Noise Reduction for Video Signals

by

Xiaoli Li

B.A.Sc, University of British Columbia, 1996

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

Master of Applied Science

 \mathbf{in}

THE FACULTY OF GRADUATE STUDIES

(Department of Electrical & Computer Engineering)

We accept this thesis as conforming to the required standard

The University of British Columbia

December 2002

© Xiaoli Li, 2002

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

& Computer

Department of <u>Electrical Engineering</u>

The University of British Columbia Vancouver, Canada

Date Dec. 19, 2002

Abstract

Noise reduction is an important aspect of video processing. This thesis proposes a motion adaptive temporal-spatial noise reduction system. The three-step motion detection algorithm is the key for the system. First the pixel-to-pixel difference between two frames are compared to a threshold. This step gives a preliminary motion detection result for each pixel in the target frame. The preliminary motion detection results then go through the impulse pattern recognition module, where some false detections are corrected. The last step of the motion detection method is the spatial motion detection. It corrects those motion pixels that were mis-detected as still pixels in previous steps. Therefore, the pixel accurate motion detection results between the target frame and a reference frame is obtained. These results are used to control the operations of the temporal filter and the spatial filter. The temporal average filter only operates on the pixels that is not in motion to avoid motion artifacts. The edge adaptive spatial filter does little spatial filtering in the still areas of the picture to maintain the spatial resolution, while does more aggressive filtering in motion areas of the picture because the human eye is less

ii

sensitive to the blurring of moving objects. Therefore, the maximum noise reduction effect is achieved.

The proposed algorithm is implemented both in a hardware prototype using FPGAs and as a real-time software module on a VLIW DSP chip. Both prototypes made the intensive subjective evaluations of the algorithm possible. It is proved that the algorithm improves the picture quality about 6 dB, in terms of signal-tonoise ratio and subjective estimation, for most of the noisy pictures. It introduces minimum artifacts.

Contents

bstract	i
ontents i	v
ist of Tables vii	i
ist of Figures i:	ĸ
cknowledgements x	i
edication xi	i
Introduction	1
1.1 Background	1
1.2 Video Signal Impairments	3
1.2.1 Analog Video Noise	3
1.2.2 Digital Video Noise	5
1.3 Possible Applications of Noise Reduction	6

1.4 Research Objectives and Challenges 1.5 Scope of the Thesis 2 Noise Reduction Filters 2 Noise Reduction Filters 2.1 Temporal Filtering 2.1.1 Direct Filtering 2.1.2 Motion Adaptive Filtering 2.1.3 Motion Compensated Filtering 2.1.4 Motion Compensated Filtering 2.1.5 Spatial Filtering 2.2 Spatial Filtering 2.2.1 Linear Spatial Filtering 2.2.2 Non-linear Filtering 2.2.3 Adaptive Filtering 2.3 Temporal-spatial Filtering 3 Motion Detection in Video Signals 3.1 Basic Algorithm 3.2 Improvements to the Basic Algorithm 3.2.2 Post-processing of the Basic Algorithm 3.3 Summary			
1.5 Scope of the Thesis 2 Noise Reduction Filters 2.1 Temporal Filtering 2.1.1 Direct Filtering 2.1.2 Motion Adaptive Filtering 2.1.3 Motion Compensated Filtering 2.1.3 Motion Compensated Filtering 2.1.4 Linear Spatial Filtering 2.2.5 Spatial Filtering 2.2.1 Linear Spatial Filtering 2.2.2 Non-linear Filtering 2.2.3 Adaptive Filtering 2.2.4 Adaptive Filtering 2.2.3 Adaptive Filtering 2.3 Temporal-spatial Filtering 2.3 Temporal-spatial Filtering 3 Motion Detection in Video Signals 3.1 Basic Algorithm 3.2 Improvements to the Basic Algorithm 3.2.1 Pre-processing of the Basic Algorithm 3.3 Summary 3.3 Summary	1.4 Re	esearch Objectives and Challenges	7
2 Noise Reduction Filters : 2.1 Temporal Filtering	1.5 Sc	cope of the Thesis	8
2.1 Temporal Filtering . 2.1.1 Direct Filtering . 2.1.2 Motion Adaptive Filtering . 2.1.3 Motion Compensated Filtering . 2.1.3 Motion Compensated Filtering . 2.1 Linear Spatial Filtering . 2.2 Spatial Filtering . 2.2.1 Linear Spatial Filtering . 2.2.2 Non-linear Filtering . 2.2.3 Adaptive Filtering . 2.3 Temporal-spatial Filtering . 2.3 Temporal-spatial Filtering . 3 Motion Detection in Video Signals 3.1 Basic Algorithm . 3.2 Improvements to the Basic Algorithm . 3.2.1 Pre-processing of the Basic Algorithm . 3.2.2 Post-processing of the Basic Algorithm . 3.3 Summary . 4 The Proposed Algorithm .	2 Noise	Reduction Filters	10
2.1.1 Direct Filtering . 2.1.2 Motion Adaptive Filtering . 2.1.3 Motion Compensated Filtering . 2.1 Spatial Filtering . 2.2 Spatial Filtering . 2.2.1 Linear Spatial Filtering . 2.2.2 Non-linear Filtering . 2.2.3 Adaptive Filtering . 2.2.3 Adaptive Filtering . 2.3 Temporal-spatial Filtering . 2.3 Temporal-spatial Filtering . 3 Motion Detection in Video Signals 3.1 Basic Algorithm . 3.2 Improvements to the Basic Algorithm . 3.2.2 Post-processing of the Basic Algorithm . 3.3 Summary . 3.4 The Proposed Algorithm .	2.1 Te	emporal Filtering	11
2.1.2 Motion Adaptive Filtering 2.1.3 Motion Compensated Filtering 2.1.3 Motion Compensated Filtering 2.2 Spatial Filtering 2.2.1 Linear Spatial Filtering 2.2.2 Non-linear Filtering 2.2.3 Adaptive Filtering 2.3 Temporal-spatial Filtering 2.3 Temporal-spatial Filtering 3 Motion Detection in Video Signals 3.1 Basic Algorithm 3.2 Improvements to the Basic Algorithm 3.2.1 Pre-processing of the Basic Algorithms 3.2.2 Post-processing of the Basic Algorithm 3.3 Summary 4 The Proposed Algorithm	2.1	1.1 Direct Filtering	11
2.1.3 Motion Compensated Filtering 2.2 Spatial Filtering 2.2.1 Linear Spatial Filtering 2.2.2 Non-linear Filtering 2.2.3 Adaptive Filtering 2.3 Temporal-spatial Filtering 2.3 Temporal-spatial Filtering 3 Motion Detection in Video Signals 3.1 Basic Algorithm 3.2 Improvements to the Basic Algorithm 3.2.1 Pre-processing of the Basic Algorithms 3.2.2 Post-processing of the Basic Algorithm 3.3 Summary 4 The Proposed Algorithm	2.1	1.2 Motion Adaptive Filtering	14
2.2 Spatial Filtering 2.2.1 Linear Spatial Filtering 2.2.2 Non-linear Filtering 2.2.3 Adaptive Filtering 2.3 Temporal-spatial Filtering 2.3 Temporal-spatial Filtering 3 Motion Detection in Video Signals 3.1 Basic Algorithm 3.2 Improvements to the Basic Algorithm 3.2.1 Pre-processing of the Basic Algorithms 3.2.2 Post-processing of the Basic Algorithm 3.3 Summary 4 The Proposed Algorithm	2.1	1.3 Motion Compensated Filtering	15
2.2.1 Linear Spatial Filtering 2.2.2 Non-linear Filtering 2.2.3 Adaptive Filtering 2.2.3 Temporal-spatial Filtering 2.3 Temporal-spatial Filtering 3 Motion Detection in Video Signals 3.1 Basic Algorithm 3.2 Improvements to the Basic Algorithm 3.2.1 Pre-processing of the Basic Algorithms 3.2.2 Post-processing of the Basic Algorithm 3.3 Summary 4 The Proposed Algorithm	2.2 Sp	patial Filtering	16
2.2.2 Non-linear Filtering 2.2.3 Adaptive Filtering 2.3 Temporal-spatial Filtering 2.3 Temporal-spatial Filtering 3 Motion Detection in Video Signals 3.1 Basic Algorithm 3.2 Improvements to the Basic Algorithm 3.2.1 Pre-processing of the Basic Algorithms 3.2.2 Post-processing of the Basic Algorithm 3.3 Summary 4 The Proposed Algorithm	2.2	2.1 Linear Spatial Filtering	17
2.2.3 Adaptive Filtering	2.2	2.2 Non-linear Filtering	18
 2.3 Temporal-spatial Filtering	2.2	2.3 Adaptive Filtering	19
 3 Motion Detection in Video Signals 3.1 Basic Algorithm	2.3 Te	emporal-spatial Filtering	20
 3.1 Basic Algorithm	3 Motio	on Detection in Video Signals	23
 3.2 Improvements to the Basic Algorithm	3.1 Ba	asic Algorithm	24
 3.2.1 Pre-processing of the Basic Algorithms	3.2 Im	provements to the Basic Algorithm	25
3.2.2 Post-processing of the Basic Algorithm	3.2	2.1 Pre-processing of the Basic Algorithms	26
3.3 Summary	3.2	2.2 Post-processing of the Basic Algorithm	28
4 The Proposed Algorithm	3.3 Su	ımmary	29
4.1 Functional Block	4 The F	Proposed Algorithm	31
	4.1 Fu	Inctional Block	32

	4.2	Motion Detection		
		4.2.1	Difference Thresholding	35
		4.2.2	Impulse Pattern Recognition	37
		4.2.3	Spatial Motion Detection	39
	4.3	Tempo	oral Filter	41
	4.4	Spatia	l Filter	42
	4.5	Opera	tions on Chrominance Signal	44
5	\mathbf{Th}	eoretic	al Analysis and Simulation Results	45
	5.1	Motio	n Detection Analysis	45
	5.2	Comp	uter Simulation Results	49
6	Pro	ototyp	e Implementation	52
6	Pro 6.1	o totyp Impler	e Implementation nentation on FPGA	52 53
6	Pro 6.1	ototyp Imple 6.1.1	e Implementation nentation on FPGA Prototype Description	52 53 53
6	Pro 6.1	ototyp Impler 6.1.1 6.1.2	e Implementation nentation on FPGA	52 53 53 55
6	Pro 6.1	Impler 6.1.1 6.1.2 Impler	e Implementation nentation on FPGA Prototype Description Design Consideration and Algorithm Optimization nentation On VLIW based DSP	52 53 53 55 55
6	Pro 6.1	Impler 6.1.1 6.1.2 Impler 6.2.1	e Implementation nentation on FPGA Prototype Description Design Consideration and Algorithm Optimization nentation On VLIW based DSP Implementation Platform	52 53 53 55 58 58
6	Pro 6.1	Impler 6.1.1 6.1.2 Impler 6.2.1 6.2.2	e Implementation nentation on FPGA Prototype Description Design Consideration and Algorithm Optimization nentation On VLIW based DSP Implementation Platform Design Considerations and Software Description	 52 53 53 55 58 59 60
6	Pro 6.1 6.2	Impler 6.1.1 6.1.2 Impler 6.2.1 6.2.2	e Implementation nentation on FPGA	 52 53 53 55 58 59 60 66
6	Pro 6.1 6.2 Ob 7.1	Impler 6.1.1 6.1.2 Impler 6.2.1 6.2.2 ojective Exper	e Implementation nentation on FPGA	 52 53 53 55 58 59 60 66 66

	7.1.2 A/D and D/A Convertors Performance Analysis	71
7.2	Measurement Result and Analysis	73
7.3	Subjective Noise Reduction Results	75
8 Co	onclusion and Future Work Suggestion	79
8.1	Future Work Suggestion	80

List of Tables

5.1	Motion Detection Post-processing Filters Performance	50
7.1	Noise Reduction by A/D Convertors	72
7.2	Noise Reduction Measurement Results	74

List of Figures

2.1	Recursive Temporal Filter Structure	13
4.1	Block Diagram of Proposed Algorithm	32
4.2	Patterns for impulse pattern recognition	39
5.1	A. Picture Difference Between Two Clean Frames; B. Picture Differ-	
	ence Between Two Gaussian Noise Corrupted Frames $\ldots \ldots \ldots$	46
5.2	A. Motion Detection Results After Difference Thresholding; A. Mo-	
	tion Detection Results After Median Filter; C. Motion Detection Re-	
	sults After Impulse Pattern Recognition; D. Motion Detection Results	
	After Spatial Motion Detection	51
6.1	Block Diagram for Hardware Prototype	54
7.1	Illustration of Experiment Setup	67
7.2	Waveform of One Line of Noise Corrupted NTSC Video Signal \ldots	68
7.3	Noise Measurment Without Filter	69

7.4	Noise Measurement with Lowpass Filetr at 4.2MHz	70
7.5	Noise Measurement with the 4.2MHz Lowpass Filter and $F_{(sc)}$ Trap	
	Filter	71
7.6	Noise Reduction Caused by A/D Convertors	73
7.7	Noise Reduction Caused by A/D Convertor	75

x

Acknowledgements

First and foremost, I would like to thank my thesis supervisor Dr. Rabab Ward. She is an ideal advisor: not only a great professor and researcher, but also a very enthusiastic and supportive mentor. She provides broad guidance, yet allows me virtually unrestricted freedom to explore new technologies.

I would like to thank my colleague and my friend Julong Du. Without his help and encouragement, this work would not have been completed. His knowledge and personality make working with him a fun and valuable experience.

I thank John Madden for his tireless effort to bring this work into a commercial product, and allowing me to involve in the process. This whole new experience has directly influenced my career.

I am grateful to my parents and my sisters for their love and support. Their belief in me carried me through the hard times. I dedicate this work to them.

Xiaoli Li

The University of British Columbia December 2002 To my parents and sisters

Zhongfu, Aiying, Xiaobin & Xiaofeng

Chapter 1

Introduction

1.1 Background

Noise is an inherent feature of any video communication system. Noise arises in each step in the video circulation path. At the source, it may be generated as photon noise or thermal noise in television cameras or as film grain in cine cameras. Recorders add yet more noise as the signal effectively re-circulates around a noisy loop during postproduction. Finally the signal is transmitted to the end users via the distribution network where more noise is added. In the case of digital transmission, although the signal is less vulnerable to transmission noise, the loss in video encoding and decoding introduces visible impairments. The sum total of all these may affect the picture quality negatively.[2][35]

The importance of noise reduction is constantly growing with the everincreasing use of cable television. To protect the right of the subscribers, the FCC regulated the baseline quality at the receiver end that any cable system has to meet. Delivering quality signal to subscribers is a difficult balancing act. It would have been a simple matter to deliver very high quality video if the cost was not a consideration. To upgrade an old cable system to meet the FCC requirements, the cable operators may have to replace old amplifiers that are present in the field. This is very costly. In the past years, cable operators have spent huge amount of money deploying cost-effective hi-tech devices to improve the quality of video signal. As time goes by and the technology improves, FCC has also lifted the bar on minimum quality constantly.

Having a noise reduction device, at a subscriber premise, is an alternative solution to the problem. Installing an inexpensive device at the subscriber's home is economical as network upgrade is much more expensive. The cost advantage of this solution becomes increasingly competitive as the cable industry is gradually migrating to digital. For quite a long period, the digital and analog services will coexist in the cable network. Embedded with such a function, a digital set-top box can decode digital programs, and remove video impairment while receiving analog programs. This can also narrow the gap of picture qualities between the digital and the analog signals.

This thesis documents the work on searching for and implementing an algorithm that removes commonly seen noise in cable television systems. This project is partially supported by Rogers Canadian Cable Foundation, an organization funded

2

by major cable operators in Canada. The goal of this project is to solve a real noise reduction problem in the cable networks.

1.2 Video Signal Impairments

Video signal in a cable systems experience many impairments. In the process of converting analog transmission to digital transmission, both analog and digital noises will arise in the network.

1.2.1 Analog Video Noise

The principal impairments for analog picture can be divided into two categories: coherent and non-coherent noise.[32]

Coherent interference includes ingress of alien signals, reflections of signals from the impedance discontinuities of transmission line, cross modulation of video signals and cross modulation of the carriers of video signal. They give rise to patterns on the screen called beats.[19] Two of the most commonly seen coherent interferences are Composite Triple Beats (CTB) and Composite Second Order Beats (CSO). The CTB impairment is due to the cumulative effect of hundreds of third order inter-modulation beat products result of the non-linearity in cable amplifiers. Graininess or a similar texture effect characterizes this impairment over the entire picture. CSO also results from the nonlinear behavior in electronic components. In optic fiber cables, it results from the laser drivers of optical fiber links, and sometimes the optical fiber itself. CSO patterns often look like moving diagonal bars or herringbones.[19][40]

Coherent impairments result in recognizable interference pattern superimposed on the picture. They are more objectionable than non-coherent impairments of equal strength. In cable systems, the noise level is expressed as a ratio of the visual carrier to the specific noise type in a television channel. This measure is called the Carrier-to-Noise Ratio (CNR) and is measured in decibels (dB). As of July 1, 1995, the FCC required the CNR of coherent disturbances to be 51 dB or higher.[40]

The principle non-coherent picture impairment is thermal noise. Thermal noise behavior is a well-understood part of general communication theory. It is a consequence of the statistical nature of the movements of electric charges in conductors. This noise is inescapable. If the intended analog signal is ever allowed to become weak enough in comparison with the noise signal, the signal will be polluted by the noise, yielding a snowy pattern in pictures. Hence, the common reference to thermal noise as snow noise in cable TV industry.

The regulatory target value of CNR for thermal noise is 45 to 46 dB. Subjective data were compiled in 1991 in a project sponsored by the Cable Television Laboratories, CATV. The work indicates "perceptible but not annoying" CNR at about 47 to 50 dB. The "slightly annoying" category was within about 1 dB of 41 dB.[40]

4

Impulse noise is another kind of the non-coherent impairments. It has short duration and high energy. It occurs most often in over-the-air transmission such as in standard broadcasting and satellite transmission.

1.2.2 Digital Video Noise

Digital signal is more robust against noise during transmission. However, the picture compression process at the encoder as well as the post-production stage and decompression process at the user end may degrade the picture quality. This degradation results in impairments including blockiness, mosquito noise, blurring, contouring, color bleeding and ringing.[19]

Most of the noise in digital images is caused by uneven resolution loss in the quantization process. Blockiness is one of the major disadvantages of block-based compression techniques, such as JPEG or MPEG.[16] Blockiness is represented by intensity discontinuities at the boundaries of adjacent blocks in the decoded image. It is caused by coarse quantization of DCT coefficients. When the quantization levels are not the same for adjacent blocks, a blockiness appearance will most likely arises. The mosquito effect is a temporal artifact and is visible mainly in smooth-textured regions as fluctuations of luminance/chrominance levels around high contrast edges or moving video objects. It is caused by having different quantization levels for the same smooth-textured area in consecutive pictures.

This section gives a brief introduction of well-known video noises. Each kind has its unique characteristics. It is almost impossible to remove all these kinds of noises by applying a single filter. This thesis is mainly focused on the improvement of thermal noise impaired pictures. Since thermal noise spread widely in all kinds of communication systems, the work of this thesis will have a broader application area than cable industry.

1.3 Possible Applications of Noise Reduction

Noise reduction can be widely implemented in consumer products such as in digital set-top boxes and in large screen television sets. As the technology progresses in every area of video processing, pictures of higher quality are provided to consumers. The consumers' viewing expectations become higher and the tolerance to poor picture quality becomes lower. This situation urges the manufactures to utilize all possible technologies to meet customers demand.

Noise reduction also finds application into digital video. Digital video compression techniques, which was optimized and popularized by the MPEG-2 international standard, have been providing turnkey solutions for transfer from analog TV to digital TV. While MPEG-2 is a very efficient video compression standard, it does not address video preprocessing and post-processing needed to further improve the quality of digital video image. It has been observed that the noise level in the input video has a big impact to the ultimate compression quality since the MPEG-2 compression scheme, especially the motion compensation process, is very sensitive to noise. Noise increases the entropy of the image sequence and therefore hinders effective compression. Thus, filtering of image sequences for noise suppression is often a desirable preprocessing step. In post-processing, noise reduction filters may be used to remove digital artifacts, such as blockiness and mosquito noise, so that lower coding rate can be achieved without compromising picture quality; this means bandwidth savings.[18]

Noise reduction is also a central issue in medical image processing. An image from a medical imaging system is acquired to obtain a 2D representation of a 3D object for medical diagnosis. Unfortunately, most images are degraded by various distortions. Sometimes, these distortions result in inadequate object representation and make accurate image analysis difficult. Effective corrections of other distortions are often closely related to and based on successful noise reduction. Furthermore, noise reduction makes pattern recognition and image analysis for industrial and scientific applications more accurate and efficient.[17]

1.4 Research Objectives and Challenges

The objective of this research work is to find a noise reduction method that reduces some of the commonly seen noise, especially the thermal noise, in video signals and to improve the objective quality, as well as the subjective quality of the picture. The deliverable of the project is a noise reduction prototype, which implements the proposed algorithm.

The evaluation of noise reduction involves two aspects, the effectiveness in

noise reduction as well as the minimization in the artifacts the algorithm introduces to the picture. Noise reduction inevitablely degrades the picture in one way or another. Many noise reduction schemes introduce varies artifacts to the processed picture. These artifacts become more evident when the pre-processed picture is relatively noise free. Therefore, the objective of the noise reduction system to be developed is to deliver superior performance in noise reduction as well as introduce minimal artifacts, especially for clean pictures.

The algorithm has to be simple and easy so that it could be implemented in both hardware and software. Some noise reduction algorithms give good noise reduction results, but are so complicated that they are of little commercial value. One objective of this project is to develop an algorithm so that the implementation cost is low enough for the price-sensitive consumer market. While meeting the cost constraint, the performance should not be compromised. Achieving an optimal combination of cost and the effectiveness is another challenge to this project.

1.5 Scope of the Thesis

The thesis is a record of nearly three years of research and development of the noise reduction project. Chapter two is a review of some noise reduction filters previous researchers have developed. Advantage and drawbacks of different filters are discussed, as well as their performances to different types of noise. Chapter three discusses motion detection algorithms, which constitutes the most important part of our developed system. Different motion detection algorithms are compared. Special attention is paid to the performance under noisy condition. Chapter four describes the proposed noise reduction algorithm in detail. Chapter five is the theoretical analysis of the proposed algorithm. The results are further validated by computer simulations. Chapter six documents the implementation of the proposed algorithm using FPGA hardware as well as DSP software. The design consideration and implementation optimization are also discussed. Chapter seven analyzes the measurement results obtained from the hardware prototype, and the subjective evaluation results on both the software and the hardware prototypes. The last chapter, chapter eight, concludes the thesis and recommends future work topics.

Chapter 2

Noise Reduction Filters

Noise reduction attempts to recover an underlying perfect image from a degraded copy. This problem is intractable unless one makes assumptions about the actual structure of the perfect image. Various noise reduction techniques make various assumptions, depending on the type of imagery and the goals of the restoration. In video processing, the assumption we make is that the noise signal is a random event whose variation is independent from that of the video signal. The noise signal is uncorrelated from pixel to pixel and from frame to frame. However, the image is highly correlated. The value of every pixel correlates with the values of its neighboring pixels, and with the value of the pixel in the consecutive frame with the same scan position. Therefore noise reduction of each pixel can be achieved through linear or nonlinear filtering on correlated pixels.

Based on this assumption, various noise reduction algorithms are proposed

in the literature.[36] Different techniques are used to search for the right correlated pixels under noise condition, and several filtering schemes are proposed to cancel the noise. This chapter discusses these filters in three categories, spatial filter, temporal filter and temporal-spatial filter.

2.1 Temporal Filtering

Noise superimposed to video signals is a random event and varies independently from video signals. It is uncorrelated from frame to frame. However, the image is highly correlated in temporal domain, especially for stationary areas. Based on the fact that the picture differences for stationary areas of two consecutive frames are only caused by noise, temporal filters can reduce the noise in the video signals without impairing the spatial resolution.[35] Thus, averaging the signals that corresponding to successive pictures leads to noise reduction since the frame-to-frame difference is random with zero mean.

Temporal noise filtering can be categorized into three types: Direct filtering, motion adaptive filtering and motion compensated filtering.

2.1.1 Direct Filtering

Direct filtering is the most straight forward implementation of temporal filtering. It filters pixels at the same raster position in two or more consecutive frames. The filter structure can be arranged as a transversal filter or a recursive filter.

Transversal Temporal Filtering

Transversal filter is a Finite Impulsive Response (FIR) filter that operates in the time domain. A simple form of transversal temporal filter is frame averaging, where pixels occupying the same raster position in consecutive frames are averaged. The frame average filter can be expressed as:

$$s(x,y,i) = \frac{1}{n} \sum_{j=0}^{n-1} p(x,y,i-j)$$
(2.1)

where s(x, y, i) is the estimated value for pixel p(x, y, i), i.e. the noise reduced pixel, and n is the number of history frames involved in the averaging.

Direct temporal averaging is well-suited to stationary parts of the image, because averaging multiple observations of essentially the same pixel in different frames eliminates noise while resulting in no loss of spatial image resolution. It is well known that direct averaging in this case corresponds to the maximum likelihood estimation under the assumption of white Gaussian noise, and reduces the variance of the noise by a factor of n.

In the implementation of a transversal filter, the history frames are obtained through a series of delay elements, each of length one picture period. These field or frame buffers are the main contributors to the filter cost. To achieve better noise reduction, more history frames are needed. This means higher cost as more delay elements are used. Recursive filtering is thus proposed to reduce the memory cost while delivering the same noise reduction results.

Recursive Temporal Filtering

A recursive temporal filter is a Infinite Impulse Response (IIR) filter operates in the time domain. It only requires one picture delay element. To avoid gain at zero frequency, the filter is arranged as in figure 2.1. Here the effect of the subtractor, divider and adder is to form the output as a fraction 1/K of the input picture and a fraction 1 - (1/K) of the previous output picture, as shown in equation 2.2.[36][16]

$$s(x, y, i) = \frac{1}{K}p(x, y, i) + (1 - \frac{1}{K})s(x, y, i - 1)$$
(2.2)

The amount of noise reduction is controlled by the feedback factor K. With this filter it is possible to obtain very high value noise reduction factor simply by increasing K. However, in digital implementation, each time K doubles, one extra bit is required to represent the value for K. The resolution for all the computation, such as multiplication, also have to increase by one bit. This brings more cost for implementation.[2]



Figure 2.1: Recursive Temporal Filter Structure

Direct filtering does a good job in the stationary parts of the video im-

ages, but introduces unacceptable motion artifacts to the moving part of the image. Therefore, it is seldom used in real implementations. Instead, motion adaptive and motion compensated temporal filtering are proposed to prevent motion artifacts.

2.1.2 Motion Adaptive Filtering

In motion adaptive filtering, the filter parameters are tuned according to a motion detection signal, such as the frame difference. The filter processes pixels in the stationary part of the picture, and usually turns off filtering when a large motion is detected, in an attempt to prevent artifacts.[25][10][34]

The motion adaptive approach can be applied to both transversal and recursive filter structures. For motion adaptive transversal filter, each pixel p(x, y, i-j) in each history frames F_{i-j} is weighted by a factor $a_j(x, y, j)$, which is a function of the motion detection results. The filter is defined as:

$$s(x,y,i) = \frac{1}{n} \sum_{j=0}^{n-1} a_j(x,y,j) p(x,y,i-j)$$
(2.3)

For the stationary parts of the image, $a_j(x, y, i)$ is equal to one. The value for $a_j(x, y, i)$ approaches zero as motion increases. For large motion, $a_j(x, y, i)$ is equal to zero, which means no filtering for the corresponding pixel.

For recursive filter, the motion detection result controls the value of K, as shown in equation 2.2. The K approaches to one as motion increases. When Kequals to 1, there is no filtering for the corresponding pixel.

It is obvious that the motion detection quality directly affects the perfor-

mance of the motion adaptive filter. If a false detection occurs, i.e. a stationary pixel is detected as a motion pixel, the overall noise reduction performance decreases as filtering will be turned off for motion pixels. If a motion pixel is detected as stationary pixel, (i.e. a miss detection occurs) then motion artifacts will appear due to filtering this pixel. Different motion detection methods are developed trying to increase the possibility of correct detection. These will be introduced in the following chapter.[1]

Motion adaptive filtering prevents the appearance of motion artifacts, and greatly improves the overall noise reduction performance. However, the noise reductions in different parts of the picture are uneven. Noise is cleaned at the stationary parts, and appears in the motion parts. Motion compensated filtering is proposed to give even noise reduction to all parts of the picture.

2.1.3 Motion Compensated Filtering

Motion compensated filtering is a rather complicated process.[7][27][12] It is based on the assumption that the variations in the pixel gray levels over any motion trajectory is mainly due to noise. Noise in both the stationary and moving areas of the image can be effectively reduced by low-pass filtering over the respective motion trajectory at each pixel.

Various filtering techniques, ranging from simple averaging to more sophisticated adaptive filtering, can be employed given the motion compensated filter support. In the ideal case, where the motion estimation is perfect, direct averaging of image intensities along a motion trajectory provides effective noise reduction. In practise, however, motion estimation is hardly ever perfect due to noise and sudden scene changes, as well as changing camera views. As a result, image intensities over an estimated motion trajectory may not necessarily correspond to the same image value, and the direct temporal average may result in artifacts.[36]

One of the drawbacks for motion compensated filtering is its computational implementation. Motion search is a very computation intensive process. Take the most commonly used motion search method, block matching method, as an example. The block matching method divides the target frame into several blocks. For each target block, the method tries to find a matching block in a search window within the history frame based on certain matching criteria. Normally the search window is very large to get accurate motion search result, but this also brings significant amount of computation.

2.2 Spatial Filtering

Interframe noise filtering is another alternative for image noise reduction. Noise added to an image generally has a larger spatial frequency spectrum than the normal image components because of the spatial uncorrelatedness of the noise. Hence spatial filtering can be effective for noise cleaning. We can classify intra-frame noise filter into three categories. (i) Linear Shift-Invariant (LSI) filters, such as the weighted averaging filters and Linear Minimum Mean Square Error (LMMSE) filters, also known as Wiener filters, (ii) nonlinear filters, such as median filters and other order statistic filters, and (iii) adaptive filters, such as directional smoothing filters and local space-varying LMMSE filters[36].

2.2.1 Linear Spatial Filtering

Linear Shift-Invariant (LSI) noise filters can be designed and analyzed using frequency domain concepts, and are easier to implement. The filters present a trade-off between noise reduction and sharpness of image details. This is because any LSI noise reduction filter is essentially a low-pass filter. It attenuates those frequencies which are higher than a certain cut-off frequency. As a result, the high frequency noise is eliminated, and unfortunately the high frequency content of the image is also attenuated, resulting in blurring of the image.

Among all linear filters, the LMMSE filter gives the minimum mean square error estimate of the ideal image. That is, it is the optimal linear filter in the minimum mean square error sense. Loosely speaking, it determines the best cutoff frequency for low-pass filtering based on the statistic of the ideal image and the noise.[13]

Wiener filter is another type of LSI filter. Noise can be reduced at the receiver side, using a Wiener filter estimator that estimates the transmitted video signal making use of the noise and image statistics. The advantage of this method is that it does not require changes in the transmitter. However, this technique is not easy to implement for most of real world channels as it requires knowledge of the noise and image statistics which are normally nonstationary in nature. Clearly measurement of such statistics at the receiver side would complicate the system implementation.

2.2.2 Non-linear Filtering

The linear processing techniques previously described perform reasonably well on images with continous noise, such as additive uniform or Gaussian distributed noise, However, they tend to provide too much image smoothing for images with impulse-like noise. Nonlinear techniques often provide a better tradeoff between noise smoothing and the retention of fine image detail.

Morphological filters are, perhaps, the most well known nonlinear filters for image processing. They are commonly used for multidimensional signal processing as they can rigorously quantify many aspects of the geometrical structure of a signal in a way that agrees with human intuition and perception.[26]

The fundamental idea behind the morphological image-cleaning algorithm is to segment the residual image into features and noise, where the residual image is the difference between an original image and a smoothed version.[31] The features from the residual image are added back to the smoothed image. Ideally, this results in an image whose edges and other 1-D features are as sharp as the original image yet has smooth regions between them. Morphological operations include erosions, dilations, openings, closings. Erosion shrinks the processing area, whereas dilation expends the area. Opening suppresses the sharp capes and cuts the narrow isthmuses, whereas closing fills in the thin gulfs and small holes of processed areas.[30]

Rank-order filtering is another popular type of non-linear filters.[4] Median filters is the most commonly used rank-order filter as it causes no edge sharpness loss in the smoothed image.[15][5][11] In one-dimensional form the median filter consists of a sliding window encompassing an odd number of pixels. The center pixel in the window is replaced by the median of the pixels in the window. The concept of the median filter can be extended easily to two dimensions by utilizing a two-dimensional window of some desired shape such as rectangle or discrete approximation to a circle. It is obvious that a two-dimensional LxL median filter will provide a greater degree of noise suppression than sequential processing with Lx1 median filters, but twodimensional processing also results in greater signal suppression. Median filter does a good job in removing impulse noise. However, in the case of uniform noise, median filtering provides little visual improvement.

2.2.3 Adaptive Filtering

The goal of a spatial noise reduction scheme is to eliminate the spatially uncorrected noise from the spatially correlated image content. The easiest way to get this is via a spatial lowpass filter. To avoid blurring edges and lines, the filter has to be content adaptive.[9]

The common approach in spatial noise filtering of image data is to define a rectangular window around the target pixel, often of size 3x3. Then the noise is reduced by filtering the target pixels and some selected neighboring pixels in the

window.

An alternative approach for edge-preserving filtering is the directional filtering approach, where the filtering is carried along the edges, but not across them. The directional filtering approach may in fact be superior to adaptive LMMSE filtering, since noise around the edges can effectively be eliminated by filtering along the edges, as opposed to turning the filter off in the neighborhood of edges.[28]

2.3 Temporal-spatial Filtering

It follows that in purely temporal filtering a large number of frames may be needed for effective noise reduction. This requires a large number of frame storage. It also follows that in pure spatial filtering, blurring is introduced if effective noise compression is achieved. Therefore, temporal-spatial filtering is proposed to provide a compromise between the number of frame storage needed and the amount of spatial blurring introduced.[21]

An image sequence can be considered as spatiotemporal data, since it is a time sequence of two-dimensional images. Thus, in image sequence filtering, we make use of not only the spatial correlation between pixels but also the temporal correlation between frames by employing three-dimensional spatiotemporal processing techniques.[20]

In spatiotemporal filtering, the local spatial statistics are replaced by their spatiotemporal counterparts. It is obvious that the non-stationary mean assumption readily extends to the spatiotemporal domain. Further, provided that the motion estimation is accurate, and the filter supports are spatially uniform, the white residual signal with non-stationary variance assumption in the spatiotemporal domain is indeed a reasonable one since the image values over supports constitutes a uniform spatiotemporal region in this case. In addition, local ergodicity and white noise assumptions are extended to the spatiotemporal domain.[36]

The motion compensated spatiotemporal LMMSE filter is an extension of spatial LMMSE noise filtering to the spatiotemporal domain. The derivation of a local LMMSE estimate for spatial filtering is based on two assumptions. 1) The image has a non-stationary mean, and the residuals (difference between the estimated value and the true value) form a white process with non-stationary variance. 2) The noise is a zero-mean white process that can be signal-dependent or signalindependent, and the image and the noise processes are both locally ergodic. The second assumption , which is commonly used, allows replacing ensemble statistics with their local counterparts that can be estimated from the observed image. Experiments demonstrated that the assumption of white residual with non-stationary variance is valid over uniform regions.

The adaptive weighted averaging (AWA) filter is based on the premise that spatiotemporal motion-compensated averaging is an effective means of suppressing noise while preserving image sharpness, provided that the spatiotemporal filter support is uniform.[39] The AWA filter assigns a weight to each image value within the motion-compensated spatiotemporal support, the value of which is a function of the difference between that image value and the noisy pixel that is being filtered. In cases when the spatiotemporal support is nonuniform due to highly detailed image structure and/or inaccurate motion estimation - and/or abrupt scene change from one frame to another - the AWA filter simply weights down the effect of those values that are decidedly different from the pixel being filtered, hence avoiding excessive blur or inefficient filtering. Therefore, compared to LMMSE filter, the AWA filter is better suited for efficient filtering of sequences containing segments with varying scene change. On the other hand, when the spatiotemporal support is sufficiently uniform, AWA approaches direct averaging.

Chapter 3

Motion Detection in Video Signals

Motion is subjectively a very important element of the video signal. All efficient signal processing algorithms have to account for motion while improving the picture quality. For the motion adaptive filter discussed in the previous chapter, motion detection results directly affect the filter performance, as incorrect motion detection may results in artifacts or poor noise reduction performance.

In a simple motion detection scheme, motion is detected by computing the difference, or a function of it, between the intensity of the pixel and that of the corresponding pixel in the previous frame. However, this scheme is very sensitive to the presence of noise, as noise between two successive frames is incorrectly interpreted as motion.[3] Noise causes two types of errors, which we refer to as false detection
and miss detection. False detection refers to the kind of error where a still pixel is detected as a motion pixel. Miss detection refers to the error when a pixel in motion is detected as a still pixel. For a motion adaptive filter, miss detection results in motion artifacts, while false detection degrades the noise reduction performance. Therefore, for noise reduction filters, any motion detection algorithm is required to maintain low false detection and miss detection rates under noisy condition.

Different approaches for motion detection, that utilize temporal and spatial correlations among motions, are discussed in this chapter. This discussion follows the introduction of the basic motion detection algorithm as it is the basis for all other improved methods.

3.1 Basic Algorithm

In a basic inter-frame motion detection algorithm, thresholding is used to segment a video frame into "motion" and "still" regions with respect to the previous frame. We define the frame difference $D_{i,i-1}(x, y)$ between frames i and i - 1 as

$$D_{i,i-1}(x,y) = p(x,y,i) - p(x,y,i-1)$$
(3.1)

that is, the pixel by pixel difference between the two frames. Assuming that the illumination remains more or less constant from frame to frame, the pixel locations where $D_{i,i-1}(x,y)$ differ from zero indicate "motion" regions, while where $D_{i,i-1}(x,y)$ equal to zero indicate stationary regions. However, the pixel difference at (x,y) hardly ever has a value of zero because of the presence of noise.[36]

In order to distinguish the nonzero differences that are due to noise from those that are due to motion or scene change, thresholding is used by computing

$$m_{i,i-1}(x,y) = \begin{cases} 1 & \text{if } |D_{i,i-1}(x,y)| > T \\ 0 & \text{otherwise} \end{cases}$$
(3.2)

where T is an appropriate threshold. The choice of the value of the threshold T is a compromise between immunity to noise and motion detection. The lower the T value, the more chance that a high intensity noise would cause a false motion detection. As the T value increases, motion resulting from small pixel variations value will be ignored, which causes miss detections. This is the case e.g. when slow motion happens at a smooth motion boundary.

3.2 Improvements to the Basic Algorithm

Because of the above mentioned shortcoming, a basic algorithm relying on pixel by pixel difference and thresholding, as discussed in the above section, is seldom used for motion detection in real systems. However, since such a method is the most straight forward one for motion detection, it becomes the starting point of more sophisticate algorithms. These algorithms yield better i.e. noise insensitive motion detection results by adding pre-processing and post-processing to the basic algorithm.

3.2.1 Pre-processing of the Basic Algorithms

The purpose of pre-processing of the basic algorithms is to reduce noise before computing the frame difference.[8][22] Therefore, the motion detection results are less affected by noise.

One of the approaches is to use a spatial filter as a pre-processor for motion detection. The noisy pictures are cleaned by linear or adaptive spatial filters before they are fed into the motion detector. A spatial filter is usually applied to a sliding window with the target pixel at the center of the window. There are two opposing factors governing the choice of the window size. The larger the window size, the more accurate is the measure of noise, as manifested by the lower variance of the output, and so the more readily can motion be detected correctly. On the other hand, the larger the window size, the more likely the detector is to miss smooth motion in a small area. Therefore, pre-processing using a spatial filter improves the performance of the basic algorithm under noisy condition, but unfortunately at a cost of decreased motion sensitivity. In addition, the spatial filter artifacts may also cause motion detection mistakes.

Michael et al. [29] described another pre-processing method. The pixel differences are first computed between the target frame and the history frame. The pixel differences in a sliding window centered at the target pixel are integrated to generate an average difference over the area. The average difference is then compared with a threshold. The objective here is to differentiate between noise and movement under the assumption that the larger the number of picture points integrated, the more likely the noise will average to zero whereas motion remains unchanged.

Roeder et al. [6] further group the pixel differences in the sliding window into overlapped subarrays, each of which includes the target pixel. Each of the subarrays is independently examined to determine if the magnitude of all of the pixel differences represented in a particular subarray exceed a predetermined threshold. If the condition is met in any one of the subarrays, the target pixel is considered as a pixel in motion. Compared with [29], this method improves the property of detail preservation, but it is less robust in heavy noise situations as less pixels are involved in the averaging.

Koivunen [23] described a motion detection algorithm for television signal. This method is intended for motion adaptive signal processing, such as scan rate conversion and noise reduction. The algorithm, which is insensitive to channel noise, is called the MSB (most significant bit) motion detection method. Firstly, the 8-bit video intensity is limited to one bit by discarding the 7 least significant bits. Therefore, the picture becomes a binary image. Secondly, the binary image is organized into 2x2 non-overlapping blocks. Thirdly, the four bits in each block form a 4-bit codeword with each bit corresponding to a pixel sample. Lastly, the two codewords corresponding to two consecutive frames are compared. If there is a difference between the codewords, the corresponding pixels are considered to be motion pixels.

27

Compared to the basic method, this algorithm has better performance under noisy condition. The hardware simplicity is another advantage of this algorithm. It requires slightly more than 1/8 the memory required by most of the algorithms. However, the noise in-sensitivity of this method is not evenly distributed, and it depends on the pixel value. For a pixel with value less than 127, this method is more tolerable to negative values of noise than to positive ones, and the tolerance declines as the pixel value approaches 127. For a pixel with value greater than 127, this method is more tolerable to positive values of noise than to negative ones, and the tolerance increases as the pixel value departs away from 127. This method also reduces motion detection resolution to a quarter of the picture resolution.

All pixel motion detection methods mentioned above do improve the methods' noise immunity for a motion area or on the boundary of a motion area, but what they sacrifice is the sensitivity of the small motion.

3.2.2 Post-processing of the Basic Algorithm

Post processing of motion detection aims to correct the false detection and missed detection caused by noise based on the knowledge of the characteristics of a noise model and a motion model, hence to improve the overall motion detection performance.[24]

After applying the basic motion detection algorithm, Gaussian noise appears as impulse noise on the two-dimensioal matrix of motion detection results. When applying post filtering to the matrix, it is desirable to remove any single or nonconnecting motion elements, since they are most likely caused by misinterpreted noise. Edges and corners of the matrix should be preserved.

One of the systems described by Michael et al. [29] is to use a majority logic gate to post-process the motion detection results obtained by the basic algorithm. The input for the gate is the detection results of a matrix centered at the target pixel. Each entry in this matrix corresponds to a decision as to whether or not the pixel is a motion pixel. The size of the matrix is usually taken as 3x3 or 5x5. If the majority of the detection results in the matrix indicate motion pixels, the target pixel is considered in motion. This method eliminates any motion detection results with a disconnected area. e.g. single motion pixel being surrounded by non motion pixels, and vice versa. However, this method also smoothes the edge of the motion objects.

Median filter is well known for its edge preserving property. When applied to a binary motion detection signal, the median filter cancels out any single inconsistent decisions, resulting in a smooth decision field with adequate resolution. Median filter is a suitable choice for motion detection post-processing filter.

3.3 Summary

The basic method employing frame intensity difference and thresholding is the basis for all motion detection methods. This method is extremely noise sensitive. Some pre-processing methods have been proposed to improve the noise immunity of the basic algorithm. However, motion sensitivity of the algorithm is compromised. Other methods try to post-process the basic method results based on the assumption that the motion affected pixels are spatially and temporally correlated and the noise corrupted pixels appear randomly. These post processing methods prove to be effective. Our motion detection method that will be proposed in chapter 4 falls also under this category.

Chapter 4

The Proposed Algorithm

This chapter describes in detail the proposed noise reduction algorithm. The algorithm is based on the techniques described in the two previous chapters. To efficiently reduce noise, a three dimensional temporal-spatial filter that is based on pixel averaging is developed. To avoid motion blur artifacts and spatial blur artifacts, a sophisticate motion detection for the temporal filter and an edge detection and motion adaptivity scheme for the spatial filter are developed. Compromises on implementation complexity are also taken into consideration at the same time.[33]

Throughout the discussion in this chapter, we will use the following expression conventions. The variables for a signal pixel are expressed in lower case indicates. A upper case variable indicates a two-dimensional matrix. The size of the matrix equals to the size of the processed picture. Each element in the matrix is the variable with the same letter, but in lower case, for one pixel in the picture. For example, P_i indicates one frame, and p(x, y, i) is a pixel in the picture whose position is at (x,y). The motion indicator for pixel p(x,y,i) in frame P_i with respect to its corresponding pixel in frame P_{i-1} is $m_1(x, y, i)$, and M1 denotes the motion indicators for all the pixels in frame P_i with respect to frame P_{i-1} .

4.1 Functional Block

The algorithm consists of three major functional blocks: a motion detector, a temporal filter and a spatial filter. Figure 4.1 illustrates the block structure of the proposed algorithm.



Figure 4.1: Block Diagram of Proposed Algorithm

The processing of the algorithm is frame based. Thus for interlaced video, such as interlaced television signal, the two interlaced fields are assembled to form one frame. The frame to be processed is called the target frame, and the pixel to be processed is called the target pixel. Each frame consists of a two-dimensional array of pixels. Each pixel has one parameter to represent its luminance value and two parameters to represent its chrominance values.

The temporal filter in the proposed algorithm is a transversal filter. The filter structure is chosen to be transversal not to be recursive because the recursive filter structure involves high resolution multiply and division which is costly to implement in hardware. Therefore, several history frames are required for temporal filter to get decent noise reduction results. The number of the frames involved in the filter is a compromise between the temporal noise reduction and the implementation cost for memory. The proposed algorithm decided to use four frames in the processing, which theoretically brings maximum 6 dB noise reduction.

The digitized video signal first passes through three frame buffers to obtain the four frames. The target frame is the most recent frame. Let us denote the target frame as P_i , and the three history frames are denoted as P_{i-1} , P_{i-2} and P_{i-3} respectively, with P_{i-1} being the nearest frame to P_i . The pixels in the target frame is denoted as p(x, y, i), and pixels in the three history frames are denoted as p(x, y, i - 1), p(x, y, i - 2) and p(x, y, i - 3) respectively, where x and y represents the pixel's horizontal and vertical position at the frame.

The motion detector computes the motion indicators for P_i with respect to P_{i-1} , P_{i-2} and P_{i-3} respectively. Therefore, for each pixel in the target frame p(x, y, i), there are three motion indicators $m_1(x, y, i)$, $m_2(x, y, i)$ and $m_3(x, y, i)$ that indicate the motion statuses of pixel p(x, y, i) with respect to pixel p(x, y, i-1), p(x, y, i-2) and p(x, y, i-3). The temporal filter averages the input four frames depending on the value of three motion indicator signals as will be discussed in later section 4.3. The resultant filtered frame P'_i is then processed by the spatial filter to create noise reduced frame P''_i . It is noted that the spatial filter is controlled by motion index value $MI1_i$ as will be discussed in section 4.4.

4.2 Motion Detection

For our proposed motion adaptive noise reduction algorithm, the performance of motion detector plays an important role. If the motion detector creates too many false detections, the noise reduction will be not as effective because the false detected noisy pixels will not be filtered by temporal filter. On the other hand, if the motion detector creates too many miss detections, the picture will be degraded after noise reduction because the miss detected motion pixels will be filtered by the temporal filter, causing motion artifacts. The proposed motion detection algorithm is composed of three steps in series: difference thresholding, impulse pattern recognition and spatial motion detection. The first step preliminarily detects whether or not a pixel is a motion pixel. The impulse pattern recognition corrects the falsely detected still pixels, and the spatial motion detection corrects the miss detected motion pixels.

4.2.1 Difference Thresholding

Difference thresholding is a preliminary method to detect if the difference between corresponding pixels in two frames is caused by motion or noise. It is based on the assumption that the difference caused motion has much larger value than the difference caused by noise.

Each pixel in the target frame is compared with pixels in the previous frames at the same raster position. The pixel difference $d_k(x, y, i)$ between two pixels p(x, y, i) and p(x, y, i - k) is defined as:

$$d_k(x, y, i) = p(x, y, i) - p(x, y, i - k) \qquad k = 1, 2, 3 \tag{4.1}$$

The pixel difference $d_k(x, y, i)$ is then compared with a threshold T_1 . If $d_k(x, y, i)$ is greater than the threshold T_1 , then the pixel p(x, y, i) is considered as in positive motion with respect to p(x, y, i - k). If $d_k(x, y, i)$ is less than the threshold $-T_1$, then the pixel p(x, y, i) is considered as in negative motion with respect to p(x, y, i - k). Otherwise, the pixel p(x, y, i) is a still pixel with respect to p(x, y, i - k). Thus the preliminary motion estimator $md_k(x, y, i)$ is defined as

$$md_{k}(x,y,i) = \begin{cases} 1 & \text{if } d_{k}(x,y,i) > T_{1} \\ 0 & \text{if } -T_{1} \le d_{k}(x,y,i) \le T_{1} \\ -1 & \text{if } d_{k}(x,y,i) < -T_{1} \end{cases}$$
(4.2)

where 1 indicates the pixel is in positive motion, -1 indicates the pixel is in negative motion and 0 indicates the pixel is still. The preliminary motion estimator $md_k(x, y, i)$ indicates whether or not the corresponding pixel is affected by motion and the polarity of the motion.

The selection of threshold T_1 is important. The higher the threshold is set, the less chances that a noisy pixel is mistaken as motion pixel, i.e. better noise immunity. However, the higher threshold also means that more small motion will be ignored, which results in miss detection. If the threshold is set too low, then the noisy pixels have higher possibility to be detected as motion pixels, depending upon the noise level.

Therefore, we set the threshold T_1 to be adaptive to noise level. Thus a fixed percentage of noisy pixels will be detected as motion. Consider a picture with additive Gaussian noise of $N(0, \sigma)$, the threshold is set as a function of the standard deviation of the Gaussian noise, i.e.

$$T_1 = k\sigma \tag{4.3}$$

In this design, k is set as 2. When k is 2, approximately 95% of pixels in a still image area have a temporal difference less than the threshold. A false motion detection occurs with a probability of about 5%. Statistically speaking, only one pixel in every 20 pixels has the chance to exceed the threshold.

Some may think that the noise level is hard to measure in real time. In fact, for television signal, the noise level can be obtained by a simple filter over the quite lines of the channels.

4.2.2 Impulse Pattern Recognition

As discussed in the previous section, the preliminary motion estimators obtained by difference thresholding contain false detections and miss detections. The impulse pattern recognition scheme aims to distinguish those pixels that are corrupted by high energy noise but are detected as motion pixels from the real motion pixels.

After the thresholding operation, only a small percentage iof noisy pixels are mis-detected as motion pixels. These pixels usually appear randomly across the picture because of the spatial uncorrelation of the noise. Their value usually are markedly different from their neighbors. On the contrary, motion usually affects all the pixels in an area. Therefore, motion pixels are correlated to their surrounding pixels. This is the assumption that the impulse pattern recognition scheme is based upon.

The impulse pattern recognition operates on a 5x3 sliding window over a preliminary motion estimators matrix MDk_i , k = 1, 2, 3. The center of the operation window corresponds to the target pixel. The operation looks at the target preliminary motion estimator $md_k(x, y, i)$ and its neighboring motion vectors. If the target motion estimator indicates the pixel is still, the operation keeps the motion estimator unchanged. If the motion estimator indicates that the target pixel is in motion, the operation will look at the correlation of the target pixel's motion estimator with its neighboring motion estimators in the operation window. If the target motion estimator is correlated with any of the motion estimator in the window, the target pixel is consider in motion. Otherwise, the target pixel is considered still. Thus a false detection is found. For two preliminary motion estimators to be correlated, they have to be the same values, as defined in equation 4.2.

The operation of impulse pattern recognition can be expressed as:

$$mp_k(x, y, i) = \begin{cases} 1 & \text{if } md_k(x, y, i) \neq 0 \text{ and } Corr(md_k(x, y, i)) = 1 \\ 0 & \text{if either or both } md_k(x, y, i) = 0 \text{ or } Corr(md_k(x, y, i)) = 0 \end{cases}$$

$$(4.4)$$

where 1 indicates that the target pixel is a motion pixel, while 0 indicates that the target pixel is still. The output of impulse pattern recognition operation is the intermediate motion estimator $mp_k(x, y, i)$. The function $Corr(md_k(x, y, i))$ is a pattern recognition operation on the 5x3 window. If the preliminary motion estimators in the window matche any of the six patterns shown in Figure 4.2, the function Corr() output is equal to 1; otherwise it equals to 0. The value 1 indicates that the target motion estimator correlates to at lease one motion estimator in the window, while 0 indicates that the target motion estimator is not correlated to any of its neighbors. The target pixel, in this case, is likely corrupted by a high energy noise.

The patterns are designed to list all the possible cases of an isolated motion estimators. It is noted that in cases B, C, E, F of Figure 4.2, two consecutive motion estimators with same value are also considered as isolated motion estimators. It is because that in analog system, high energy noise tends to have relatively longer



Figure 4.2: Patterns for impulse pattern recognition

duration. It often affect two consecutive pixels after the video signal is digitized.

4.2.3 Spatial Motion Detection

The spatial motion detection scheme detects the motion pixels that are impaired by noise so that are mistaken as still pixels.

The basic idea for spatial motion detection is that motion will not affect an isolated pixel. instead, it affects all the pixels in an area. If majority of a pixel's surrounding pixels are in motion, the pixel has a large opportunity to be also in motion.

The spatial motion detection calculates a motion index value on an mxnwindow over the intermediate motion estimators matrix $MPk_i, k = 1, 2, 3$ obtained by impulse pattern recognition. The intermediate motion estimator for the target pixel is at the center of the operation window. In our algorithm, the window size is chosen to be 3x3. Each motion estimator in the window is multiplied by a weighting factor. These weighted elements are then added together to form the motion index for the target pixel, as show in Equation 4.5.

$$mi_k(x,y,i) = \sum_{p=-\frac{m}{2}}^{\frac{m}{2}} \sum_{q=-\frac{n}{2}}^{\frac{n}{2}} w(p,q,i) mp_k(x-p,y-q,i) \ k = 1,2,3$$
(4.5)

The weighting factors w(p,q,i) can be properly adjusted for better performance according to the specific application. In our algorithm, the weighting factor for a pixel are inversely proportional to the distance between this pixel and the target pixel.

The motion index is then compared with a threshold T_2 . If the motion index is higher than the threshold, the target pixel is considered as in motion. The spatial motion detection operation can be expressed as Equation 4.6.

$$m_k(x, y, i) = \begin{cases} 1 & \text{if either or both } m_{i_k}(x, y, i) \ge T_2 \text{ or } m_{p_k}(x, y, i) = 1 \\ 0 & \text{otherwise} \end{cases} \quad k = 1, 2, 3$$

$$(4.6)$$

where $m_k(x, y, i)$ is the final motion indicator obtained from the motion detection algorithm. The threshold T_2 can be adjusted for specific application.

After motion detection, each pixel p(x, y, i) in frame P_i has three motion indicators $m_1(x, y, i)$, $m_2(x, y, i)$ and $m_3(x, y, i)$, where $m_1(x, y, i)$ indicates if the target pixel is in motion with respect to its corresponding pixel in P_{i-1} , and $m_2(x, y, i)$ indicates if the target pixel is in motion with respect to its corresponding pixel in P_{i-2} , and so on. These motion indicators are used to control the temporal filter. The motion index $mi_1(x, y, i)$ is passed to spatial filter to parameters tuning.

4.3 Temporal Filter

The temporal filter in the proposed algorithm is a motion adaptive transversal average filter. There are four frame involved in the temporal average. Theory proves that average filter is the most effective for Gaussian noise. Therefore, the average filter is chosen for temporal noise reduction. The target frame is averaged with some or all of the three history frames pixel by pixel. Whether or not each pixel in the history frames will be counted in the average is decided by the motion indicator of the corresponding pixel.

The temporal filtering is expressed as,

$$p'(x,y,i) = \begin{cases} Avg4(x,y,i) & \text{if } m_1(x,y,i) = m_2(x,y,i) = m_3(x,y,i) = 0\\ Avg3(x,y,i) & \text{if } m_1(x,y,i) = m_2(x,y,i) = 0, m_3(x,y,i) = 1\\ Avg2(x,y,i) & \text{if } m_1(x,y,i) = 0, m_2(x,y,i) = 1\\ p(x,y,i) & \text{if } m_1(x,y,i) = 1 \end{cases}$$
(4.7)

where p(x, y, i) and p'(x, y, i) are the present and the estimated values of the target pixel respectively, and $m_k(x, y, i)$ is the motion indicator of the target pixel p(x, y, i)with respect to the pixel p(x, y, i-k) in history frame P_{i-1} . The averaging functions are defined as

$$Avg4(x, y, i) = \frac{1}{4}[p(x, y, i) + p(x, y, i - 1) + p(x, y, i - 2) + p(x, y, i - 3)]$$

$$Avg3(x, y, i) = \frac{1}{3}[p(x, y, i) + p(x, y, i - 1) + p(x, y, i - 2)]$$

$$Avg2(x, y, i) = \frac{1}{2}[p(x, y, i) + p(x, y, i - 1)]$$
(4.8)

After processed in the temporal filter, the resultant video signals are sent into the spatial filter.

4.4 Spatial Filter

The spatial filter is an edge adaptive filter. It operates on a mxn sliding window over the picture. In this design, we choose the window size as 3x3. Each pixel in the window is first compared with the target pixel. If the absolute value of the difference between two pixels is less than a threshold T_3 , the pixel is marked as support pixel. The filter computes the average of all the support pixels, and uses this value to replace the target pixel value. Therefore, the spatial filter preserves fine detail and edge while reducing the noise effectively. The spatial filter is defined as:

$$p''(x,y,i) = \frac{\sum_{p=-\frac{m}{2}}^{\frac{m}{2}} \sum_{q=-\frac{n}{2}}^{\frac{n}{2}} c(p,q,i) p'(x-p,y-q,i)}{\sum_{p=-\frac{m}{2}}^{\frac{m}{2}} \sum_{q=-\frac{n}{2}}^{\frac{n}{2}} c(p,q,i)}$$
(4.9)

where p''(x, y, i) is the noise estimated value for the target pixel. The decision factor c(p, q, i) for each pixel p'(x - p, y - q, i) in the window is defined as:

$$c(p,q,i) = \begin{cases} 1 & \text{if } |p'(x,y,i) - p'(x-p,y-q,i)| \le T_3 \\ 0 & \text{otherwise} \end{cases}$$
(4.10)

The choice for the level of T_3 is a trade off between preserving the fine details and reducing the noise. Aggressive spatial filtering (high value for T_3) may cause smoothing effects, which appears as blurring of the picture. To avoid degrading the picture quality, a common practice is to sacrifice noise reduction by setting the threshold to a fairly low level.

Based on the facts that the human eye is less sensitive to blurring when the picture is in motion, the proposed algorithm varies the T_3 level as a function of the motion index value obtained in equation 4.5. For picture areas which are still or have low motion, the motion index value is expected to be low, and vice versa. When the motion index value is low, the threshold is set to a low level. Therefore, the spatial filter does little noise reduction on still parts of the picture, and tries to preserve all the fine details. Although a small threshold results in less noise reduction, the overall noise reduction on these still parts are maintained because the temporal filter achieves its maximum noise reduction in the still parts of the picture. As motion index increases, more aggressive spatial noise reduction is done on the motion parts.

Experiments also found that motion adaptive temporal filter creates a clear boundary between unprocessed and processed part of the picture at the edge of the moving objects, which tends to be very annoying to viewers. By applying a spatial filter whose processing intensity gradually increases from still parts to moving parts of the picture, the boundary created by temporal filter will be smoothed out. This is also called "soft boundary" in some noise reduction techniques.

4.5 Operations on Chrominance Signal

Chrominance signal normally has less resolution than luminance signal as human eyes are less sensitive to color resolution. For example, the resolution for chrominance signal is half that of the luminance signal for digital TV.

A simplified version of the noise filter for luminance is applied to the chrominance signal. The chrominance filter shares the same motion indicator with the luminance signal for the same pixel. Instead of using a variable threshold in the spatial filter, the spatial filter for chrominance signal uses a fixed threshold which is set to a very low level because aggressive noise reduction may cause artifacts such as color bleeding.

Chapter 5

Theoretical Analysis and Simulation Results

As a motion adaptive temporal-spatial noise filter, the most innovative part of our proposed algorithm is the motion detection algorithm. In this chapter, the theoretical reasoning behind all the motion detection steps are discussed. The computer simulation results are presented as verification of the analysis.

5.1 Motion Detection Analysis

The motion detection method in our proposed algorithm aims to accurately detecting motion status of each pixel in a noisy environment. We use two scales, the miss detection rate and the false detection rate, to evaluate a motion detection algorithm.



Figure 5.1: A. Picture Difference Between Two Clean Frames; B. Picture Difference Between Two Gaussian Noise Corrupted Frames

For clean pictures, frame difference are the motion detection results between the two frames. Figure 5.1A shows the picture difference between two consecutive clean frames. The pixels in the white areas are pixels whose pixel difference are greater than zero. These pixels are considered in motion. However, this motion detection scheme no longer works in noisy environment. Figure 5.1B shows the picture difference of the same frames but corrupted by Gaussian noise. It shows that both motion and noise cause the difference between two frames. The motion information is buried in the Gaussian noise. Therefore, motion detection for noisy picture is essentially a feature extraction and picture restoration problem.

Let us define the difference between two noise corrupted frames P_i and P_{i-1} as D_i , where D_i is a two-dimensional matrix, as shown Figure 5.1B. In the basic motion detection method, each element of D_i is then compared with a threshold T1. If the absolute value of the pixel difference is greater than the threshold, the pixel is considered in motion, otherwise, the pixel is considered still, as defined in equation 4.2. This process converts D_i into a binary matrix MD_i , as shown in 5.2A. The Gaussian noise superposed on D_i becomes impulse like noise on MD_i . The thresholding causes two kinds of detection errors. Miss detection occurs when a pixel whose pixel difference caused by motion is lower than the threshold is considered as still pixel. False detection occurs when a high energy noise corrupted pixel whose picture difference is higher than the threshold is detected as in motion.

The goal of the post-processing is to correct the errors made by the thresholding method. It is equivalent to removing the impulse like noise from the intermediate motion estimator MD_i .

The proposed motion detection post-processing methods, the impulse pattern recognition and the spatial motion detection, are methods using adaptive nonlinear scheme. The spatial noise detection scheme is applied on the impulse pattern recognition results. Their design is based on the knowledge of the noise model and the characteristics of motion.

The impulse pattern recognition method aims to decrease the false detection rate. After thresholding, the noisy pixels whose pixel differences exceed the threshold are the source causing the false detection. The impulse pattern recognition scheme removes these false detection pixels based on the difference of the spatial distribution of the noise and motion.

If the threshold is set as twice the standard deviation of the noise, which is assumed as Gaussian distributed, 95% of the noise will be below the threshold, and only 5% of the noise corrupted pixel will be over the threshold and are misdetected as motion. Since the noise is randomly distributed across the image, after thresholding the picture differences, the noise becomes impulse like because it is uncorrelated in the spatial domain. Statistically, one is every 20 still pixels has the chance to be misdetected after thresholding.

In the proposed algorithm, the impulse pattern recognition is applied to a 5x3 sliding window over the pixel difference matrix. It is safe to assume that there is only one pixel in the window that has the chance be affected by noise. Therefore, if more than two motion pixels are detected in the window, we consider that the target pixel is affected by motion.

One exception stands when the noise has very high energy. It is found that in analog signal, if the noise peak is very high, it also affects the pixel adjacent to it. Therefore, if two consecutive pixels are detected as motion pixels, and their surrounding pixels are not in motion, we decide that both pixels are affected by noise. This exception is reflected in the pattern design shown in figure 4.2.

After the impulse pattern recognition is applied on MD_i , the intermediate motion estimator matrix MD_i is transformed into another binary matrix MP_i , as shown in figure 5.2C. Compared to MD_i , many isolated motion pixels are removed. The spatial motion detection applied on MP_i aims to correcting those miss detections. It is based on the observation that the probability of a pixel being in motion is related to the motion status of its surrounding pixels. The more of its surrounding pixels are in motion, the more likely that the pixel is a motion pixel. In proposed algorithm, the spatial motion detection operates on a 3x3 sliding window on matrix MP_i . It is essentially a majority detector of the pixels in the window. The results of the spatial motion detection Mi is the outcome of the motion detection, as shown in figure 5.2D.

5.2 Computer Simulation Results

The computer simulations are conducted on sample pictures to evaluate the performance of the motion detection method.

Here, we compare our motion detection post-processor with a median filter. Median filter is the most popular example of nonlinear filters based on order statistics. Median filtering is believed to be ideal for impulse noise reduction. It is able to preserve picturer edge and details, while suppressing isolated noise.[?] It thus forms a good choice of post-processing filter for motion detection. For an array X of nelements, a median filter can be defined as

$$med(X_i) = \begin{cases} X_{v+1} & \text{if } n = 2v+1 \\ \frac{1}{2}(X_v + X_{v+1}) & \text{if } n = 2v \end{cases}$$
(5.1)

where $X_1, X_2, ... X_n$ are random variables arranged in ascending order of magnitude

$$X_1 \le X_2 \le \dots \le X_n \tag{5.2}$$

The following table shows the miss detection and false detection rates for different motion detection post-processing filters.

Processing		Miss Detection Rate	False detection Rate
Proposed Method	Impulse Pattern Recognition	0.0372	0.0048
	Spatial Motion Detection	0.0359	0.0052
Median Filter		0.0550	0.0159

Table 5.1: Motion Detection Post-processing Filters Performance

It shows that the proposed motion detection post-processing filters perform as designed. They deliver better performance than median filter in terms of false detection rate and miss detection rate. The impulse pattern recognition greatly reduced the false detection rate compared to median filter. Although the false detection rate slightly increased after the motion detection results from the impulse pattern recognition are further processed by the spatial motion detection, the miss detection rate dropped. Experiments found that the spatial motion detection recovered those isolated motion objects, such as the highlight of moving ball, from being detected as high energy noise corrupted still objects. This greatly improves the subjective performances because the motion artifacts of highlight objects are extremely annoying.



Figure 5.2: A. Motion Detection Results After Difference Thresholding; A. Motion Detection Results After Median Filter; C. Motion Detection Results After Impulse Pattern Recognition; D. Motion Detection Results After Spatial Motion Detection

Chapter 6

Prototype Implementation

Evaluation of any noise reduction algorithm is not complete if it is not done in a real-time environment. Computer simulation may give accurate picture quality improvement results in terms of SNR improvement, but it is hard to perform subjective evaluation based on computer simulation results. Besides, noise reduction filter may behave differently in different scenes. Computer simulations can only cover a small portion of the the real world situation. Therefore, prototypes that run in real time were build to implement the proposed algorithm so as to conduct objective and subjective evaluation in real time.

As the most important application for the proposed algorithm lies in consumer products, the cost becomes a very crucial issue. The biggest challenge in prototype implementation is thus to simplify the design so as to reduce the cost. Different implementation platforms have their strengths and drawbacks, thus the proposed algorithm is modified to give the best performance in each platform. The design considerations are discussed for implementations on both FPGAs and DSPs.

6.1 Implementation on FPGA

For consumer product, the noise reduction is usually integrated in an Application Specific Integrated Circuit (ASIC) chip, such as a video decoding chip for digital set-top boxes and advanced television sets. Function modules in an ASIC chip are normally tested and emulated using an FPGAs before it is ready for mass production. An FPGA prototype also provides good reference for the final gate count for an ASIC implementation. Therefore, the proposed algorithm is first implemented using FPGAs.[14][38][37]

6.1.1 Prototype Description

The FPGA prototype resides in a chassis with especially design backplane. The system consists of four plug-in Printed Circuit Boards (PCB)s to the backplane. The algorithm is implemented in the FPGAs on the PCBs. The prototype takes a digitized video signal input in CCIR601/656 format, and outputs a composite analog video signal. The logic connection of the prototype is illustrated in Figure 6.1. Each block in the diagram represents one plug-in card. The inter-board connection is implemented by the backplane on the chassis.

The backplane for the chassis is especially designed for video processing. It



Figure 6.1: Block Diagram for Hardware Prototype

can support up to eight plug-in cards. It has a 30-bit serial bus for parallel luminance and chrominance baseband video signals. The baseband video is transmitted from board to board via this bus. It also has an 8-bit serial control bus which support control signal exchange between boards. All the boards share a global clock and a power supply bus. The global clock bus is impedance controlled and terminated during PCB layout to ensure high speed clock signal transmission with low skew.

The first board decodes the CCIR601/656 signal to luminance and chrominance baseband signals. It then extracts the synchronization signals, such as the field signal, horizontal sync and vertical sync signals. All these functions which only take 32 flip-flops are implemented in a small scale FPGA chip.

The motion detection and the temporal filter are implemented in the second board. The input video signal from the video bus are first fed into three frame buffers. These along with the present frame form four video frames. The frame buffers are controlled by sync signals transmitted from the first board via the control bus. The motion detection and temporal filter are implemented in one FPGA chip. The spatial filter is implemented on the third board. The signal processed by temporal filter is passed to the board via data bus, and the motion detection results are carried on the control bus. Dedicated line buffer chips are used to form the 5x3 window for spatial filter.

The fourth board functions as a D/A convertor. It takes the filtered digital video signal and converted it into baseband composite analog video signals using Philips NTSC encoder chip.

6.1.2 Design Consideration and Algorithm Optimization

The proposed algorithm is implemented in Altera Flex series FPGA. The programs are written in Altera Hardware Description Language (AHDL). The challenge is to convert the equations into logic designs that efficiently fit into the FPGA architecture. For most parts, the program is a straight forward digital circuit design. Synchronized timing design is used to ensure timing consistency, and some techniques, such as mathematic approximation and multiplexing, are used to reduce the gate count.[17]

Mathematical Approximations

For the Flex FPGA architecture, arithmetic operations, like multiplication and division, are difficult to implement and use many gate counts. Adders and subtractors on the other hand are easy to implement, and there exist many reference designs for these functional units. Therefore, it is advantageous to use adders to implement the function of multiply or divide operations.

For the temporal filter, the average of 2,3,and 4 pixel values has to be calculated. culated. For the spatial filer, the average of 1 to 9 pixels has to be calculated. Therefore, we do not need a general purpose divider. As long as the circuit can calculate a division by a number from 1 to 9, the filters can be implemented. In logic design, multiplication and division by 2^n is easy to implement as it can be achieved by left or right shifting the number by n bits. Based on the theory that any number between 0 and 1 can be represented by a series of fractions, as shown in Equation 6.1,

$$x = \sum_{i=0}^{\infty} a_i \frac{1}{n^i} \tag{6.1}$$

We write some of the division operations as in Equation 6.2 when n is 2.

$$\frac{1}{3} = \frac{1}{2^2} + \frac{1}{2^4} + \frac{1}{2^6} + \frac{1}{2^8} + \frac{1}{2^{10}}$$

$$\frac{1}{5} = \frac{1}{2^3} + \frac{1}{2^4} + \frac{1}{2^7} + \frac{1}{2^8}$$

$$\frac{1}{6} = \frac{1}{2^3} + \frac{1}{2^5} + \frac{1}{2^7} + \frac{1}{2^9} + \frac{1}{2^{11}}$$

$$\frac{1}{7} = \frac{1}{2^3} + \frac{1}{2^6} + \frac{1}{2^9}$$

$$\frac{1}{9} = \frac{1}{2^4} + \frac{1}{2^5} + \frac{1}{2^6} + \frac{1}{2^{10}} + \frac{1}{2^{11}} + \frac{1}{2^{12}}$$
(6.2)

The division is implemented using 5 adders. The approximation is based on the knowledge that the quotient has 8-bit resolution. As long as the approximation error is less than one bit, this approach will not introduce any noise into the operation.

Multiplexing

One of the technique to reduce the gate count is to increase the processing speed. Therefore the gate resource can be used by different functional modules in a time share manner. For example, in this design, both chrominance signals Cr and Cb undergo the same processing. The straight forward implementation is to use two identical functional modules for each signal. However, since the chrominance signal has half the resolution as the luminance signal, the clock rate of the processing for chrominance signal is also half that of the luminance clock rate. In this design, the two chrominance signals are multiplexed into one signal stream and go through a circuit that operates at the luminance clock rate. After the processing, the multiplexed signal is demultiplexed and downsampled into the regular chrominance sample rate. In this way, the gate count for the chrominance processing is cut to half.

The multiplexing technique will certainly reduce the cost as less gate count means a smaller die size is needed for the silicon. However, there is a limit for the cost saving. The multiplexing technique requires higher clock speeds. When the speed increases to above certain level, better semiconductor manufacture technologies have to be required to deal with the power consumption and timing reliability issues. This may increase the silicon manufacture cost. In addition, multiplexing increases the complexity of the design, which means more effort is required in the design, simulation and testing.

In the implementation for the proposed algorithm, both mathematical ap-

proximation techniques and multiplexing techniques brought gate count reduction to this design. The proposed algorithm implemented in FPGA occupies about 120,000 gates, excluding the gates for memories.

6.2 Implementation On VLIW based DSP

The emergience of the Very-Long Instruction Word (VLIW) media processors made it possible to implement complicated video processing in software. The VLIW architecture is an alternative for exploiting instruction-level parallelism in programs, that is, for executing more than one basic (primitive) instruction at a time. These processors contain multiple functional units. They fetch from the instruction cache a Very-Long Instruction Word containing several primitive instructions, and dispatch the entire VLIW for parallel execution. These capabilities are exploited by a compiler which generates codes that have grouped together independent primitive instructions that are executable in parallel. The processors have relatively simple control logic because they do not perform any dynamic scheduling nor reordering of operations (as is the case in most contemporary superscalar processors). VLIW has been described as a natural successor to RISC, because it moves complexity from the hardware to the compiler, allowing simpler, faster processors. However, the programming for this kind of processor becomes more difficult.

6.2.1 Implementation Platform

There are few VLIW chips available in the market. These include TI's C62x family, Philips' TriMedia TM1300 familiy and Equator's MAP-CA. Compared to the MAP-CA, the C62x family lacks a high bandwidth I/O port, and the TriMedia chip does not provide adequate computational power. Therefore, the MAP-CA is chosen for the implementation platform. Our noise reduction algorithm is implemented as a software module for digital settop boxes based on the MAP-CA processor.

The MAP-CA processor consists of a VLIW core, programmable co-processors, on-chip memories, and I/O interfaces. The VLIW core executes four operations in parallel and supports partitioned SIMD operations for 8, 16, 32, and 64-bit data types. There are 128 32-bit registers usable separately or in pairs as 64-bit registers, 32 1-bit predicate registers and 8 special 128-bit registers. Co-processors on the MAP-CA helps accelerate serial operations like variable length encoding/decoding and video filtering.Several audio/video interfaces are supported, including CCIR601/656 input and output. These I/O functions execute in parallel with the CPU and eliminate the need for several external ASICs with their associated cost and bandwidth issues. A glueless SDRAM controller supports access up to 133 MHz SDRAM. The MAP-CA digital signal processor supports a 128 MB memory size. A 32-bit 33/66MHz PCI bus interface is also supported.

The MAP-CA processor operations are primarily 3-oprand RISC operations. As in a typical RISC architecture, load and store operations are the only means
of referencing memory. The MAP-CA DSP has four functional units: two I-ALUs and two IG-ALUs. Each I-ALU contains a load-store unit, an integer ALU, and a branch unit. Each IG-ALU contains an integer/graphics unit and a multimedia operation unit. The I-ALU and IG-ALU support different operations, but many integer and logic operations are implemented in both units. This overlap allows the compiler to schedule more operations in parallel and make more efficient use of all the functional units.

6.2.2 Design Considerations and Software Description

Programming of VLIW DSPs requires skills. Straight-forward implementation may not fully take advantage of the processing parallelism. Optimization in code structure and coding techniques can sometimes cut the processing cycle count to as low as 20%. This section discusses a few techniques to reduce the cycle count.

Memory Structure

There are three levels of memories in the MAP-CA processor: a register file includes 128 32-bit registers, 32 KB data cache and 32KB instruction cache, and on-board memory. Both instruction and initialization data are stored in the on-board memory. During the execution of the software, these data are fetched from the memory to some of the registers, and a copy is saved in cache. The cache is a temporal storage space embedded in the processor. It stores recently used data and instructions. It takes much less time for the processor to fetch data from the cache than from the on-board memory. The register file is the fastest memory for operation. For RISC like DSPs, all operations have to be done in the registers. All data involved in the operation have to be loaded into the registers first with load/store instructions.

When some data are required by the program, the processor will first look in the register file. If they are in the register file, the operation is applied on the data. If they are not in the register file, the processor will look for the data in the cache. If the data are in cache, the data are loaded from the cache to registers for operation. If the data are not in the cache, it is called a cache miss. When a cache miss occurs, the processor has to halt to wait for the data fetching from the on-board memory. New data to be fetched from memory will be saved as a copy in the cache. If the cache is full, some old data in the cache will be copied back into memory, and removed from the cache.

Based on the memory structure, the software design should consider the data flow between memory, cache and register files. The data exchange should be kept minimal so as to reduce data cache miss. If the temporal filter and the spatial filter are implemented in serial the data cache miss will be significant. After the whole frame is processed by temporal filter, the intermediate results from the temporal filter are refreshed out of cache and copied back to the on-board memory. They should then be moved back to cache when needed for spatial filtering. Instead of following this process, we divide each frame into small blocks such that each block data can fit in the data cache. The complete noise reduction process is done on each block before the processing moves to the next block. This structure may increase the software complicity as the boundary of each block has to be specially treated. However, it is worth doing since the data cache missed and the instruction cache misses are dramatically decreased. The total cycles used by the software are also decreases. The software therefore is able to process real-time video.

Under this memory structure, cache coherency is another important issue to consider when sharing data with co-processors, such as input and output ports. The co-processors do not have access to data cache. They only access the on-board memory. If a co-processor needs to read data that the DSP core just processed, it is the program's responsibility to copy the data in the cache back to the memory.

Synchronization with Co-processors

There are several co-processors for the VLIW core, such as the video-in, the videoout and the data streamer. They are optimized for certain applications to help improve the overall performance. The communication and synchronization between the core processor, the internal co-processor and the external devices are via interrupts and buffers.

The MAP-CA DSP has a flexible interrupt structure. Interrupts and exceptions internal to the core are reflected directly in the system registers. All other interrupts from on-chip devices and PCI interrupts from external devices are gathered by an on-chip interrupt controller. The interrupt controller also provides a number of software interrupts. Routing, masking and prioritization of interrupts is completely software programmable. Each of the interrupts handled by the interrupt controller can be individually masked, or routed to one of four core interrupts or to one of two PCI interrupt signals. It is up to the program to set the interrupt priority and mask other interrupts.

Each co-processor can trigger interrupts when some pre-determined events happen. Once an interrupt occurs, the software reads the system register to decide what devices triggered the interrupt, and calls the corresponding interrupt service routine to handle the interrupt. The programmer can set a flag in the interrupt service routine to report an event, or use the timing information the interrupt provides to control the flow of the program.

Buffering is another commonly used technique to exchange bulk data between processors. The idea for ping-pong buffering is to make both core processor and the co-processor working at the same time, but on different data. In this design, Ping-Pong buffers are used to transfer data between DSP core and the I/O coprocessors. While the video-in co-processor is feeding data into buffer1, the core processor operates on buffer2. When the video-in co-processor fills the buffer1, it then goes on to fill buffer2. The DSP then goes ahead to process the data just filled in buffer1. This way, both processors can work in parallel so that improves the data throughput of the processing. Unlike hardware implementation where the processing has constant delay, the delay for the software processing varies from frame to frame. If the buffer size is well chosen, this jitter in the delay can be smoothed out while maintaining the constant data throughput.

Algorithm Optimization

Programming style plays an important role in code efficiency for VLIW programming. It is known that code branching and jumping dramatically decrease the code efficiency, and that the VLIW processor does not perform well on bit operation. Based on these characteristic of the VILW architecture, some operations of our proposed algorithm are modified.

In the isolated pattern recognition routine part of the motion detection algorithm, the proposed method is a pattern match operation on six patterns. This operation is simple when implemented in FPGAs, but takes much computational power on VLIW processors because of the frequent code branchings and bit extractions for bit operations. Therefore, instead of the pattern matching operation, we use Maximum and Minimum rank statistic filters. These give equivalent results but take much less cycle to compute.

Recall that in the edge adaptive spatial filter, all the pixels in the sliding window are compared with the central pixel to determine if this pixel is involved in the averaging. After the comparison are carried, the processing jumps to different branches depending on the comparison results. To avoid the coding branching caused by the jump, all the pixels follows the same operation no matter what the comparison results are. All the pixels are included in the average after the pixel values are multiplied by the comparison results. If the pixel is over threshold, the comparison result is zero, so the weighted pixel becomes zero after the multiplication. This way, a mathematic computation replaces the condition judgement which causes code jump. This is a frequently used technique in the VLIW programming.

By applying the above mentioned techniques in code structure, data access pattern and algorithm optimization, the proposed noise reduction is able to run on the MAP-CA chip in real time.

Chapter 7

Objective and Subjective

Evaluation

The evaluation of noise reduction on real video presented in real time consists of two parts: objective and subjective evaluation. For objective evaluation, the hardware prototype was tested in by standard TV measurement equipment Tektronix VM700. Subjective evaluation were conducted on both hardware and software prototypes on sample noisy pictures and live video feed. A quantized subjective evaluation approach is also investigated in this chapter.

7.1 Experiment Setup

The experiment setup for objective evaluation is illustrated in Figure 7.1. The video source is generated from SONY Betacam video cassette recorder. The baseband

composite video signal passes through a Tektronix 1430 random noise generator, which adds thermal noise into the video signal. The noise impaired video signal is fed into a video A/D converter, the output of which is a parallel digital signal in CCIR601 format. The digital signal is split into two paths. In one path the signal is converted back into analog format by WardLabs D/A converter. In the other path, the signal is processed by our noise reduction, before it enters the WardLabs D/A converter which is identical to the one in path one.



Figure 7.1: Illustration of Experiment Setup

Figure 7.2 shows the waveform of one line of NTSC video, illustrating how noise generator inserted noise into video signal. In each scan line, a portion of video signal is superimposed by noise.



Figure 7.2: Waveform of One Line of Noise Corrupted NTSC Video Signal

7.1.1 Proper Measurement Settings

The noise is measured by Tektronix NTSC video measurement equipment VM700. The input to the equipment goes through a gate in time domain, which truncates a portion of the signal in each scan line. The gate position in this test is shown in Figure 7.2 by the two parallel vertical lines. The signal between these lines are measured, which in this case is the noise corrupted part.

There are several filter settings in the equipment. The following sections discuss how the setting affects the test result. Figure 7.3 shows the noise signal measured by VM700 without any filter enabled. The noise signal is a broadband white signal, and the level decays around 4.2 MHz.



Figure 7.3: Noise Measurment Without Filter

Lowpass Filter

There are two low pass filters to choose from. One lowpass filter has the cut-off frequency at 4.2 MHz, and the other cut-off frequency is at 5.0 MHz. Since the NTSC signal bandwidth is 4.2 MHz, we choose 4.2 MHz lowpass filter setting as we are only interested in the noise reduction in the valid video bandwidth.

Figure 7.4 shows the noise spectrum after the lowpass filter is turned on. It is noted that the noise outside the 4.2 MHz band is cut out.



Figure 7.4: Noise Measurement with Lowpass Filetr at 4.2MHz

$F_{(sc)}$ Trap Notch Filter

The $F_{(sc)}$ trap filter is a notch filter to take out the color carrier. By turning on the filter, only luminance signal is examined. Although noise reduction also operates on chrominance signal, the effects of the luminance signal and chrominance signal are not linearly combined. The resultant lumiance signal of the notch filter is a better representation of the effects of real noise reduction if the $F_{(sc)}$ trap filter is turned on.

Figure 7.5 shows the noise spectrum after the low pass filtering and color carrier notch filtering. This is the conventional setting to measure noise reduction for NTSC luminance signals. This setting is used for all the measurements in the test.



Figure 7.5: Noise Measurement with the 4.2MHz Lowpass Filter and $F_{(sc)}$ Trap Filter

7.1.2 A/D and D/A Convertors Performance Analysis

Two A/D converters may be used in the testing. One is a SONY A/D convertor and the other is a Miranda A/D convertor. To estimate how the A/D and the D/A convertors affect the picture quality in these tests, we measure the noise level before and after the signals pass through the A/D and the D/A convertors as shown in Table 7.1. The data in the column *noise* are obtained by putting the VM700 probe at position 1 in Figure 7.1 and data in columns *SONY* and *Miranda* are obtained by putting the VM700 probe at the position 2 in Figure 7.1 while using the SONY and Miranda A/D convertors.

Noise (dB)	SONY (dB)	Difference (dB)	Miranda (dB)	Difference (dB)
20	23.9	3.9	23.7	3.7
25	28.9	3.9	28.5	3.5
30	33.8	3.8	33.5	3.5
35	38.9	3.9	38.7	3.7
40	44.1	4.1	44.1	4.1
45	49.1	4.1	49.5	4.5
50	52.8	2.8	54.1	4.1
55	54.8	-0.2	57.5	2.5
60	57.2	-2.8	59	-1

Table 7.1: Noise Reduction by A/D Convertors

Normally when the signal goes through a system, its noise level increases as the circuits also introduce noise. However, the measured data show that the noise level decreases after the signal goes through the A/D and D/A convertors. We found out that A/D convertors carry out noise reduction on the signal. An A/D convertor separates the color signal from the composite video signal by detecting the phase of the color burst. A noise corrupted color burst may result in serious color phase noise or color shift in the digitized image. Therefore, a common practice for A/D convertor is to first apply a simple filter on the input signal, so that it could get a relatively clear color burst to ensure the accuracy of color separation.

Figure 7.6 shows how the noise reduction introduced by A/D convertors varies as the inserted noise level increases. Both A/D convertors achieve around 4 dB noise reduction when the noise level is less than 45 dB. When the SNR is higher than 45 dB, which means the picture quality gets better, both A/D convertors do less noise reduction. When the noise introduced by the processing circuitry overrides the noise reduction effects, the noise level becomes worse than the input level. The figure also shows that the Sony A/D convertor has a relative even noise reduction profile when the noise input ranges from 20 dB to 45 dB, and does less noise reduction when picture quality is high. It shows that the Sony A/D convertor has better performance for this noise reduction test. Therefore, the Sony A/D convertor is used in the following measurements.



Figure 7.6: Noise Reduction Caused by A/D Convertors

7.2 Measurement Result and Analysis

To measure the noise reduction achieved solely by our noise reduction circuits, we must eliminate the noise reduction influence of the A/D convertor. The filtered data are obtained by putting the VM700 probe at position 2 in Figure 7.1 and the unfiltered data are obtained by putting the VM700 probe at position 3 in Figure 7.1. The filtered and unfiltered data are shown in Table 7.2.

Unfiltered (dB)	Filtered (dB)	Improvement (dB)
30.8	31.1	0.3
31.7	32.8	1.1
32.8	34.3	1.5
33.8	35.3	1.5
34.7	37	2.3
35.8	38.9	3.1
36.8	40.2	3.4
37.4	42	4.6
38.9	43.2	4.3
39.6	44.3	4.7
40.6	45.7	5.1
41.3	46.4	5.1
42.6	47.5	4.9
43.7	48.4	4.7
44.7	49.2	4.5
46	49.8	3.8
47.1	50.9	3.8
47.8	51.2	3.4
48.5	51.7	3.2
49.4	51.9	2.5
50.5	52.1	1.6
51.6	52.3	0.7
52.3	52.8	0.5

Table 7.2: Noise Reduction Measurement Results

Figure 7.7 shows how the noise reduction results vary as the input noise level changes. The results show that the implementation matches the design expectations. The prototype reduces the noise when the input noise is in the range of 30 to 50 dB. When the input noise is heavy, ie. the SNR is low, the algorithm does little noise reduction because the picture is too corrupted with noise and thus the motion detection and the edge detection processes may fail to perform well. When the picture is clean, the algorithm also does little noise reduction. The performance peaks when the input noise level is around 41 dB. This is below what the FCC regulations require for picture quality. By applying the noise reduction, the picture quality can be lifted to meet the FCC requirement.(Presently 45 dB)



Figure 7.7: Noise Reduction Caused by A/D Convertor

7.3 Subjective Noise Reduction Results

The ultimate performance evaluation for noise reduction is the subjective reaction of viewers. Considering the human visual system, which is a nonlinear system, a lot of visual phenomena can not be described totally by an objective criterion. Therefore, subjective tests are important for a fair evaluation of video filters.[3]

For noise reduction, a subjectively better picture implies two aspects. One

is that the picture is cleaner than the original picture, which means the noise is smoothed out. The other aspect is that the picture should not be degraded after processing. This implies that the picture details and textures are well kept, no smearing edges are introduced and no motion lagging artifacts are visible.

The subjective testing conducted on both prototypes includes two different settings. One is to play a pre-recorded video tape, and run the signal through our noise reduction system. The tape had video sequences corrupted with different kinds of noise at various levels. The other one is to feed the live transmission signal into the noise reduction system and observe the noise reduction for many different scenes.

The first test is best used for observing how the noise reduction prototypes perform with different noise kinds and levels. Snow, CTB, CSO, blockiness and mosquito noise are tested. The results show that the noise reduction prototypes perform the best for snow noise. The prototypes also deliver moderate noise reduction on CTB noise, but are less effective for CSO noise. For digital noise, they give an improved result on mosquito noise partly because the mosquito noise has similar characteristics as snow noise. Clean signal is also run through the prototypes. It is found that the noise reduction algorithm introduce no artifacts into the clean picture. It is one of the most outstanding merits of this algorithm.

The live television feed tested the noise reduction's performance for a very large numbers of different scenes. The results show that our noise reduction algorithm is most effective for movie and news contents with slow motion, and for cartoon with not much detail and texture. Commercials, sports and action movies are the toughest contents to deal with. The noise reduction is not as impressive on these contents. However, there is no lagging and ghosting motion artifacts found during the process.

Efforts were also made to find a quantitive scale for subjective evaluation. A set of reference video with the same content but with different noise levels are used for the evaluation. The reference sequences are fed into the noise reduction prototype and displayed on one monitor, and the original pictures are displayed on another monitor. The viewers are asked to match the processed video to one of the sequence in the original reference that they think have the same subjective picture quality. The difference in the noise levels between the input video and the matched reference video are considered as the improvement made by the noise reduction system. The tests were conducted on randomly selected visitors to the lab during an open house session. All viewers preferred the processed video to the original picture. The noise reduction of the subjective evaluation varies from 3 dB to 6dB on snow noise and mixed CTB and snow noise corrupted pictures.

Both hardware and software prototypes are compared with other existing noise reduction products in the market. There are three types of video noise reduction available. One kind of products are PC based softwares, such as SyntheticAperture's video Finesse. Although some of the products claim better picture quality improvement than the proposed algorithm, they can not process the picture in real time. The other category includes the high end devices for professional video production. These products usually utilize sophisticated algorithms and give good results, but they also have high costs which range from three to twenty thousand dollars. There are also some companies put noise reduction into their consumer products. Both Sony and Panasonic have integrated noise reduction into their TV sets. However, our proposed algorithm gives better visual improvement than those products. If implemented in ASIC, compares to a noise reduction chip used by Panasonic which utilizes die size of 19.5 mm^2 in 0.6 μm technology, our proposed algorithm has similar cost, but yields high noise reduction.

Chapter 8

Conclusion and Future Work Suggestion

Noise reduction is an important aspect of video viewing. It improves the picture quality so that the viewer has better viewing experience. Based on previous work on noise reduction, this thesis proposes a motion adaptive temporal-spatial noise reduction filter. With its 3-step motion detection algorithm, high resolution yet noise insensitive motion detection results are achieved. Therefore, the filter is able to achieve intensive noise reduction while introducing minimum motion artifacts.

In addition to computer simulation on some typical images, a hardware and a software prototypes are built so that subjective evaluation can be done on realtime video sequences. Both objective measurements and subjective evaluations are conducted on the noise reduction algorithm. The measurements show that the algorithm improves the picture quality up to 5 dB when the picture CNR is in the range of 30 dB to 55 dB. Subjective evaluation also reveal that this algorithm introduces the least artifact compared with other noise reduction devices in the market, and improves the CNR up to 6 dB in the subject evaluation.

The prototypes also show that the filter can be implemented in consumer products with an acceptable cost. If implemented in an ASIC chip, the total gate count is about 120,000 excluding the memory. It can also be implemented as a real-time software module on VLIW DSP based platform without extra hardware cost.

8.1 Future Work Suggestion

Although the work done for this thesis achieved the goals set before the project, there are a few areas worth further research.

A more sophisticated motion search algorithm can be integrated with this noise reduction filter. The current filter only adopted motion detection because motion search is significantly more complicated and requires higher cost for implementation. In the future, this cost will be greatly reduced as the technology in the semiconductor industry improves and the DSP chips acquire higher speeds. With the motion search implemented, better temporal noise reduction can be achieved.

More research can be conducted for pre/post-processing for digital video. The proposed algorithm is focused at analog noise reduction, and it may not be optimal for digital noise reduction. Research on how noise reduction affects the video coding efficiency and digital noise characteristic will help improve the noise reduction algorithm so that better performance in digital applications is obtained.

Further research can be also carried on recursive temporal filtering. The proposed algorithm does not consider recursive temporal filter because of the complexity in the hardware implementation. However, implementations using DSPs are becoming relatively easy as recent DSPs provide computations such as multiplications and divisions. Research may be done on how the feedback factor K in the recursive filter affects the noise reduction performance, and how should the value of K changes as the motion and input noise level varies.

Bibliography

- S. N. Efstratiadis A. K. Katsaggelos, R. P. Kleihorst and R. L. Lagendijk. Adaptive image sequence noise filtering methods. In Proc. SPIE Conf. Visual Comm. and Image Proc., Nov. 1991.
- [2] Ayten Atasoy Ali Gangal, Temel kayikcioglu and Mahmut Ozer. Improvement of video signal-to-noise ratio with adaptive recursive filtering. In Meditarranean Electrotechnical Conference, volume 3, 1996.
- [3] A. Amer and H. Schroder. A new video noise reduction algorithm using spatial subbands. In Proceeding of the Third IEEE International Conference on Electronics, Circuits ans System, volume 1, pages 45-48, 1996.
- [4] G. R. Arce. Multistage order statistic filters for image sequence processing. In IEEE Trans. Signal Proc., volume 39, pages 1146-1163. IEEE, 1991.
- [5] T. Jarske K. Ostamo B. Alp, P. Haavisto and Y. Neuvo. Median based algorithms for image sequence processing. In Proc. DPIE Visual Comm. and Image Proc., pages 122-134, Octorber 1990.

- [6] Hermann J. Weckenbrock Barbara J. Roeder, Leopold A. Harwood. Interfield image motion detector for video signals. US Patent 4,661,853, April 1987.
- [7] J. Boyce. Noise reduction of image sequences using adaptive motion compensated frame averaging. In Proc. IEEE Int. Conf. Acoust. Speech and Sign. Proc., volume 3, pages 461-464, March 1992.
- [8] S. Chang C. L. Lee and C. W. Jen. Motion detection and motion adaptive proscan conversion. In *IEEE international Symposium on Circuit and Systems*, volume 1, pages 666-669, 1991.
- [9] P. Chan and J. S. Lim. One-dimensional processing for adaptive image restoration. In *IEEE Trans. Acoust. Speech and Sign. Proc.*, volume 33, pages 117-126. IEEE, Feb. 1985.
- [10] T. J. Dennis. Non-linear temporal filter for television picture noise reduction.
 In *IEEE Proc.*, volume 127G, pages 52-56, 1980.
- [11] T. Doyle and P. Frencken. Median filtering of television images. In Proc. IEEE Int. Conf. on Consummer Elec., pages 186–187, 1986.
- [12] E. Dubois and S. Sabri. Noise reduction in image sequences using motioncompensated temporal filtering. In *IEEE Trans. Comm.*, pages 826-831. July 1984.
- [13] L. S. Favis and A. Rosenfield. Noise cleaning by iterated local averaging. In IEEE Trans. Syst. Man. and Cybern., volume 8, pages 705-710. IEEE, 1978.

- [14] M. Larragy G. de Haan, T. G. Kwaaitaal-Spassova and O. A. Ojo. Memory integrated noise reduction ic for television. In *IEEE Transactions on Con*sumer Electronics, volume 42. IEEE, May 1996.
- [15] N. C. Gallagher G. R. Arce and T. A. Nodes. Median filters: Theory for one or two dimensioanl filters. In T. S. Huang, editor, Advances in Computer Vision and Image Processing. JAI Press, 1986.
- [16] R. C. Gonzalez and R. E. Woods. Digital Image Processing. Addison-Wesley, 1992.
- [17] L. Kaufman H. Chen, A. Li and J. Hale. A fast filtering algorithm for image enhancement. In *IEEE Transactions on Medical Imaging*, volume 13. IEEE, September 1994.
- [18] Si Jun Huang. Adaptive noise reduction and image sharpening for digital video compression. In IEEE International Conference on System, Man, and Cybernetics, Computational Cybernetics and Simulation, volume 4, pages 3142-3147, 1997.
- [19] Ward Laboratories Inc. http://www.wardlabs.com.
- [20] Klaus Jostschulte and Aishy Amer. A new cascaded spatio-temporal noise reduction scheme for interlaced video. In Proceedings of International Conference on Image Processing, volume 2, pages 493-497, 1998.
- [21] M. Schu K. Jostschulte, A. Amer and H. Schroder. A subband based spatio-

temporal noise reduction technique for interlaced video signal. In International Conference on Consumer Electronics, pages 438-439. ICCE, Digest of Technical Papers, 1998.

- [22] T. Koivunen. Motion detection of an interlaced video signal. In IEEE Transactions on Consumer Electronics, volume 40, pages 753-760. IEEE, August 1994.
- [23] Tero Koivunen. A noise-insensitive motion detector. In IEEE Trans. Consumer Electronics, volume 38, pages 168–174. IEEE, August 1992.
- [24] M. K. Ozkan M. I. Sesan and S. V. Fogal. Temporally adaptive filtering of noisy image sequences using a robust motion estimation algorithm. In Proc. IEEE Int. Conf. Acoust. Speech, Singal Processing, pages 2429-2432, 1991.
- [25] V. D'Alto M. Mancuso and R. Poluzzi. Fuzzy edge-oriented motion-adaptive noise reduction and scanning rate conversion. In IEEE Asia-Pacific Conference on Circuits and System, pages 652-656, 1994.
- [26] Petros Maragos and Ronald W. Schafer. Morphological systems for multidimensional signal processing. In *Proceedings of the IEEE*, volume 78, pages 690-710. IEEE, April 1990.
- [27] D. Martinez and J. S. Lim. Implicit motion compensated noise reduction of motion video scenes. In Proc. IEEE Int. Conf. Acoust. Speech, Signal Proc., pages 375-378, 1995.

- [28] M. Ibrahim Sezan Mehmet K. Ozkan and A. Murat Tekalp. Adaptive motioncompensated filtering of noisy image sequences. In *IEEE Trans. Circuits* and System for Video Technology, volume 13, pages 277-290. IEEE, August 1993.
- [29] Martin R. Trump Peter C. Michael, Richard J. Taylor. Video noise reduction. US Patent 4,240,106, December 1980.
- [30] Richard Alan Peters. A new algorithm for image noise reduction using mathematical morphology. In *IEEE Trans. Image Processing*, volume 4, pages 554-568. IEEE, May 1995.
- [31] Ioannis Pitas and Anastasios N. Venetsanopoulos. Order statistics in digital image processing. In *Proceedings of the IEEE*, volume 80, pages 1892–1921.
 IEEE, December 1992.
- [32] William K. Pratt. Digital Image Processing. John Wiley & Sons, second edition, 1991.
- [33] Pingnan Shi Xiaoli Li Rabab Ward, Julong Du. Noise reduction for video signal.US Patent 6,061,100, May 2000.
- [34] R. Samy. An adaptive image sequence fitlering scheme based on motion detection. In SPIE, volume 596, pages 135-144, 1985.
- [35] C. P. Sandbank, editor. Digital Television. John Wiley & Sons, 1990.
- [36] A. Murat Tekalp. Digital Video Processing. Prentice Hall, 1995.

- [37] D. Teytelman and E. Feria. A simple real-time digital video noise reduction system. In IEEE Proceedings on Aerospace and Electronics Conference, volume 1, pages 507-511, 1994.
- [38] Dmitry Teytelman and Erlan H. Feria. A simple real-time digital video noise reduction system. In IEEE Proceedings on Aerospace and Electronics Conference, volume 1, pages 507-511, 1994.
- [39] M. Unser and M. Eden. Weighted averaging of a set of noisy images for maximum signal-to-noise ratio. In IEEE Trans. on Acoust., Speech and Signal Proc., pages 890-895. IEEE, 1990.
- [40] James Farmer Walter Ciciora and David Large. Modern Cable Television Technology. Morgan Kaufmann Publishers, Inc., 1999.