proposed method is better than referenced methods for all noise levels.

Table 2.1: The average and the standard deviation of the estimation error for 20, 30 and 40 dB noise.

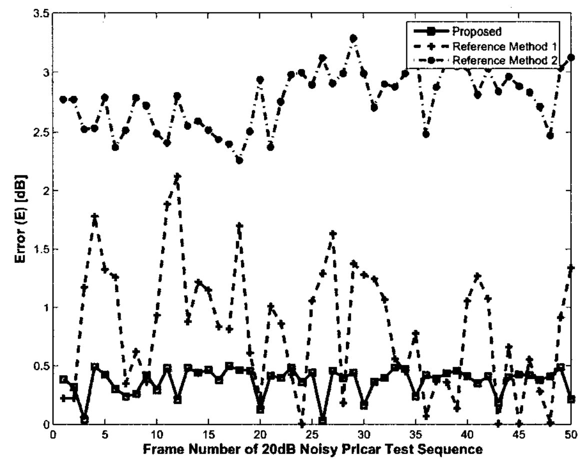
	20 dB		30 dB		40 dB		
Alg.	μ_E	σ_E	μ_E	σ_E	μ_E	σ_E	TR
Inter-frame							
Proposed	0.23	0.33	0.50	0.41	0.65	0.68	1.0
[2]	2.80	0.77	2.53	1.19	3.1	5.78	1.5
Intra-frame							
[13]	1.99	1.20	3.21	1.42	4.34	1.70	4.5
[14]	1.75	1.26	2.12	1.81	3.36	2.70	2.2
[1]	0.79	1.13	1.01	1.20	1.10	1.24	2.3
[11]	1.60	1.55	2.39	1.25	1.91	1.16	0.6

2.5 Summary

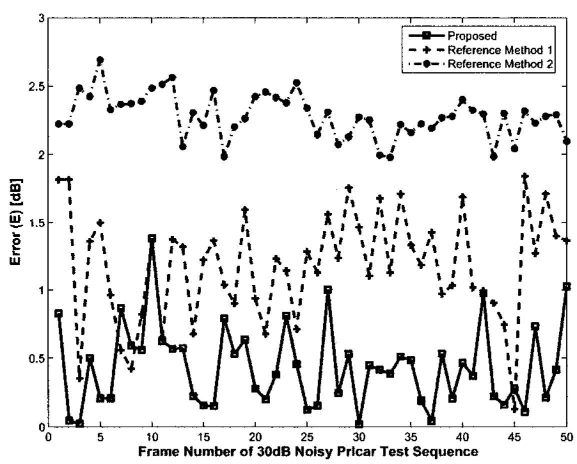
This chapter proposed a technique in which the variance of the AWGN noise is estimated from selected homogeneous cubes in the 3D video signal. Spatial, temporal and spatio-temporal homogeneity are measured using 3D Laplacian of Gaussian operators. The noise variance is estimated from the local variances of selected homogeneous cubes calculated along intensity uniform plains. Least Median of Squares



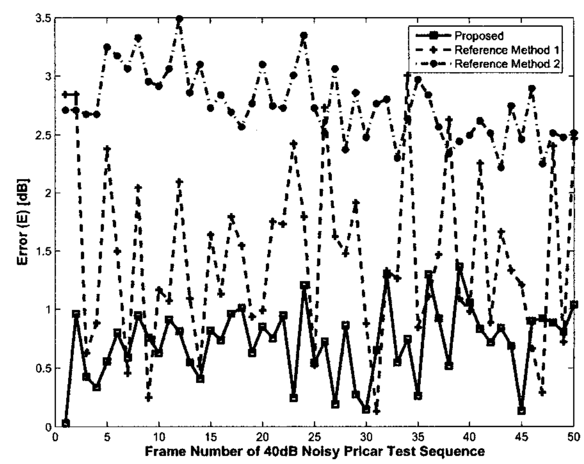
(a) Prlcar



(b) 20dB Comparison



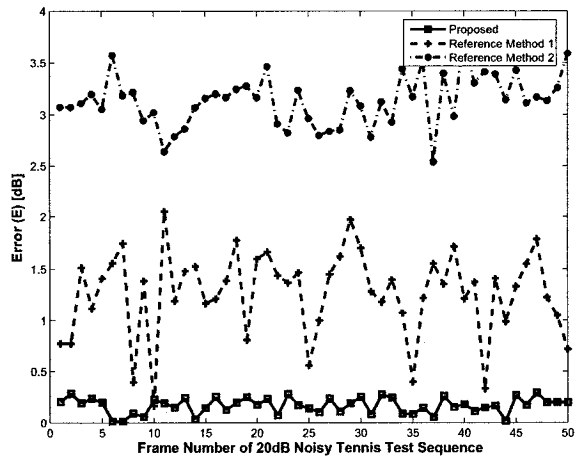
(c) 30dB Comparison



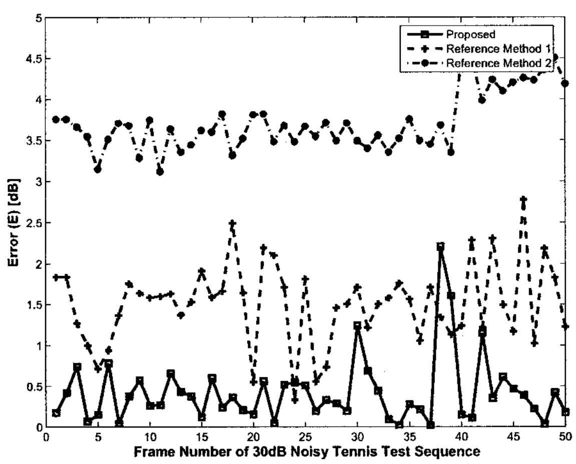
(d) 40dB Comparison

Figure 2.5: Estimation error over time for proposed and referenced methods (Reference 1 [1] and Reference 2 [2]) for 20dB, 30dB and 40dB noisy *Prlcar* test sequence.

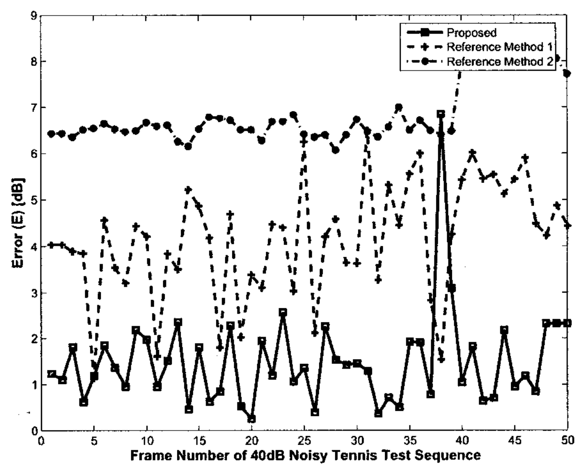
(a) Tennis



(b) 20db Comparison



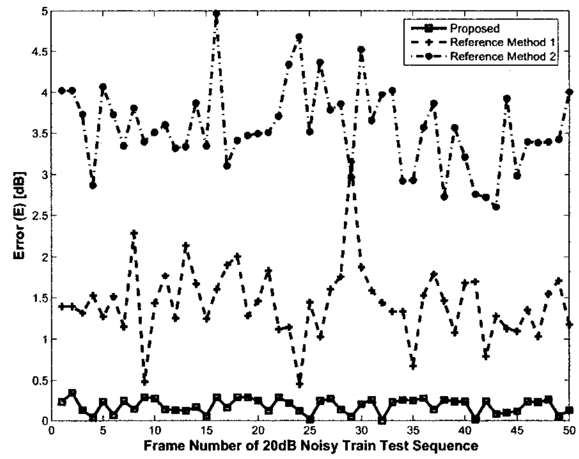
(c) 30db Comparison



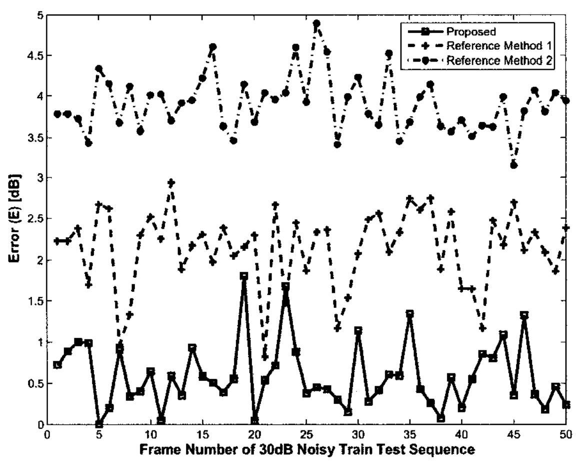
(d) 40db Comparison

Figure 2.6: Estimation error over time for proposed and referenced methods (Reference 1 [1] and Reference 2 [2]) for 20dB, 30dB and 40dB noisy *Tennis* test sequence.

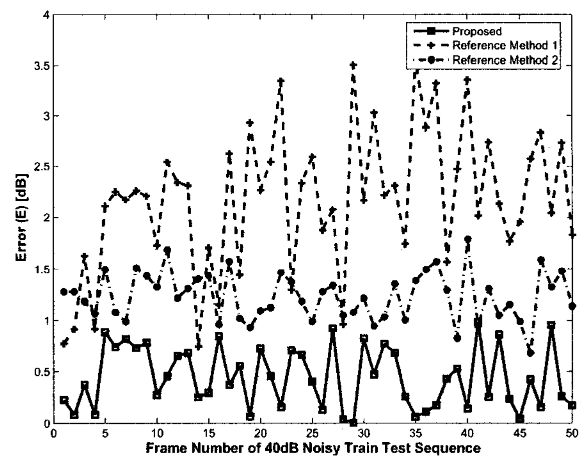
(a) Train



(b) 20db Comparison



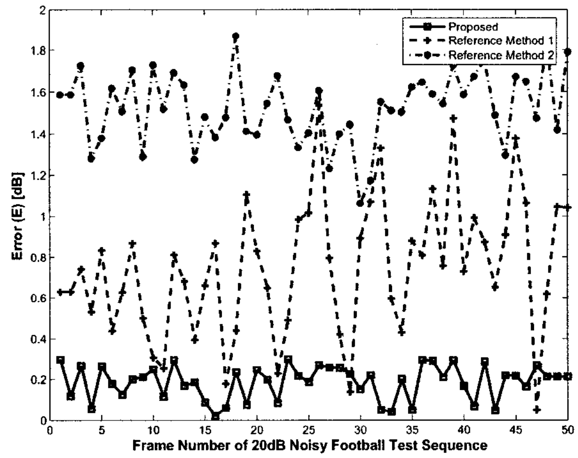
(c) 30db Comparison



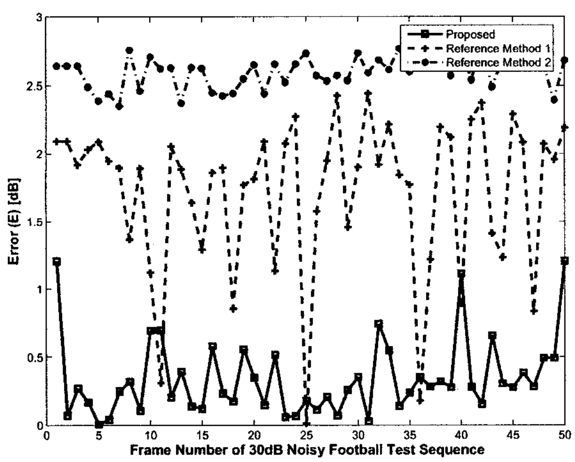
(d) 40db Comparison

Figure 2.7: Estimation error over time for proposed and referenced methods (Reference 1 [1] and Reference 2 [2]) for 20dB, 30dB and 40dB noisy *Train* test sequence.

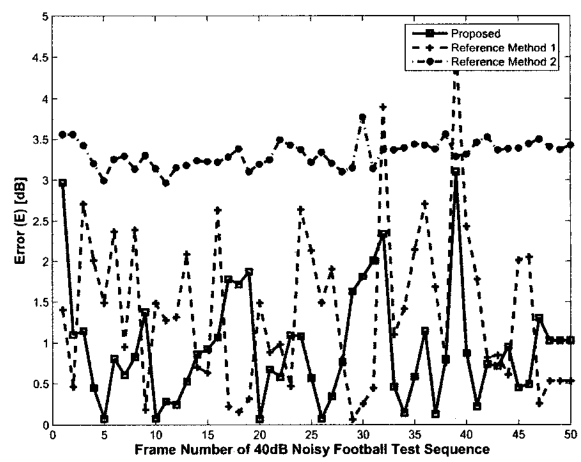
(a) Football



(b) 20db Comparison



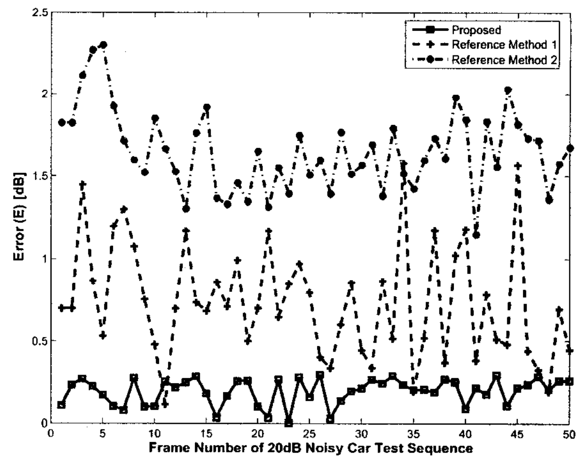
(c) 30db Comparison



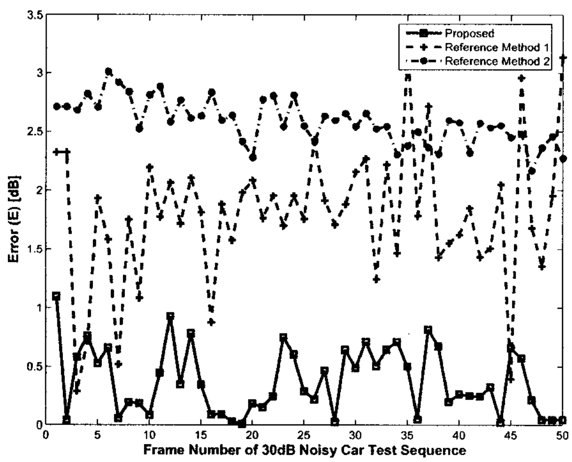
(d) 40db Comparison

Figure 2.8: Estimation error over time for proposed and referenced methods (Reference 1 [1] and Reference 2 [2]) for 20dB, 30dB and 40dB noisy *Football* test sequence.

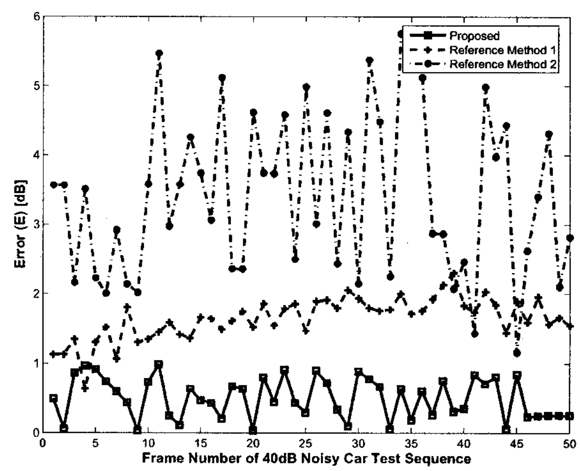
(a) Car



(b) 20db Comparison

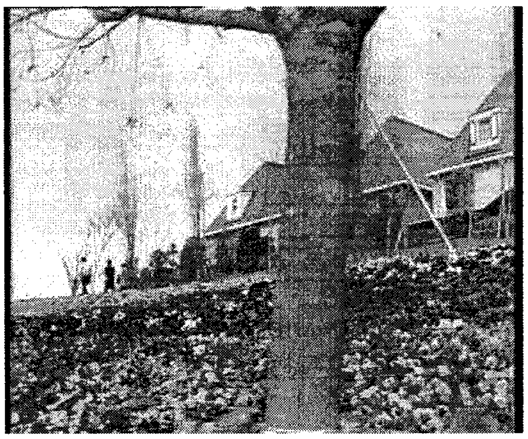


(c) 30db Comparison

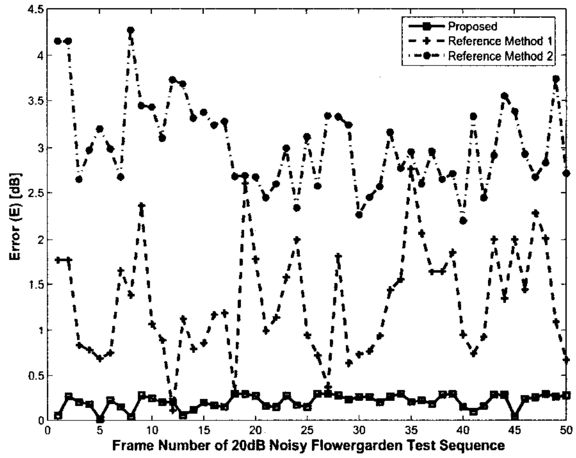


(d) 40db Comparison

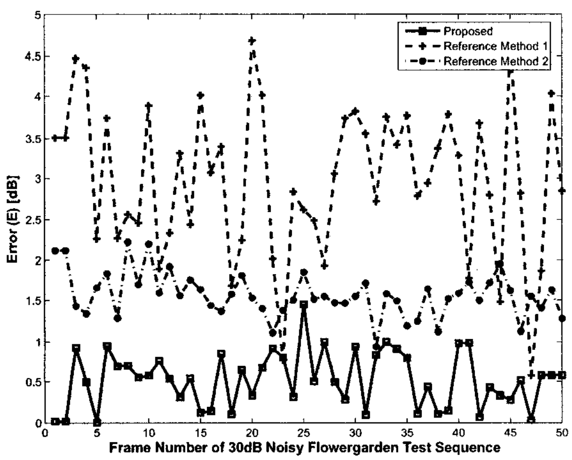
Figure 2.9: Estimation error over time for proposed and referenced methods (Reference 1 [1] and Reference 2 [2]) for 20dB, 30dB and 40dB noisy *Car* test sequence.



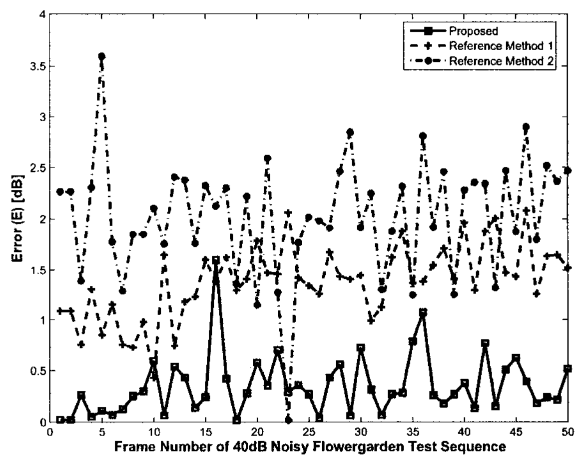
(a) Flowergarden



(b) 20db Comparison

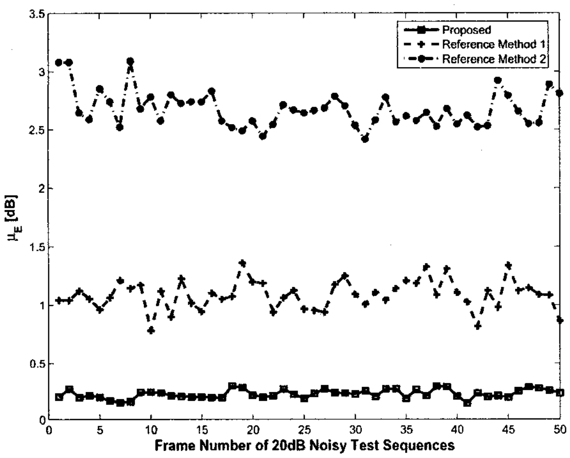


(c) 30db Comparison

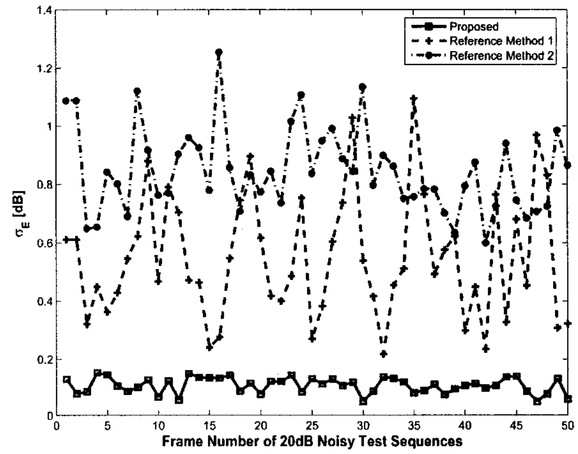


(d) 40db Comparison

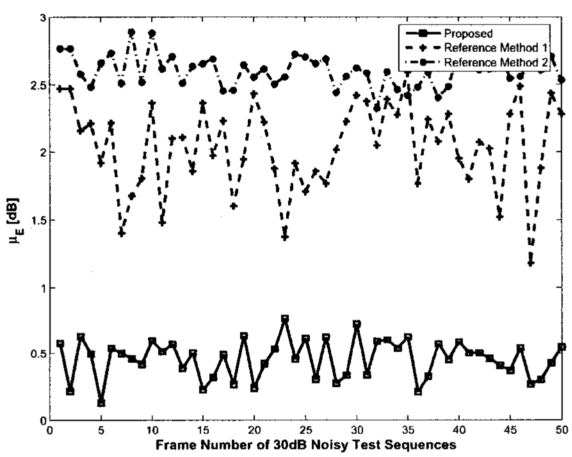
Figure 2.10: Estimation error over time for proposed and referenced methods (Reference 1 [1] and Reference 2 [2]) for 20dB, 30dB and 40dB noisy *Flowergarden* test sequence.



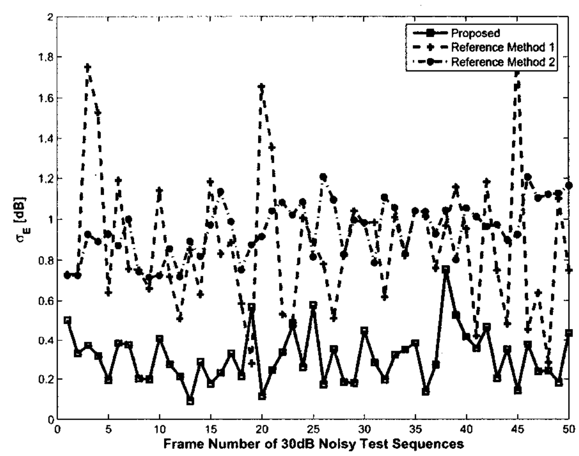
(a) Mean of Error for 20dB



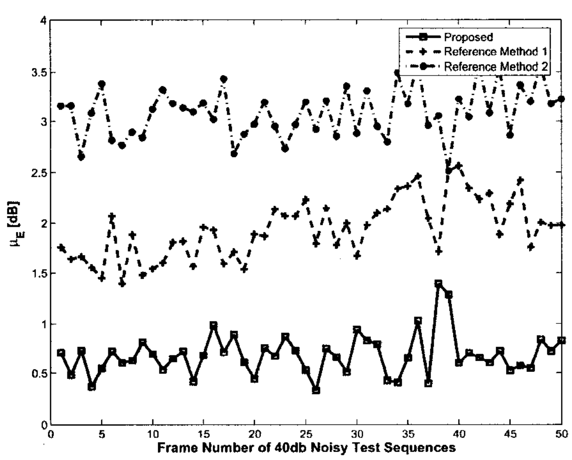
(b) Standard Deviation of Error for 20dB



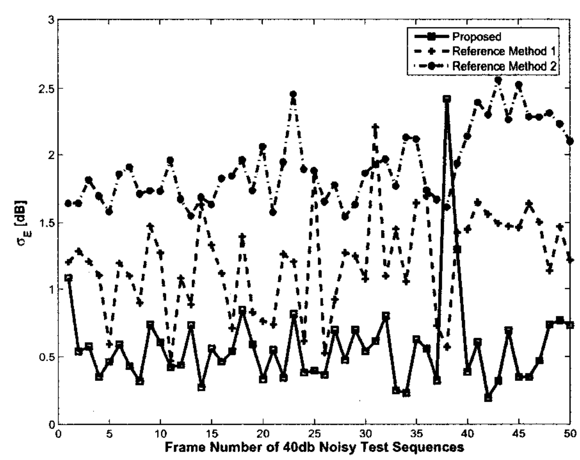
(c) Mean of Error for 30dB



(d) Standard Deviation of Error for 30dB



(e) Mean of Error for 40dB



(f) Standard Deviation of Error for 40dB

Figure 2.11: Mean μ_E and standard deviation σ_E of error over time for proposed and referenced methods (Reference 1 [1] and Reference 2 [2]) for 20dB, 30dB and 40dB noisy test sequences.

(LMS) robust estimators are deployed to calculate the domain-wise (spatial, temporal and spatio-temporal) noise variance estimate. The domain-wise noise variance estimates are averaged to obtain the frame-wise final noise variance estimate. The proposed algorithm works well for video sequences with high structure and motion activity. It performs reliably with different noise levels with a maximum estimation error of 1.7 dB.

Chapter 3

Spatial Adaptive Multi-Directional Noise Reduction

3.1 Introduction

AWGN is evenly distributed over the frequency domain (i.e., white noise), whereas an image contains mostly low frequency information. Hence, the noise is dominant in high frequencies and its effects can be reduced using lowpass filtering performed using frequency domain or spatial domain filters. Often a spatial domain filter (or a spatial filter) is preferable, as it is computationally less expensive and faster than a frequency domain filter making it attractive for real-time video processing applications.

Lowpass filtering an image or a video signal leads to suppression of fine details and structures or blurring as shown in Fig. 3.1. The Human Visual System (HVS) is sensitive to high frequencies and can easily visualize blurring. For this purpose adaptive lowpass filters have been developed. Adaptive lowpass filtering is performed in the spatial, temporal or spatio-temporal domains using linear or non-linear operators. Motion is the biggest challenge facing filtering in the temporal domain. Therefore, it must be either detected and adapted for or estimated and compensated for. With

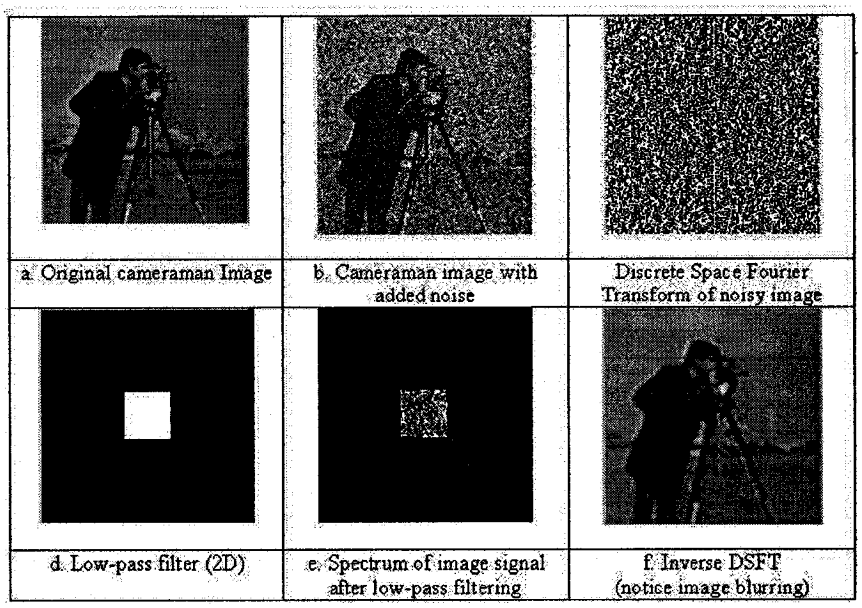


Figure 3.1: Lowpass filtering and the blurring side effect

motion detection, filtering is performed in areas where small or no motion is detected. The disadvantage is that little, if any, noise reduction is performed in strongly moving areas. In motion estimation, filtering is performed along motion trajectories. The disadvantage is the computational cost associated with motion estimation. Moreover, noise impairs the motion estimation process. Temporal filtering based on impaired motion information can result in a more visually objectionable signal than the original noisy signal. Also, only global motion can be measured with a degree of reliability. Spatio-temporal filters suffer from the same disadvantages as temporal filters. Moreover, spatio-temporal filters such as a 3D Wiener filter suffer from the fact that the wide-sense stationarity assumptions are virtually never true because of moving objects. For the above reasons, spatial filtering is still more preferable than temporal filtering in the absence of accurate motion information. More information about temporal and spatio-temporal filters can be found in [7].

In this chapter, a framework for multi-directional noise filtering and improved noise filters: multi-directional Wiener and Sigma filters are proposed. The improved filters aim at limiting filtering to pixels within the same population or region to avoid cross population filtering which leads to blurring. With the proposed framework, the limiting behavior of spatial filters is improved based on the assumption that pixels are spatially grouped into homogeneous (intensity invariant) regions in natural images or video frames. To demonstrate the proposed framework, we propose a spatially multi-directional Sigma (Section 3.4) and Wiener (Section 3.5) filters the advantage of which, relative to existing methods, is increased gain and better preservation through adapting the filter's selectivity to image content and noise levels.

The remainder of this chapter is as follows. Section 3.2 summarizes the related work in the field. Section 3.3 generalizes directional filtering to propose multi-directional filtering and its adaptation to image structure and noise level. Section 3.4 presents the proposed multi-directional Sigma filter and Section 3.5 the proposed multi-directional Wiener filter. Objective simulation results are presented and discussed in Section 3.6. Finally, Section 3.7 concludes the paper.

3.2 Related Work

Filters in [2–4, 17–19] are spatial filters intended for real-time video processing. The method in [4] introduces a spatial filter that is based on the Sigma probability of the Gaussian Distribution. This filter, known as the Sigma filter, is widely used as a benchmark for testing spatial adaptive filters against. It selects which pixels to include in or exclude from the filtering process based on the estimated noise level. The drawback of the Sigma filter is that it does not impose any constraints on the pixels that pass the Sigma probability test. Thus, any neighboring pixel in a block surrounding the processed pixel whose value is within $2\sigma_\eta$ away from the value of the processed pixel is included in the filtering process. The filter in [3] introduces a criteria to measure homogeneity and a homogeneity-based filter and applies filtering to the intensity most homogeneous direction. The work presented in [17] and [2], defines a real-time recursive Sigma filter that adds a number of modifications to the Sigma filter. These modifications include changing the shape of the block, making the filtering recursive and using a tri-level weighting function. The filter in [18] solves a global variational optimization problem making it slower than other real-time methods. The filter in [19] replaces the center pixel by the average of the direction which has minimum variance. The problem with [19] is that the variance is not a reliable measure of homogeneity in the presence of noise [11].

Filters in [5, 20–25] are used in offline application where high gain is more needed than high algorithmic speed. A classical approach to spatial noise reduction is the adaptive spatial Wiener filter in [5]. The drawback of this Wiener filter is despite producing high PSNR gain, it suffers from residual blurring in the image or video near edges [7]. The work in [20] builds a modified Sigma filter with a larger block than the recursive Sigma filter and an improved weighting function. These modifications give better gain at the expense of lower performance in less noisy images and slower execution. The filter in [21] is another filter to trade speed with higher gains by proposing a weighting function that depends on an optimized parameter causing the filter to be significantly slower than other filters. The method in [22] uses Gauss curvature driven diffusion and is computationally expensive. Recent methods that use the wavelet domain for denoising such as the ones in [23–26] are also computationally expensive.

Noise filters work better if they are adaptive to image content, e.g., using directional or rational filters. Directional filters [3, 19], apply filtering to selected directions (see Fig. 3.2) within a local image block and rational filters [27–29] modulate the coefficients of a linear low-pass filter to limit its actions in the presence of image details.

Rational filters can be applied directionally for increased preservation of image content. For example, [28] is applied to the 0^o, 45^o, 90^o and 135^o directions. The filter in [28] does not adapt to more complex structures such as corners. Moreover, filter parameters are set manually to tune the filter's response. The work in [28] is extended in [29] with the rational filter applied temporally. The method requires multiple frame delays and still adapts parameters manually.

Noise filters can also be adapted to noise level for increased noise reduction gain. In the remainder of this paper, adaptation is to image content and noise level.

3.3 Theory

We propose a framework for multi-directional noise filtering and also propose improved noise filters: multi-directional Wiener and Sigma filters. The resulting filters aim at limiting filtering to pixels within the same population or region to avoid cross population filtering which leads to blurring. With the proposed framework, the limiting behavior of spatial filters is improved based on the assumption that pixels are spatially grouped into homogeneous (intensity invariant) regions in natural images or video frames. To demonstrate the proposed framework, we propose a spatially multi-directional Sigma (Section 3.4) and Wiener (Section 3.5) filters the advantage of which, relative to existing methods, is increased gain and better preservation through adapting the filter's selectivity to image content and noise levels.

In AWGN, noisy pixels are assumed to be independent identically distributed (iid) Gaussian. Accordingly, the noisy image or video frame F_η is modeled as

$$F_\eta = F + \eta. \tag{3.1}$$

where F is the noise-free image or video frame and η is the added noise. The problem now is to obtain an approximation of F, or $\hat{F}$, given F_η only.

3.3.1 Generalizing Directional Filtering to Include Multiple Directions

Image and video frames have the characteristic that spatial correlations between sample pixel values exist. These correlations are created by the surfaces of video objects in the scene. Spatially uncorrelated video noise can be reduced with spatial averaging, which corresponds to low-pass filtering in the frequency domain. With low-pass filtering, however, structured areas with fine details like edges and corners are blurred.

To overcome the blurring side effect, directional filtering can be used in which filtering is performed along edges and fine details and not across them. The detection of image structure is done in [3] using intensity-homogeneity analyzer masks applied along eight candidate directions in a block (see Fig. 3.2).

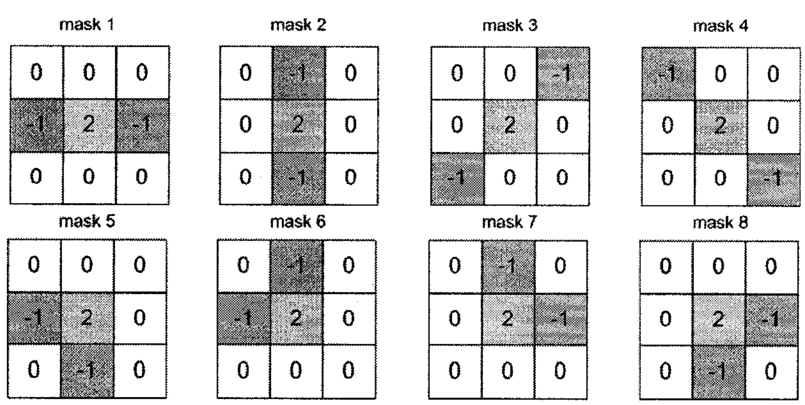


Figure 3.2: Homogeneity Analyzer Masks in [3].

These analyzers work as directional second order Laplacian operators with the coefficients $\{-1, -1, \ldots, W-1, \ldots, -1, -1\}$, where W denotes the block size and is a positive odd integer. For example, when $W = 3$, mask coefficients are $\{-1, 2, -1\}$. Let $z \in \{1, 2, ..., 8\}$ be the mask index. Each mask measure homogeneity, ζ, along a direction. Since each mask represents a direction, we use z to identify a direction as well. Let ζ_z denote the response of mask z. ζ_z is close to zero for pixels with close intensity values. Let $S = \{\zeta_z\}$ be the set of mask responses. To generalize directional filtering to include multiple directions, we start by defining $z_{min_1} = \underset{z}{\operatorname{argmin}}\,(S)$ to be the index of the most homogeneous direction. Accordingly, the index of the second most homogeneous direction is $z_{min_2} = \underset{z}{\operatorname{argmin}}\,(S - \{z_{min_1}\})$ and the index of the n^{th} most homogeneous direction is

$$z_{min_n} = \underset{z}{\operatorname{argmin}}\,(S - \bigcup_{d=1}^{n-1}\{z_{min_d}\}). \tag{3.2}$$

Let $D = (z_{min_1}, z_{min_2}, ..., z_{min_8})$ denote the ordered sequence of mask indexes based on ζ. The size of the sequence D is $|D| = 8$. Let $1 \leq D_n \leq |D|$ be the number of directions used in filtering. For example, when $D_n = 2$, the most (e.g., z_{min_1}) and second most (e.g., z_{min_2}) homogeneous directions are used. In Fig. 3.3, we define a set of spatial masks $G_W(z)$ used to isolate a direction from a block of size W. Note

$$G_3(1)\ \begin{bmatrix}0&0&0\\1&0&1\\0&0&0\end{bmatrix}\quad G_3(2)\ \begin{bmatrix}0&1&0\\0&0&0\\0&1&0\end{bmatrix}\quad G_3(3)\ \begin{bmatrix}1&0&0\\0&0&0\\0&0&1\end{bmatrix}\quad G_3(4)\ \begin{bmatrix}0&0&1\\0&0&0\\1&0&0\end{bmatrix}\quad G_3(5)\ \begin{bmatrix}0&0&0\\0&0&1\\0&1&0\end{bmatrix}\quad G_3(6)\ \begin{bmatrix}0&0&0\\1&0&0\\0&1&0\end{bmatrix}\quad G_3(7)\ \begin{bmatrix}0&1&0\\1&0&0\\0&0&0\end{bmatrix}\quad G_3(8)\ \begin{bmatrix}0&1&0\\0&0&1\\0&0&0\end{bmatrix}$$

Figure 3.3: Spatial mask $G_3(z)$ for $z \in \{1, 2, \ldots, 8\}$.

that the value assigned to the center pixel is zero in all $G_W(z)$ or any combination $(G_W(z_i) + G_W(z_j))$ of them. In other words, we also use $G_W(z)$ to exclude the central pixel from the direction so as to assign a separate weighting to it. Fig. 3.3 depicts $G_3(z)$ for $W = 3$ and $z \in \{1, 2, \ldots, 8\}$.

With D and $G_W(z)$, we generalize directional filtering to include multiple directions with

$$
\begin{aligned}
\hat{F}(i,j) &= \frac{F_\eta(i,j) + \sum\limits_{a=1-W}^{W-1} \sum\limits_{b=1-W}^{W-1} \omega(a,b) F_\eta(i+a,j+b)}{\sum\limits_{a=1-W}^{W-1} \sum\limits_{b=1-W}^{W-1} \omega(a,b)}, \\
\omega(a,b) &= \sum_{d=1}^{D_n} G_W(z_{min_d}), \\
\sum_{a=1-W}^{W-1} \sum_{b=1-W}^{W-1} \omega(a,b) &= \sum_{a=1-W}^{W-1} \sum_{b=1-W}^{W-1} \left[\sum_{d=1}^{D_n} G_W(z_{min_d}) \right] \\
&= D_n \times (W-1) + 1,
\end{aligned}
\tag{3.3}
$$

where $F_\eta(i,j)$ is the pixel at spatial coordinate (i,j) of the noisy image or video frame and $\hat{F}(i,j)$ is the noise-reduced pixel. The spatial mask $\omega(a,b)$ is the result from combining two or more spatial masks $G_W(z)$. Multiplying $F_\eta(i+a,j+b)$ by $\omega(a,b)$ isolates a subset from the $W \times W$ population by choosing pixels that make up the D_n most homogeneous directions. For example, if $D_n = 1$ and $z_{min_1} = D(1) = 1$, (i.e., the horizontal direction is detected to be the most homogeneous), (3.3) reduces to

$$
\begin{aligned}
\hat{F}(i,j) &= \frac{F_\eta(i,j) + F_\eta(i,j-1) + F_\eta(i,j+1)}{3}, \\
\omega(a,b) &= G_3(z_{min_1}) = G_3(1), \\
\omega(a,b) &= \sum_{a=-2}^{2} \sum_{b=-2}^{2} G_3(1) = D_n \times (W-1) + 1 \\
&= 1 \times 2 + 1 = 3
\end{aligned}
\tag{3.4}
$$

We can express specific directional filters such as [3] and [19] using (3.3). For example, we can obtain [3] from (3.3) using fixed block size, $W = 3$, and a fixed number of directions, $D_n = 1$ and obtain [19] by setting $W = 3$, $D_n = 1$ but with a different S. In [19], $S = \{\sigma_z^2\}$ where σ_z^2 is the variance of pixels in direction z. [3] and [19] are adapted to image content by filtering in one direction only and their order will always be 3. With 3-tap filter, small gain can be achieved especially at high noise levels.

3.3.2 Proposed Framework for Adaptive Multi-Directional Filtering

We propose to improve the adaptivity to image content of directional filtering by combining directions using $\omega(a,b)$ and to introduce adaptivity to noise level to (3.3) to optimize the filtering process. First we vary W and D_n depending on σ_η^2 which is the noise variance in F_η estimated using the approach in [11]. W_η is the noise-adaptive block size and D_{n_η} the noise-adaptive number of directions. We also assign pixel weights based on the noise level. We define C_η as the weight of the center pixel and $C'_\eta = \frac{1-C_\eta}{D_{n_\eta}(W_\eta - 1)}$ as the weight of the non-center pixels where pixel weightings sum to 1 to preserve the signal mean. The aim is to increase W_η and D_{n_η} and decrease C_η for noisier images to achieve higher gain and vice versa to combat blurring. The function for the noise-adaptive multi-directional filtering based on (3.3) is

$$\hat{F}(i,j) = \frac{C_\eta F_\eta(i,j) + \sum\limits_{a=1-W_\eta}^{W_\eta-1} \sum\limits_{b=1-W_\eta}^{W_\eta-1} \omega(a,b) F_\eta(i+a, j+b)}{\sum\limits_{a=1-W_\eta}^{W_\eta-1} \sum\limits_{b=1-W_\eta}^{W_\eta-1} \omega(a,b)},$$

$$\omega(a,b) = \sum_{d=1}^{D_{n_\eta}} G_W(z_{min_d}) C'_\eta, \tag{3.5}$$

$$\sum_{a=1-W_\eta}^{W_\eta-1} \sum_{b=1-W_\eta}^{W_\eta-1} \omega(a,b) = C'_\eta D_{n_\eta}[W_\eta - 1] + C_\eta.$$

To demonstrate the efficacy of the proposed adaptation in (3.5), the effect of enlarging the block and increasing the number of directions is studied. It can be shown that the output noise variance, $\sigma^2_{\eta_o}$ in $\hat{F}$, after applying (3.5) is

$$\sigma^2_{\eta_o} = C^2_\eta \sigma^2_\eta + \frac{(1-C_\eta)^2}{D_{n_\eta}(W_\eta - 1)} \sigma^2_\eta. \tag{3.6}$$

The noise reduction gain $R[dB]$ as a function of W_η, D_{n_η} and C_η is

$$R_{(W_\eta, D_{n_\eta}, C_\eta)}[dB] = 10 \log \frac{\sigma^2_\eta}{\sigma^2_{\eta_o}} = \tag{3.7}$$
$$10 \log \left(\frac{D_{n_\eta}(W_\eta - 1)}{\left[D_{n_\eta}(W_\eta - 1) + 1\right] C^2_\eta - 2C_\eta + 1} \right).$$

Fig. 3.4 compares $R_{(3,1,C_\eta)}$ (the gain for block size 3×3 and using the most homogeneous direction only) to $R_{(5,1,C_\eta)}$ and $R_{(5,2,C_\eta)}$. As can be seen, enlarging W and increasing D_n produce an increased (theoretical) gain, $R[dB]$.

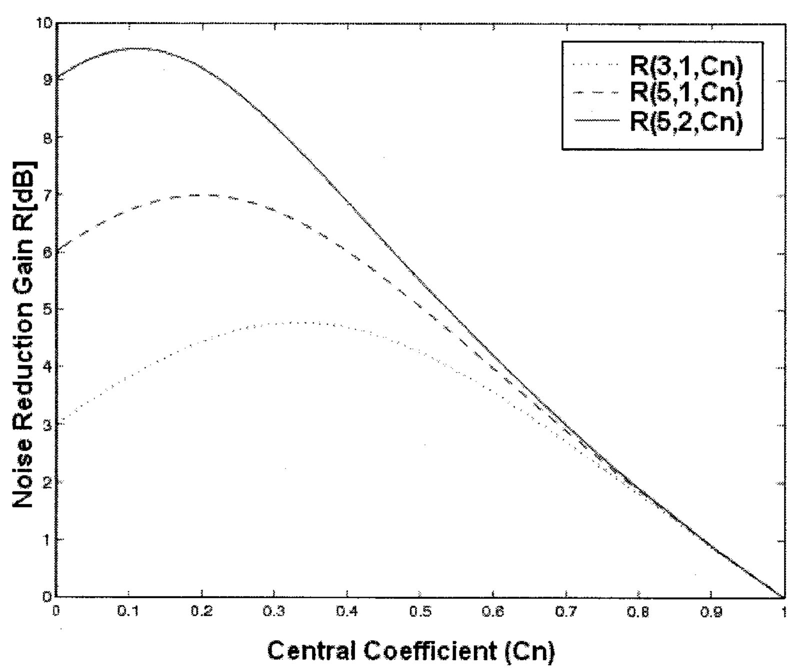


Figure 3.4: The relationship between the central coefficient C_η and the noise reduction gain R [dB] for $\{W_\eta = 3\ D_{n_\eta} = 1\}$, $\{W_\eta = 5\ D_{n_\eta} = 1\}$ and $\{W_\eta = 5\ D_{n_\eta} = 2\}$.

On the other hand, Fig. 3.5 shows that using a larger block and more directions introduce blurring in less noisy images. This is evident as filtering with $W_\eta = 3$ and $D_{n_\eta} = 1$ is compared to filtering with $W_\eta = 5$ and $D_{n_\eta} = 1$. The performance decrease in the $W_\eta = 5,\ D_{n_\eta} = 1$ filter after PSNR = 28dB is due to blurring. In the next Sections, we propose adaptive multi-directional Sigma and Wiener filters to increase the noise reduction gain and the structure preservation.

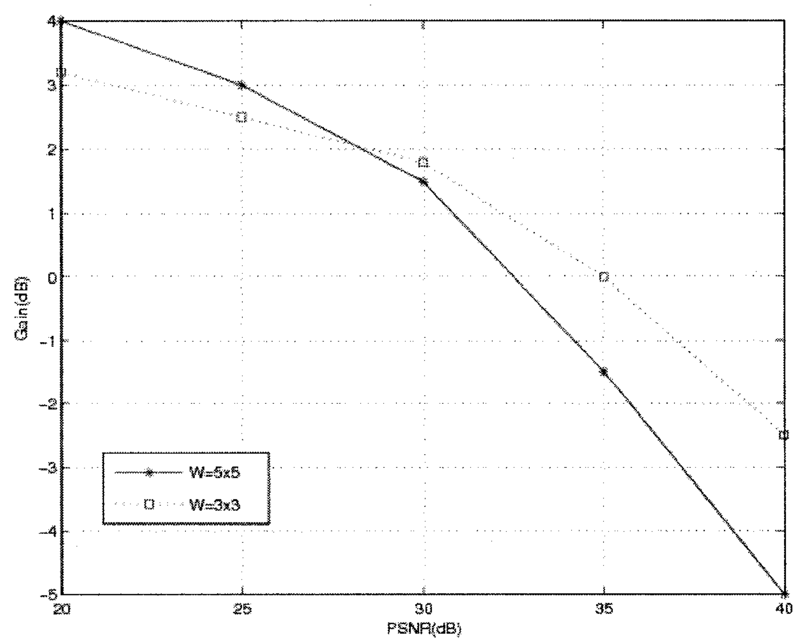


Figure 3.5: Average gain at different noise levels for a $\{W_\eta = 3,\ D_{n_\eta} = 1\}$ and a $\{W_\eta = 5,\ D_{n_\eta} = 1\}$ blocks.

Now we propose a procedure by which the optimum W_η, D_{n_η} and C_η can be determined. Optimality here refers to maximizing the gain $R[dB]$ in (3.7) in PSNR or

$$R(\hat{\text{PSNR}}_\eta, \text{PSNR}_\eta) = \hat{\text{PSNR}}_\eta - \text{PSNR}_\eta \tag{3.8}$$

where $\hat{\text{PSNR}}_\eta$ denotes the PSNR of the noise-reduced image or video frame $\hat{F}$ and PSNR_η is the PSNR of the noisy image or video frame. $\hat{\text{PSNR}}_\eta$ is either measured in the presence of F or estimated in its absence. Finding the optimal W_η and D_{n_η} can be treated as a quantization problem. For W_η and D_{n_η}, we divide the region of support B of the PSNR$_\eta$, B = [PSNR$_\eta^{min}$, PSNR$_\eta^{max}$] = [20dB, 55dB], into intervals $B_l = [b_{l-1}, b_l)$ where $l = \{0, 1, 2, ..., L\}$ and L is the number of possible block sizes or directions. Then, we assign a level (block size or number of directions) g_l from possible block sizes (e.g., $g_l \in \{3, 5, 7, 9\}$) or possible number of directions (e.g., $g_l \in \{1, 2, ..., 8\}$). Optimal g_l and b_l [30] which completely define W_η or D_{n_η} can be found by solving

$$B_l = \{\mathrm{PSNR}_\eta : R_{g_l}(\hat{\mathrm{PSNR}}_\eta, \mathrm{PSNR}_\eta) \leq R_{g_{\bar{l}}}(\hat{\mathrm{PSNR}}_\eta, \mathrm{PSNR}_\eta)\} \tag{3.9}$$

$$g_l = \underset{g_l}{\mathrm{argmax}}\ E\{R_{g_l}(\hat{\mathrm{PSNR}}_\eta, \mathrm{PSNR}_\eta) | \mathrm{PSNR}_\eta \in B_l\} \tag{3.10}$$

where R_{g_l} denotes the gain in PSNR for using $W_\eta = g_l$ or $D_{n_\eta} = g_l$. Eqs. (3.9) and (3.10) are known in quantization theory as the generalized nearest neighbor and generalized centroid conditions, respectively. With absence of information about the pdf of the PSNR$_\eta$, Llyod algorithm for optimal scalar quantization [31] can be used on a representative set of training data. Finding the optimal set of filter coefficients C_η and C'_η can be treated as an optimal prediction problem. The solution of such problem is given by

$$C_\eta = [A]^{-1} r \tag{3.11}$$

where $[A]$ is the correlation matrix between non-center pixels and r is the correlation vector between the center pixel and all non-center pixels.

3.3.3 Adaptive Multi-Directional Sigma and Wiener Filters

In Sections 3.4 and 3.5, we demonstrate the extendability of the proposed adaptive multi-directional filtering in (3.5) by proposing adaptive multi-directional Sigma and Wiener filters. In both filters, the block size W_η, the number of directions used in filtering D_{n_η} and pixel weights C_η are adapted to noise level σ_η^2. Since the proposed

Sigma filter is intended for real-time systems, W_η, D_{n_η} and C_η are designed to be fast to compute. Contrary to the Sigma filter, the Wiener filter is slow due to the computation of local variances. As a result, the proposed Wiener filter is meant for offline systems. Both the proposed Sigma and Wiener adapt W_η the same way. However, in the proposed Wiener filter C_η is adapted based on the Wiener estimate and it controls D_{n_η}.

3.4 Proposed Adaptive Multi-Directional Sigma Filter

3.4.1 Principle Idea

The Sigma filter [4] is based on the Sigma probability of the Gaussian distribution. It smooths the image noise by averaging neighborhood pixels which have intensity values within a fixed Sigma range of the center pixel.

We propose an adaptive multi-directional Sigma filter that combines the D_{n_η} most homogeneous directions in a block and creates with $\omega(a,b)$ in (3.5) a kernel that best fits the image structure and then excludes using a noise adaptive $G_W(z)$ the pixels that fail the Sigma probability test to further adapt the kernel shape to noise level. In details, the filter:

- Populates the ordered sequence of mask indexes D based on the homogeneity of the mask's direction ζ_z using (3.2) and adapt D_{n_η} to the noise level σ_η^2 using

$$D_{n_\eta} = \begin{cases} \kappa & : \sigma_\eta > t_\eta \\ 1 & : otherwise \end{cases} . \tag{3.12}$$

  In simulation, $\kappa = 2$ is used to maintain the fast algorithmic speed of the filter.

- Adapts the block size W_η to σ_η^2 as

$$W_\eta = \begin{cases} W_h & : \sigma_\eta > t_\eta \\ W_l & : otherwise \end{cases}, \tag{3.13}$$

where W_h ranges from 5 to 9 and W_l from 3 to $W_h - 2$. We used $W_h = 5$ and $W_l = 3$ in simulations.

- Selects t_η to combine the benefits of both block sizes $W_\eta = 3$ and $W_\eta = 5$. In Fig. 3.5, the point the curves of the $W_\eta = 3$ and $W_\eta = 5$ block sizes intersect corresponds to a noise level of 28 dB ($\sigma_n = 10.13$) which is a used as t_η in simulations.

- Adapts pixel weights C_η in (3.5) as follows

$$C_\eta = \frac{\text{PSNR}_\eta}{\text{PSNR}_\eta^{max}}, \quad \text{PSNR}_\eta = 10\log_{10}\left(\frac{255^2}{\sigma_\eta^2}\right), \qquad \text{PSNR}_{max} = 55\text{dB}. \tag{3.14}$$

- Changes the spatial mask $G_W(z)$ to $G_W(z, \sigma_\eta)$ based on the Sigma probability to make pixel selection noise level adaptive as follows

$$G_W(z,\sigma_\eta) = \begin{cases} G_W(z) & : F_\eta(k,l) - 2\sigma_\eta \leq F_\eta(i,j) \leq F_\eta(k,l) + 2\sigma_\eta \\ 0 & : otherwise \end{cases}, \tag{3.15}$$

where $F_\eta(k,l)$ denotes the neighboring pixel in the combined D_{n_η} most homogeneous directions or $F_\eta(i+a, j+b)$ in (3.5).

3.4.2 Performance Analysis and Comparison

In Section 3.3.2 Fig. 3.4, we demonstrated the benefits of the proposed adaptive multi-directional filtering in (3.5) in terms of the gain in quality (or PSNR_η). The multi-directional Sigma filter inherits these benefits. With the enhanced (noise-adaptive) selection function $G_W(z, \sigma_\eta)$ defined in (3.15), we can now analyze other aspects of the proposed filter such as structure preservation. First, we use the Modulation Transfer Function (MTF) [32] to illustrate this structure-preservation capabilities relative to non-adaptive methods [4] and [3]. Second, we show, using local histograms, why the use of the Sigma probability to adapt the spatial mask $G_W(z, \sigma_\eta)$ resulted in improved preservation of structure. Third, we examine the change in the size, shape (as controlled by combining the D_{n_η} most homogeneous directions and Sigma test in (3.15)), and weighting of the filter block as a result of the added modification. Finally, we propose a solution to ensure that the added modification do not decrease the speed of the proposed filter.

To approximate the MTF from a local region Z in a single image or video frame F, let $H(x)$ be the normalized local histogram calculated from pixels in Z as

$$H(x) = \frac{K_x}{K}, \; 0 \leq x \leq 255, \tag{3.16}$$

where K_x is the number of pixels with gray level $x \in Z$ and K is the total number of pixels in Z. $H(x)$ approximates the probability density function $p(x)$ of region $Z \subset F$. By finding the cumulative sum of $H(x)$ with

$$P(x) = \sum_{0}^{x} H(x), \; 0 \leq x \leq 255 \tag{3.17}$$

we get an estimation of the cumulative distribution function (CDF) or $P(x)$ of region Z [33]. $P(x)$ is related to the Edge-Spread Function (ESF(x)), defined as the response of the system to an ideal edge [34], by $P(x) = x(ESF)$ when it is known that the locality or region Z on which the histogram was calculated represents an edge. MTF measures the degradation of object contrast due to blurring as a function of the spatial frequency ω. It can be calculated as the Fourier transform, $\mathcal{F}\{.\}$, of the first derivative of the ESF, that is,

$$MTF(\omega) = \mathcal{F}\{\frac{\partial ESF(x)}{\partial x}\}. \tag{3.18}$$

Fig. 3.6 shows the MTF of the local region Z in Fig. 3.7 (*Cameraman* image) for the proposed multi-directional Sigma filter versus the filters in [4] and [3]. The proposed Sigma filter achieves a higher contrast transfer ratio than [4] and [3] due to using (3.12)-(3.15).

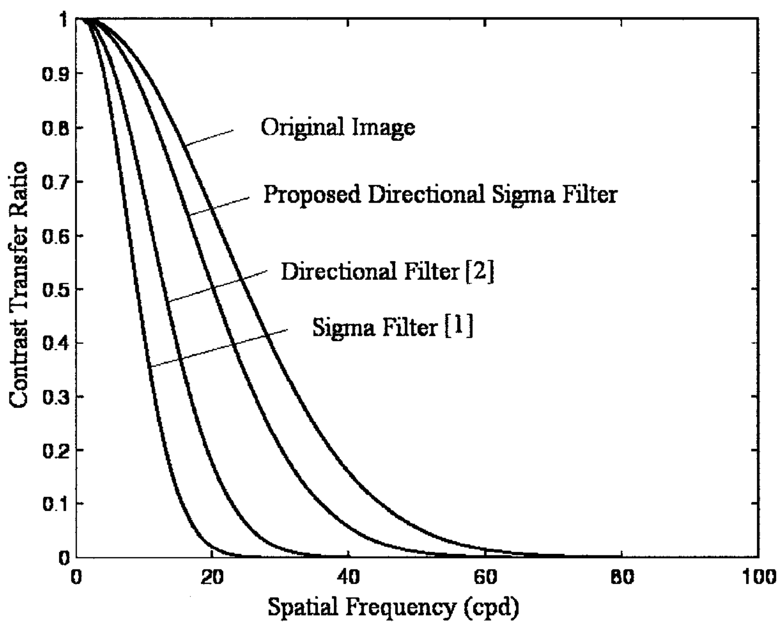


Figure 3.6: Estimation of Modulation Transfer Function (MTF) of region Z in Fig. 3.7 for the proposed multi-directional Sigma and referenced methods.

We will use the local histograms shown in Fig. 3.7 to explain the improved structure-preservation capabilities of the proposed Sigma filter. For unstructured regions X and Y in Fig. 3.7, the noise reduction gain, $R_{(W,D_N,C)}$, can be calculated from (3.7). The local histogram at area Z shows two distinct sample populations. The rectangles superimposed on the *Cameraman* image are magnified blocks. If filtering takes place over pixels in region Z, the assumption being that they belong to the same population, blurring will occur. The multi-directional application of the Sigma probability will act as a low-pass filter that will isolate one population so that no cross population averaging takes place, hence, effectively reducing the blurring side effect. This is shown in Fig. 3.8.

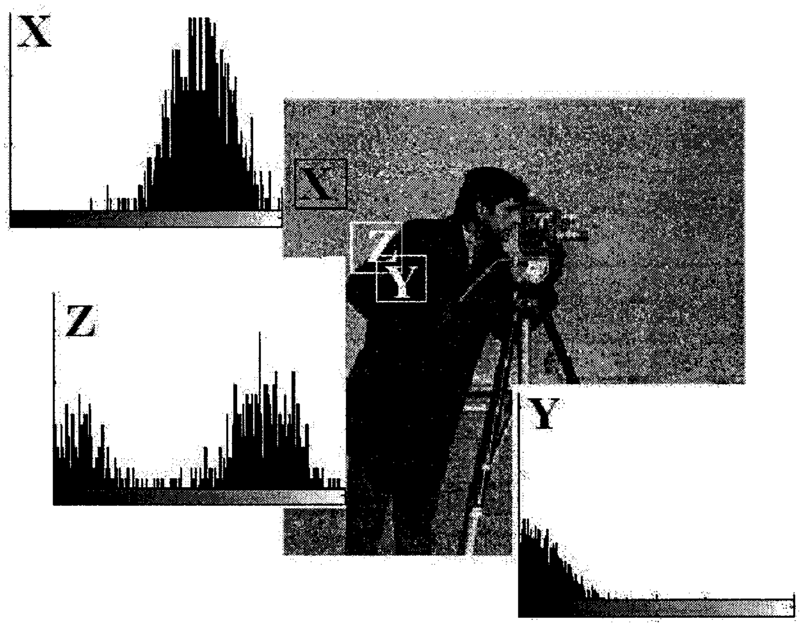


Figure 3.7: Local histograms at structured and unstructured areas of the noisy *Cameraman* (see Fig. 3.8).

Moreover, the improved structure-preservation capabilities of the proposed multi-directional Sigma filtering over the Sigma filter [4] is due to the fact that two criteria are used in the proposed method as opposed to one in the Sigma filter. The proposed method imposes an extra constraint that pixels have to be spatially grouped in a direction rather than scattered as in the Sigma filter. The proposed method ensures exclusion of pixels which randomly satisfy the Sigma probability without being in the

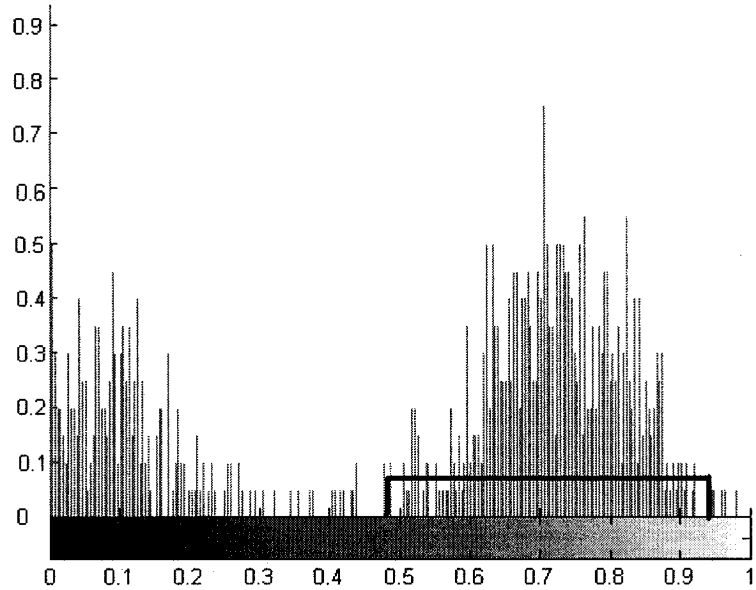


Figure 3.8: The multi-directional Sigma probability isolates a sample population to prevent blurring while filtering. Assuming the block on region Z in Fig. 3.7 is centered around a high-intensity pixel.

same population as the center pixel.

Fig. 3.9 shows the change in the shape of the block at different noise level ranges between the proposed Sigma filter and [3]. In Fig. 3.9(a,b), all pixels in the homogeneous direction are assumed to belong to one population and are included in the filtering process. Fig 3.9(c,d) illustrates how the proposed filter adapts block size, shape and weighting to frame and noise characteristics.

In the proposed Sigma filter, the selection of the parameters W_η, D_{n_η}, C_η and $G_W(z)$ determines its speed of computation. We select these parameters to ensure real-time performance. For example, the filter is designed to work with $W_\eta = W_h$ (see (3.13)) with the outermost pixels given extra binary weights of 1 or 0 based on σ_η^2. This way those pixels can be turned off completely to reduce the block size to W_l without complexity.

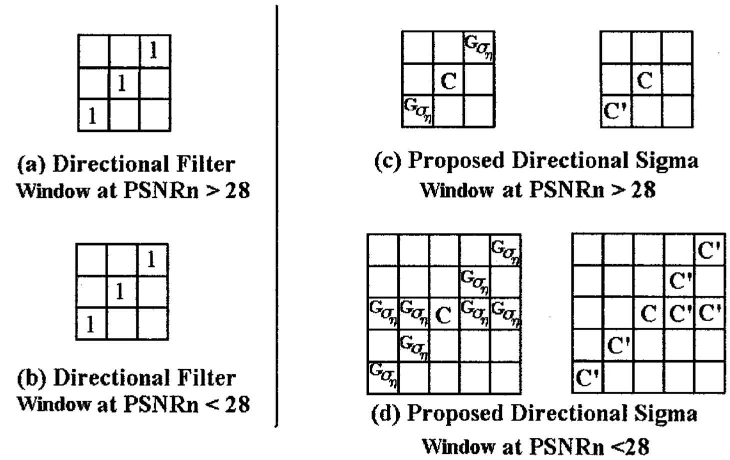


Figure 3.9: Fixed directional filter [3] block size versus proposed variable block size, shape and weighting for the multi-directional Sigma filter at different noise levels.

3.5 Proposed Multi-Directional Wiener Filtering

A classical variant of the spatial Wiener filter is based on the Minimum Mean Square Estimate (or Wiener estimate) of $F(i,j)$ in terms of $F_\eta(i,j)$

$$F(i,j) = \frac{\sigma_F^2}{\sigma_{F_\eta}^2}(F_\eta(i,j) - \mu_{F_\eta}) + \mu_{F_\eta}, \tag{3.19}$$

where μ_{F_η} is the local mean of a block of size W centered at coordinates (i,j) in F_η, $\sigma_{F_\eta}^2$ is the corresponding local variance in F_η and σ_F^2 is the same local variance but in F. Recall that $F(i,j)$ is a pixel in the noise-free image or video frame F, $F_\eta(i,j)$ is the noisy pixel in F_η and $\hat{F}(i,j)$ is the noise reduced pixel in $\hat{F}$.

In the classical Wiener filter, an estimate of σ_F^2 (or $\hat{\sigma}_F^2$) is obtained from the noisy image F_η is using the Maximum Likelihood (ML) estimator as

$$\hat{\sigma_F^2} = \max(\sigma_{F_\eta}^2 - \sigma_\eta^2, 0). \tag{3.20}$$

By combining (3.19) and (3.20), we get the pixel-wise spatial Wiener filter input output

$(F_\eta(i,j),\hat{F}(i,j))$ relationship as

$$\hat{F}(i,j) = \frac{\hat{\sigma_F^2}}{\sigma_{F_\eta}^2}(F_\eta(i,j) - \mu_{F_\eta}) + \mu_{F_\eta}. \tag{3.21}$$

3.5.1 Proposed Adaptation

We propose an adaptive multi-directional Wiener filter by applying multi-directionally the Wiener filter in (3.21) and defining the mean, μ_z, along a direction z (see Fig. 3.2) as

$$\mu_z = \frac{\sum_{a=1-W_\eta}^{W_\eta-1} \sum_{b=1-W}^{W_\eta-1} \omega(a,b) F_\eta(i+a, j+b)}{\sum_{a=1-W_\eta}^{W_\eta-1} \sum_{b=1-W_\eta}^{W_\eta-1} \omega(a,b)}. \tag{3.22}$$

where $\omega(a,b)$ is given by (3.3). Similarly, the variance, σ_z^2, along a direction z in Fig. 3.2 is

$$\sigma_z^2 = \frac{\sum_{a=1-W_\eta}^{W_\eta-1} \sum_{b=1-W_\eta}^{W_\eta-1} \omega(a,b)(F_\eta^2(i+a, j+b) - \mu_z^2)}{\sum_{a=1-W_\eta}^{W_\eta-1} \sum_{b=1-W_\eta}^{W_\eta-1} \omega(a,b)}. \tag{3.23}$$

Thus, the input output $(F_\eta(i,j),\hat{F}(i,j))$ relationship of the proposed multi-directional Wiener filter based on (3.22) and (3.23) is

$$\hat{F}(i,j) = \frac{1}{|D|} \sum_{z=1}^{|D|} \frac{\max(\sigma_z^2 - \sigma_\eta^2, 0)}{\sigma_z^2} \times (F_\eta(i,j) - \mu_z) + \mu_z. \tag{3.24}$$

In (3.22)-(3.24), we use the block size adaptation W_η as in (3.13) and the pixel weight, C_η, is given by

$$C_\eta = \frac{1}{W} - \frac{\max(\sigma_z^2 - \sigma_\eta^2, 0)}{W\sigma_z^2} \tag{3.25}$$

The number of directions to include in the filtering process, D_{n_η}, is decided by the

Wiener weighting function in (3.24). If the direction is homogeneous (small ζ_z), the local directional variance σ_z^2 will be less than the estimated noise variance σ_η^2 causing (3.24) to reduce to $\hat{F}(i,j) = \mu_z$. If σ_z^2 is larger than σ_η^2, indicating a direction with high activity, (3.24) will reduce to $\hat{F}(i,j) = F_\eta(i,j)$ which will ensure that no filtering takes place along that direction. This way D_N varies depending on both the level of structure and the level of noise. This behavior is different from that of the Wiener filter in (3.21) where the entire block is considered in the local mean and variance calculation causing outlier pixels to be included in the averaging process which will eventually increase blurring.

3.5.2 Filter Analysis and Comparison

Here, we analyze the proposed multi-directional Wiener filter as follows: we first use the Modulation Transfer Function (MTF) (defined in Section 3.4.2) to illustrate the degradation of contrast due to blurring by the proposed multi-directional Wiener filter relative to the Wiener filter [5] and comment on the reason behind the improved structure preservation. Then, we examine the change in the size, shape and weighting of the filter block as a result of the proposed structure and noise adaptations.

Fig. 3.10 shows the MTF of the local region Z in Fig. 3.7 for the proposed multi-directional Wiener filter versus the Wiener filter. The proposed multi-directional Wiener filter achieves a higher contrast transfer ratio than the Wiener filter. This is because the number of used directions is adapted to image content and noise level. This means that filtering is adjusted to full filtering action in case of a totally homogeneous region and to no filtering in case of a highly-structured region.

Fig. 3.11 shows the change in the shape of the block at different noise level ranges between the directional filter in [3] and the proposed multi-directional Wiener filter. In Fig. 3.11(a,b), all pixels in the homogeneous direction are assumed to belong to

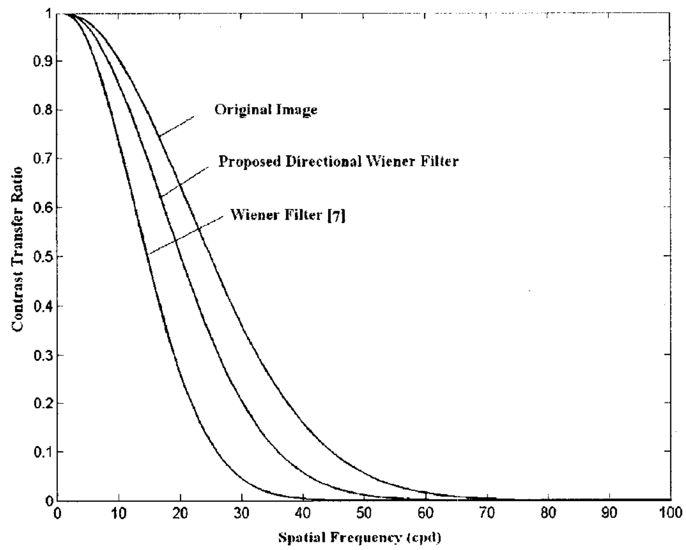
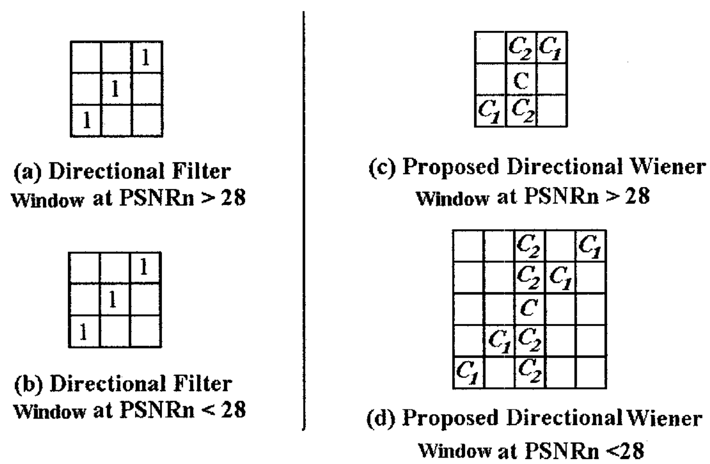


Figure 3.10: Modulation Transfer Function of region Z in Fig. 3.7 for proposed multi-directional Wiener and Wiener filter.

one population and are included in the filtering process. Fig. 3.11(c,d) illustrates how the block size and weighting are adapted to frame and noise characteristics in the proposed multi-directional Wiener filter.



$$C = \frac{1}{W} - \frac{\max(\sigma_Z^2 - \sigma_\eta^2, 0)}{W\sigma_\eta^2}$$

Figure 3.11: Fixed directional filter [3] block versus proposed variable block size, shape (number of directions) and weighting at different noise levels for the multi-directional Wiener filter.

3.6 Results

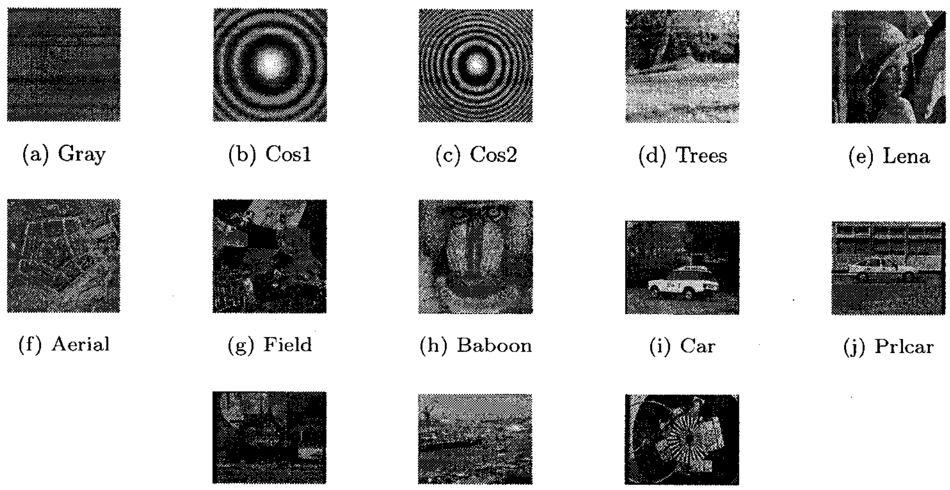
To experimentally evaluate the proposed framework, we use the criteria time complexity (Section 3.6.1), temporal stability in PSNR gain (Section 3.6.2), PSNR gain at different noise levels for the proposed multi-directional Sigma filter (Section 3.6.3) and the proposed Wiener filter (Section 3.6.4). (Note that in Figs. 3.6 and 3.10 we have shown the relative structure preservation capabilities of the proposed filters relative to referenced methods in terms of the MTF.)

To validate the proposed approach, 8 images (Fig 3.12(a-h)) and 5 video sequences (Fig. 3.12(i-m)) were used in simulation. The images were selected to represent different levels of structure. The video sequences were selected to represent different types of video motion and levels of structure. Video sequences *Car* and *Prlcar* represent tracking camera motion. Video sequence *Train* represents translational object motion with fixed camera. Video sequence *Wheel* represents rotational object motion and video sequence *Kiel* represents zoom motion. The images and video sequences were corrupted by noise levels ranging from 20-40 db PSNR in steps of 5 dB and the video sequences with levels 20-40 in steps of 10 db.

For related work comparison, we implemented and compared the proposed Sigma filter with real-time filters [2–4] and the proposed Wiener filter with computationally expensive filters [5,21]. We compare with [21] because it attempts to optimize filtering adaptively in the same manner as the classical Wiener filter.

3.6.1 Time Complexity

Table 3.1 shows the time (averaged over images in Fig. 3.12(a-h) for all noise levels) needed by the referenced methods compared to the time needed by the proposed multi-directional Sigma method to process a 512 $\times$ 512 image when implemented using C++ under an Intel(R) Xeon(TM) CPU 2.40GHz machine running Linux. The



(a) Gray (b) Cos1 (c) Cos2 (d) Trees (e) Lena

(f) Aerial (g) Field (h) Baboon (i) Car (j) Prlcar

(k) Train (l) Kiel (m) Wheel

Figure 3.12: Images and Video Sequences used in Simulation.

proposed multi-directional Wiener filter is still faster than the Wiener filter. The added complexity of using the Sigma probability in the multi-directional Sigma filter is justified by the significant increase in gain.

Table 3.1: Time Complexity Comparison.

Algorithm	(Time Ratio/512x512 frame)
Directional Filter [3]	0.80
Directional Sigma	1.00
Standard Sigma [4]	1.44
Recursive Sigma [2]	1.67
Directional Wiener	2.57
Standard Wiener [5]	3.02
Zed Filter [21]	6.50

3.6.2 Temporal Stability

The average gain (in PSNR dB) over time for the proposed multi-directional Sigma, the proposed multi-directional Wiener, and referenced methods is shown in Fig. 3.13 for the video sequences in Fig. 3.12(i-m) corrupted with 25[dB] noise. It can be seen that the proposed algorithms outperform the referenced methods and are also more stable over time.

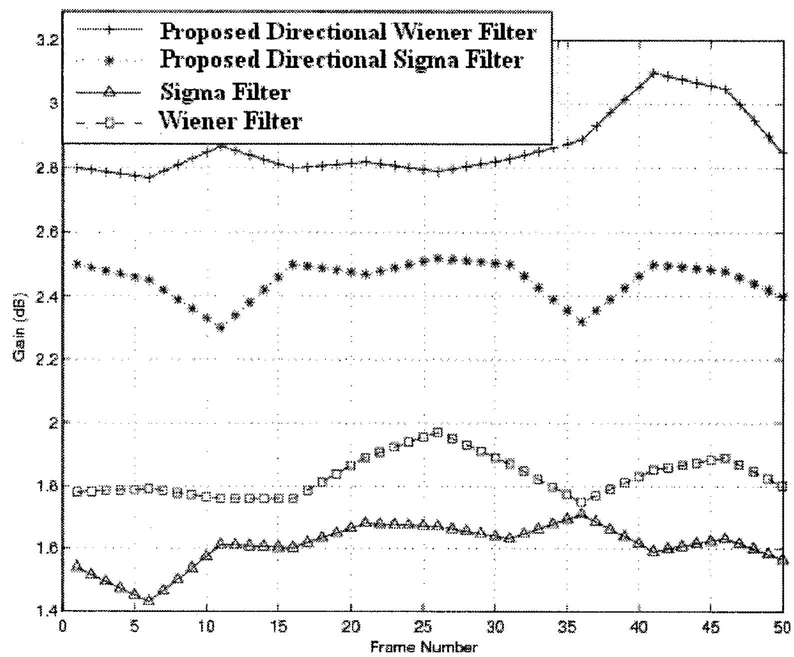


Figure 3.13: Gain (in [dB]) over time achieved from applying proposed and referenced methods to 25[dB] noisy video sequences in Fig. 3.12(j-l).

3.6.3 Results of the Proposed Directional Sigma Filter

A comparison of the gain achieved by the proposed multi-directional Sigma filter and referenced real-time methods at different noise levels is shown in Fig. 3.14 (for the test images) and Fig. 3.15 (for the test video sequences). The proposed multi-directional Sigma filter outperforms the Sigma filter [4], the directional filter [3] and the recursive sigma filter [2] in terms of performance. Recall that it is faster than all referenced methods except [3] (as shown in Table 3.1).

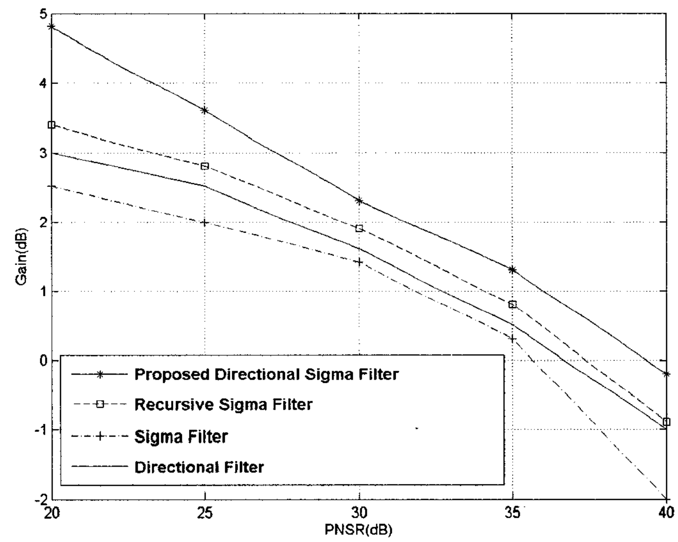


Figure 3.14: Applied to images in Fig. 3.12(a-h), the average gain achieved by proposed multi-directional Sigma, directional filter [3], Sigma filter [4] and recursive Sigma filter [2].

3.6.4 PSNR Gain of the Proposed Wiener Filter

A comparison of the gain achieved by the proposed multi-directional Wiener, standard Wiener [5] and the Zed Filter [21] at different noise levels is shown in Fig. 3.16 (for the test images) and Fig. 3.17 (for the test video sequences). The proposed multi-directional Wiener filter outperforms both the Wiener [5] and the Zed filters [21] while still being faster than both (see Table 3.1).

3.6.5 Visual Comparison of the Proposed Wiener Filter

Fig. 3.18 shows how the proposed multi-directional Wiener filter, increases the structure preservation capabilities of the Wiener filter while still yielding high gain. Note the blurring introduced by the Wiener filter in high structured areas such as the hair in the *Lena* image or the vertical stripes in the *Barbara* image or the facial hair of the *Baboon* image. Fig. 3.19 show the first frames of the *Train* sequence. With the

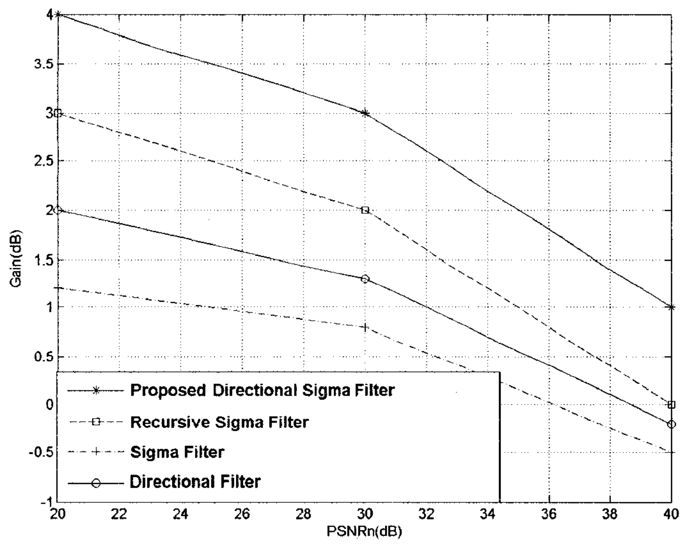


Figure 3.15: Applied to video sequences in Fig. 3.12(i-m), the average gain achieved by proposed multi-directional Sigma, directional filter [3], Sigma filter [4] and recursive Sigma filter [2].

Wiener filter, details at the south part of the *Train* image are suppressed.

3.7 Summary

This paper proposed a framework for adaptive multi-directional spatial filtering of AWGN in images and video frames. The adaptation to image content is achieved through combining most homogeneous directions to tailor a kernel that best fits image structure. The adaptation to noise level is through controlling the order and coefficients of the filter using the block size, number of directions used in filtering and pixel weighting. We proposed a low cost multi-directional Sigma filter suitable for real-time noise reduction. The proposed multi-directional Sigma filter achieves higher gains in PSNR than the classical Sigma filter. It can achieve up to 4.8 dB PSNR gain in real-time. We also proposed a multi-directional Wiener filter capable of achieving gains in PSNR of up to 5.6 dB. The proposed multi-directional Wiener filtering is well

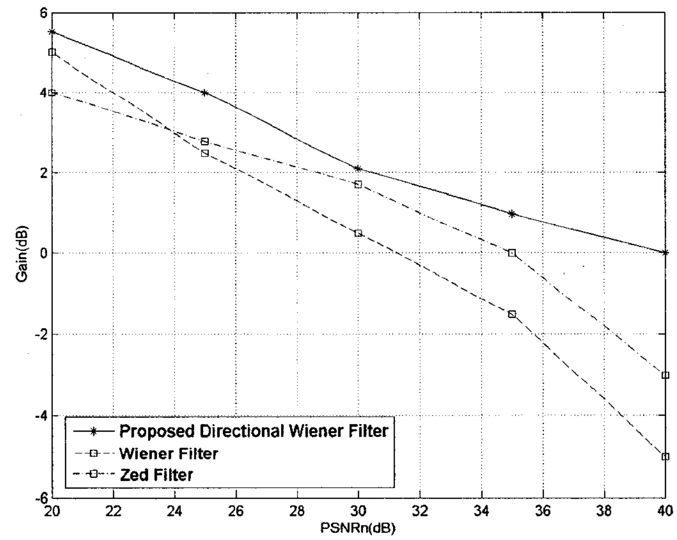


Figure 3.16: Applied to images in Fig. 3.12(a-h), the average gain achieved by proposed multi-directional Wiener filter and the Wiener filter [5]. Note the average gain of 3.31 [dB] of the proposed multi-directional Wiener compared to the 1.62 of the Wiener filter in the 25-30 [dB] range.

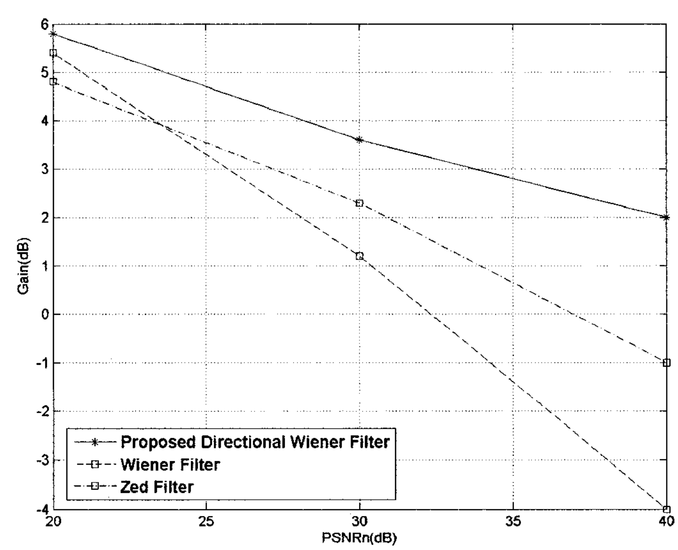


Figure 3.17: Applied to video sequences in Fig. 3.12(i-m), the average gain achieved by by proposed multi-directional Wiener and the Wiener filter [5]. The proposed multi-directional Wiener clearly outperforms the classical Wiener filter.

(a) 25dB Noisy Lena. (b)Wiener filter [5], 3.5 dB Gain. (c)Proposed Directional Wiener, 4.7 dB Gain.

(d) 25dB Noisy Barbara. (e)Wiener filter [5], 2.9 dB Gain. (f)Proposed Directional Wiener, 3.4 dB Gain.

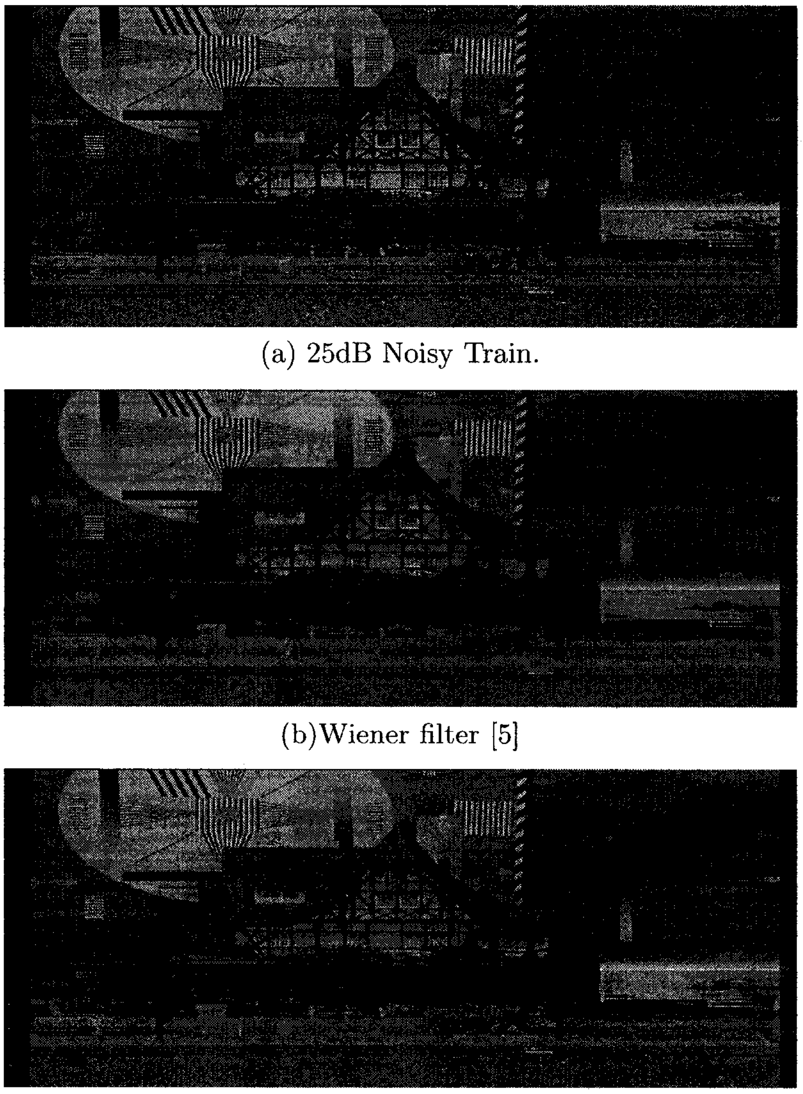
(d) 20dB Noisy Baboon. (e)Wiener filter [5], 2.7 dB Gain. (f)Proposed Directional Wiener, 3.1 dB Gain.

(g) Reduced residual blurring next to the Baboon's left eye. Classical Wiener to the left and proposed Wiener to the right.

(h) Reduced residual blurring around the Baboon's mustasch Classical Wiener to the left and proposed Wiener to the right.

Figure 3.18: Improved structure-preservation in the multi-directional Wiener filter over the classical Wiener filter for Lena, Barbara and Baboon images.

suited for offline image or video noise reduction. The Modulation Transfer Function (MTF) was used to measure the relative structure preservation capabilities of the



(a) 25dB Noisy Train.

(b)Wiener filter [5]

(c)Proposed Directional Wiener

Figure 3.19: Reduced residual blurring in first frame of the Train sequence. Note the high structure in the grass area south of the picture is lost due to blurring with the classical Wiener filter and is better preserved with the proposed Wiener filter.

proposed and referenced methods. The proposed Wiener filter reduces the residual blurring caused by the classical Wiener filter.

Chapter 4

Conclusion and Future Work

4.1 Summary

Video noise estimation and reduction methods are among the most integral parts of video processing systems. They are typically deployed at the preprocessing or post-processing stages. The video noise estimation stage relays information about the noise process to the video noise reduction stage which utilizes this information to optimize the filtering process. Video signals are corrupted with noise at different stages such as acquisition, recording, transmission and storage. The aggregate effect of noise can be modeled as Additive White Gaussian Noise (AWGN). With AWGN, the noise estimation process reduces to estimating the variance of the AWGN which completely characterizes the noise process. Since most of the information in a typical video signal is low frequency information and the noise is assumed to be white, most of the significant noise is located in the high frequency band. For this reason, low pass filtering is used to reduce noise. Low pass filtering can be performed in the spatial, temporal or spatio-temporal domains. Temporal and spatio-temporal filters are challenged by the presence of motion. In the absence of accurate motion information, spatial filtering is more desirable for its real-time performance.

This thesis proposed approaches for estimation and reduction of additive white Gaussian noise (AWGN) in video signals. First, it proposed a spatio-temporal algorithm for the estimation of video noise. The method divides the video signals into 3D portions of the signal or cubes. Cube homogeneity is measured using 2D and 3D Laplacian of Gaussian operators in the spatial, temporal and spatio-temporal domains. The variance across homogeneous plains (or domains) is calculated and recorded. The Least Median of Squares (LMS) robust estimator is used to produce the domain-wise estimate. Domain-wise estimates are averaged to get the frame-wise final noise variance estimate. The proposed algorithm works well for video sequences with high structure and motion activity. It performs reliably with different noise levels with a maximum estimation error of 1.7 dB.

Second, this thesis proposed a framework for spatial adaptive multi-directional filtering of image and video noise. Generalized multi-directional filtering was first developed and then adapted to the noise level. The benefit of such adaptation was examined. Adaptive multi-directional Sigma and Wiener filters are also proposed. The structure preservation capabilities of the proposed filters were studied using the Modulation Transfer Function. The proposed multi-directional Sigma filtering achieves gains in PSNR of up to 4.8 dB in real-time. The proposed multi-directional Wiener filter capable of achieving gains in PSNR of up to 5.6 dB and is well suited for offline applications.

4.2 Conclusion

Engineers are challenged on daily basis to produce faster, more robust and higher performance video processing algorithms to match the increased demand for visual information. The work in this thesis emphasized the importance of noise estimation and reduction in this process due to the performance deterioration of modern video processing algorithm that do not pay attention to the presence of noise.

It was observed that in video noise estimation and reduction, block-based approaches receive research interest partly because they lend themselves to practical hardware implementation. With only few lines delay, block-based methods can achieve comparable results to transform based techniques. Unfortunately, the research focus nowadays is mainly on transform-based methods. This thesis, in a way, shows that there is still potential for improved performance that is coupled with real-time deployability in block-based methods. As an example, there is still a need for robust block-based signal profiling algorithms such as block-based structure or homogeneity detection.

It was also concluded that directional approaches are useful in signal preservation during noise estimation and reduction. The main reason is that the assumptions upon which many algorithms are built are less violated inside specific regions within blocks. Directional approaches aim at finding these regions. In that sense, directional processing can help improve the performance of block-based algorithms. Therefore, this thesis recommends that more research is conducted to try to integrate directional approaches and block-based methods.

4.3 Future Work

This section describes the possible future work to this thesis. It starts by presenting the future work for the proposed noise estimation algorithm and moves to present the future work of the proposed noise reduction algorithm.

Spatio-Temporal Video Noise Estimation

The proposed approximation of the Laplacian of Gaussian operator performs well in detecting cube homogeneity but is still open for improvements. One way to improve it is to increase the window size. Because it is a 3D operator, increasing the window size means better sampling of the Laplacian of Gaussian impulse response but also means more frame-delays. The gain in accuracy of using a larger window size should be examined. It is also possible to design rectangular parallelopiped operators that can increase the window size (e.g., $5 \times 5 \times 3$ or $7 \times 7 \times 3$) without the need for frame delay operators. Another future task is to study the effect of changing the variance of the Laplacian of Gaussian operator on the overall performance of the system. Increasing the variance means more smoothing and less structure detection and vice versa. Adapting the variance of the operator to the variance of the noise should increase the accuracy of the proposed method. Fast convolution algorithms for the LoG operators have been proposed in the literature. Another future task would be to try to use one of the fast convolutions to increase the algorithmic speed of the proposed method. Also, there is room for improving the model used for Least Median of Squares calculation. Currently a simple linear model is used. Nonlinear models may produce better results.

Spatial Adaptive Multi-Directional Noise Reduction

A future task would be to propose other multi-directional spatial filters to try and find the average gain of performing spatial filtering multi-directionally. Such task should also improve the generalization of multi-directional filtering. Another future task would be to apply multi-directional filtering to the temporal domain and examine the overall performance gain. The homogeneity analyzers should then perform as low cost motion detectors. There is also room to improve the window size adaptation of the proposed Sigma and Wiener filters. Also, the optimal window size and number of directions can be found from a representative set of training data using Lloyd quantization algorithm.

Bibliography

[1] D.-H. Shin, R.-H. Park, Y. Seungjoon, and J.-H. Jung, "Block-based noise estimation using adaptive gaussian filtering," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 1, pp. 218–226, Feb 2005.

[2] G. de Haan, T. G. Kwaaitall-Spassova, M. M. Larragy, O. A. Ojo, and R. J. Schutten, "Television noise reduction IC," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 1, pp. 143–154, 1998.

[3] K. Jostschulte and A. Amer, "A new cascaded spatio-temporal noise reduction scheme for interlaced video," *IEEE International Conference on Image Processing*, vol. 2, pp. 293–294, 1998.

[4] J. S. Lee, "Digital image smoothing and the sigma filter," *Computer Vision, Graphics and Image Processing*, vol. 24, pp. 225–269, 1983.

[5] J. S. Lim, *Two-Dimensional Signal and Image Processing*, pp. 536–540, Prentice Hall, 1990.

[6] Television Viewing, "The daily," *Statistics Canada*, Nov.21, 2003.

[7] A. Bovik, *Handbook of Image and Video Processing*, vol. 2, pp. 275–285, ELSEVIER Academic Press, 2005.

[8] A. Amer, "Object and event extraction for video processing and representation in on-line video applications," *Ph.D. thesis, INRS-Télécommunications, Université du Québec,*, pp. 13–15, December 2001.

[9] A. Bovik, *Handbook of Image and Video Processing*, vol. 2, pp. 400–401, ELSEVIER Academic Press, 2005.

[10] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, vol. 2, pp. 222–223, Prentice Hal, 2002.

[11] A. Amer and E. Dubois, "Fast and reliable structure-oriented video noise estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, pp. 113–118, January 2005.

[12] B. C. Song and K. W. Chun, "Noise power estimation for effective de-noising in a video encoder," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 357–360, March 2005.

[13] S. I. Olsen, "Estimation of noise in images: an evaluation," *Graphical Models and Image Processing*, vol. 55, no. 4, pp. 319–323, July 1993.

[14] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, April 1994.

[15] G. L. Shevlyakov and N. O. Vilchevski, *Robustness in Data Analysis: Criteria and Methods*, pp. 6–10, VSP BV, 2002.

[16] Charles V. Stewart, "Robust parameter estimation in computer vision," *SIAM Review*, vol. 41, no. 3, pp. 513–537, 1999.

[17] G. de Haan, T. G. Kwaaitaal-Spassova, and O. A. Ojo, "Automatic 2-D and 3-D noise filtering for high quality television receivers," *Proceedings of the 7^{th} International Workshop on Signal Processing and HDTV*, October 1994.

[18] T. F. Chan, S. Osher, and J. Shen, "The digital TV filter and nonlinear denoising," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 231–241, Febuary 2001.

[19] M. Nagao and T. Matsuyama, "Edge preserving smoothing," *Computer Graphics Image Processing*, vol. 9, pp. 394–407, April 1979.

[20] Y. Huang and L. Hui, "Adaptive spatial filter for additive Gaussian and impulse noise reduction in video signals," *International Conference on information Communication and Signal Processing*, vol. 1, pp. 523–526, December 2003.

[21] F. Russo, "A method for estimation and filtering of Gaussian noise in images," *IEEE Transactions on Instrumentation and Measurements*, vol. 52, no. 4, pp. 1148–1154, August 2003.

[22] S.-H. Lee and J. K. Seo, "Noise removal with Gauss curvature-driven diffusion," *IEEE Transactions on Image Processing*, vol. 14, no. 7, pp. 904–909, July 2005.

[23] L. Zhang P. Bao, "Denoising by spatial correlation thresholding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 6, pp. 535–538, June 2003.

[24] J. Zhong and R. Ning, "Image denoising based on wavelets and multifractals for singularity detection," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1435–1447, October 2005.

[25] E. J. Balster, Y. F. Zheng, and R. L. Ewing, "Feature-based wavelet shrinkage algorithm for image denoising," *IEEE Transactions on Image Processing*, vol. 15, no. 3, pp. 789–789, March 2006.

[26] L. Zhang, P. Bao, and X. Wu, "Multiscale LMMSE-based image denoising with optimal wavelet selection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 4, pp. 469–481, April 2005.

[27] R. Castagno, S. Marsi, and G. Ramponi, "A simple algorithm for the reduction of blocking artifacts in images and its implementation," *IEEE Transactions on Consumer Electronics,*, vol. 44, no. 3, pp. 1062–1070, August 1998.

[28] G. Ramponi, "The rational filter for image smoothing," *IEEE Signal Processing Letters*, vol. 3, no. 3, pp. 63–65, March 1996.

[29] F. Cocchia, S. Carrato, and G. Ramponi, "Design and real-time implementation of a 3-D rational filter for edge preserving smoothing," *IEEE Transactions on Consumer Electronics*, vol. 43, no. 4, pp. 1291–1300, November 1997.

[30] Y. Wang, J. Ostermann, Y.-Q. Zhang, Y. q. Zhang, and J. Ostermann, *Video Processing and Communications*, vol. 1, pp. 241–248, Prentice Hall, 2002.

[31] P. S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions in Information Theory*, March 1957.

[32] S. Inoue and K. R. Spring, *Video Microscopy: The Fundamentals*, pp. 37–39, Springer, 1997.

[33] R. C. Gonzalez and R. E. Woods, *Digital Image Processing - Second Edition*, pp. 230–231, Prentice Hall, 2002.

[34] M. Luxen and W. Förstner, "Characterizing image quality: Blind estimation of the point spread function from a single image," *Proceedings of the Photogrammetric Computer Vision and Image Analysis Symposium*, p. A: 205, 2002.