

Ph.D Thesis

**Software Hardware Co-development for
SMRT based Texture Analysis
Applied in Medical and Non Medical Images**



Manju B

School of Engineering
Cochin University of Science and Technology,
Kochi - 682 022
August 21, 2019

**Software Hardware Co-development for
SMRT based Texture Analysis**
Applied in Medical and Non Medical Images

Thesis submitted in partial fulfillment of the requirements
for the award of the Degree of
Doctor of Philosophy
under the Faculty of Engineering

By

Manju B
(Reg. No. 4034)

Under the guidance of
Prof. R. Gopikakumari



School of Engineering
Cochin University of Science and Technology,
Kochi - 682 022

October 10, 2018

**Software Hardware Co-development for
SMRT based Texture Analysis**
Applied in Medical and Non Medical Images
Ph.D Thesis under the Faculty of Engineering

Author

Manju B
Research Scholar
Division of Electronics Engineering
School of Engineering
Cochin University of Science and Technology.

Supervising Guide

Dr. R. Gopikakumari
Professor
Division of Electronics Engineering
School of Engineering
Cochin University of Science and Technology.



**School of Engineering
Cochin University of Science and Technology,
Kochi - 682 022**

School of Engineering
Cochin University of Science and Technology,
Kochi - 682 022



Certificate

Certified that the thesis titled **Software Hardware Co-development for SMRT based Texture Analysis Applied in Medical and Non Medical Images** submitted by Manju B., is an authentic record of the research work carried out by her under my supervision for the award of degree of Doctor of Philosophy in Faculty of Engineering, Cochin University of Science and Technology. The work presented in this thesis or part thereof has not been presented for any other degree. All the relevant corrections and modifications suggested by the audience during the pre-synopsis seminar have been incorporated in the thesis and recommended by the Doctoral Committee.

Prof. (Dr.) R Gopikakumari
(Supervising Guide)

Kochi,
10 October, 2018.

Declaration

It is declared that the thesis titled **Software Hardware Co-development for SMRT based Texture Analysis Applied in Medical and Non Medical Images** is an authentic record of the research work done by me under the guidance of Prof. R. Gopikakumari at Division of Electronics Engineering, School of Engineering, Cochin University of Science and Technology. This work or any part thereof has not been presented to any other institution for any other degree.

Manju B
Reg. No. 4034
Research Scholar
School of Engineering
Cochin University of Science and Technology.

Curriculum Vitae

Manju B. was born in Ernakulam District of Kerala, India in 1975. She obtained her B. Tech Degree in Electrical Engineering from M.G. University, in the year 1996 and further completed M. Tech in Industrial Electronics securing the third rank from Viswesariah Technological University, Belgaum in the year 2007. She got selected through PSC, Govt. of Kerala for appointment as Lecturer in Electrical Engineering in 1999. Since then, she has been teaching in Government Engineering Colleges in Kerala. Presently she is working as Assistant Professor in Electrical Engineering Department at Government Engineering College, Thrissur, Kerala. She has so far published two research papers in international journals and presented more than fifteen papers in international/national conferences. Her areas of interest include Image Processing, Intelligent Processing, Digital System Design, Parallel Computing and Embedded Systems.

Acknowledgements

At the outset, I thank God Almighty for providing me the opportunity, willpower and knowledge for the successful completion of this research work.

I would like to express my profound gratitude to Dr. R. Gopikakumari, Professor, Division of Electronics Engineering, School of Engineering, Cochin University of Science and Technology, for her valuable guidance, timely advice, suggestions and personal attention as supervising guide and also for being a source of constant inspiration and motivation during the course of this research work. Heartfelt thanks are due to her for allowing me the freedom for pursuing research on my line of thought while providing all the support including spending long hours outside the office hours with me. Without her able guidance and patient support, it would not have been possible for me to complete the work and deliver this thesis.

I would like to acknowledge the help rendered by the Principal and office staff of School of Engineering, CUSAT for providing proper resources for research. I also would like to thank the research committee members and Dr. P. Mythili, Doctoral Committee member for their timely advice and guidance. Thanks are also due to the faculty, office staff and technical staff of Division of Electronics Engineering for the support during the entire duration.

I am much indebted to all my senior research fellows, especially Dr. Rajesh Cheriyan Roy, Dr. Bhadrans V., Dr. K. Meenakshy, Dr. Anishkumar M.S. and Dr. Jaya V.L. who led me through their experience of research in this domain and practical issues beyond textbooks. Their support and guidance was a great help

for me. I thank my fellow research scholars Mini P P and Febina Ikkal for their timely help and suggestions.

My sincere thanks are due to Dr. H. Krishnamoorthy, Consultant Urologist, Lourdes Hospital for providing research ideas on Prostate disease diagnosis. I also thank his team for providing me with the necessary clinical data for research.

I am grateful to the staff of Kerala Agricultural University for providing me the photographic images of coconut in different growth stages.

I would also like to appreciate the help rendered by some of the Under Graduate students Anjali M.P, Anjaly Das, Arya K.A, Aswin Babu and Adarsh M.A of the Division.

I wish to acknowledge the blessings of all my teachers from the elementary level.

I am greatly obliged to Dr. M. Nandakumar, Head, Department of Electrical Engineering, Govt. Engineering College, Thrissur, for the support rendered. I express my deep sense of gratitude to my colleagues and friends at the Department of Electrical Engineering, Govt. Engineering College, Thrissur, for helping me to complete the research.

I am thankful beyond words to my colleague and friend, Dr. Jiji K.S., who was a constant source of inspiration throughout the research period. She was always there to share my anxieties and helped me build my confidence.

A number of my friends have made me feel comfortable during the course of this work. The list seems a long one but Dr. Asha P Rao, K. Prameelakumari, Kalamandalam Lekha, Sneha Sailesh, Deepa Vinod, Umashankar A, Sanilkumar and Bindu Joseph definitely need special mention.

No words can express my indebtedness to my husband Deepu

Surendran, for the motivation, encouragement, support and care extended throughout the course of the study. Thanks to my lovely daughters Gauri and Parvathi who knew well how to take care of the ups and downs of my mood. I am greatly indebted to my sister Meena, brother in law Shibu and my dear nephew Madhav for their deep love, care and patience which helped me to move on with my research work. I like to thank my father in law K.Surendran & mother in law K.Omana (late) for the encouragement and moral support. I also would like to thank all my relatives for their support and prayers.

Finally, I would like to dedicate this work to my parents, Babu & Indira, whose constant encouragement and support have really brought me here.

Manju B

Abstract

Image Analysis is an important tool which uses machine vision technology for differentiating and recognizing different types of images. Texture is an important characteristic present in all images which play a vital role in image analysis. Texture-based image analysis, otherwise termed as texture analysis of images has many applications like image classification, image segmentation, image synthesis etc. Texture analysis of images is used in both medical and non medical fields. The thesis explores the area of texture based image classification.

Difficulties in diagnosing the disease in early stages makes prostate cancer one of the major causes of death among elderly men. An attempt is made to explore the possibilities of texture analysis of abdomen CT images in diagnosing the disease at an early stage. Prostate gland is isolated from the abdomen CT images using active shape model based segmentation techniques. The analysis of the segmented prostate image is done by extracting texture features based on an evolving transform named Sequency based MRT (SMRT). The SMRT based texture features are optimized using Genetic algorithm.

Texture analysis method developed for prostate disease diagnosis is extended for skin cancer detection also. The SMRT based texture analysis method developed for medical applications is extended to identify the maturity stages of coconut, considering its social relevance.

The rapid evolution of new technologies bring new challenges as well. An example of such a challenge is the need to improve the speed of method developed for texture analysis. An attempt is made to increase the speed by using software hardware co-

development. All the image preprocessing tasks are developed in software. The texture feature extraction part is developed in hardware, whose output is fed to software classifier for image identification. In this work, the hardware is implemented as a parallel distributed architecture for further improvement in the processing speed. Visual representation of SMRT coefficients are used to develop the parallel distributed architecture.

The analysis of visual representation used for $N \times N$ SMRT hardware is extended to develop forward and inverse 1-D SMRT algorithm.

Contents

List of Figures	vi
List of Tables	ix
List of Abbreviations	xiii
1 Introduction	1
1.1 Introduction	3
1.2 Digital Imaging Techniques	3
1.3 Digital Image Processing	5
1.4 Image Analysis	8
1.5 Software Hardware Co-development	26
1.6 Motivation	27
1.7 Organization	29
2 Literature Survey	31
2.1 Introduction	32
2.2 Texture Analysis	32
2.3 Image Segmentation	38
2.4 Feature Selection	41
2.5 Classifiers	43

2.6	Applications of Texture Analysis	47
2.7	Software Hardware Co-development	50
2.8	Conclusion	52
3	GA based optimization of SMRT Texture Features using K-NN classifier	53
3.1	Introduction	54
3.2	SMRT based Texture Features	54
3.3	GA based Optimization of 8×8 SMRT Texture Features	57
3.4	Texture Features based on SMRT and Wavelet Transform: A Comparison	63
3.5	Conclusion	67
4	Texture Analysis of Medical and Non Medical Images	68
4.1	Introduction	69
4.2	Prostate Disease Diagnosis	69
4.3	Skin Cancer Detection	84
4.4	Coconut growth stage identification	88
4.5	Conclusion	94
5	Parallel Distributed Architecture for 8×8 SMRT	95
5.1	Introduction	96
5.2	Modified Primitive Symbols based on 2×2 Data	96
5.3	Visual Representation of 8×8 SMRT based on 2×2 Data	98
5.4	Parallel Distributed Architecture for 8×8 SMRT based on 2×2 data	100
5.5	Parallel Distributed Architecture for 8×8 SMRT based on M-spacing Data	112
5.6	Hybrid Architecture for 8×8 SMRT	124
5.7	FPGA Implementation	134
5.8	Conclusion	135

6	Parallel Distributed Architecture for SMRT	136
6.1	Introduction	138
6.2	Hybrid Architecture for $N \times N$ SMRT, N a power of 2	139
6.3	Software Hardware Co-development	154
6.4	Forward and inverse N-point SMRT, N a power of 2	161
6.5	Conclusion	169
7	Results, Conclusion and Future Scope	171
7.1	Summary	172
7.2	Research Contributions	174
7.3	Scope for Future Work	175
	List of Publications	177
	Bibliography	178
	Appendix	201
A	Texture Features	202
A.1	GLCM and Haralick Features	202
A.2	GLRL matrix and texture features	205
A.3	Wavelet Transform based Texture features	206
B	Patient Proforma for CT Based Texture Analysis in Prostate Disease	209
C	Primitive Symbols used in Visual Representation of 2-D DFT coefficients based on 2×2 Data	211
D	Poof of Theorem 5.3	214
D.1	Theorem 5.3	214
D.2	Proof	214
E	Modular Arithmetic	222
E.1	Definition of Congruent Modulo	222

E.2	Inverses in Modular Arithmetic	222
F	4×4 SMRT coefficients	223
F.1	Visual Representation of 4×4 SMRT	223
F.2	Algorithm for Computation of 4×4 SMRT based on 2×2 Data	224
F.3	Algorithm for Computation of 4×4 SMRT based on M-spacing Data	227
G	Computational Illustration of $N \times N$ Hybrid SMRT Architecture, N a power of 2	230
H	Shift invariance in sequence packets of SMRT	242

List of Figures

1.1	Block diagram of Image Analysis System	8
1.2	Pixel Pattern of UMRT Coefficients	17
1.3	(k_1, k_2, p) placement of 8×8 SMRT Coefficients .	19
1.4	Sequency Packets in 8×8 SMRT Coefficients . .	19
3.1	Sample Images from Brodatz Data Base	56
4.1	Proposed Scheme	69
4.2	Samples of Abdomen CT Image Slices	71
4.3	Abdomen CT Image	72
4.4	Result of Edge Based Segmentation	73
4.5	Land mark points in training images	74
4.6	Gray level Profile perpendicular to contour	75
4.7	Iterations	76
4.8	Segmentation Mask obtained	76
4.9	Structure of Skin	84
4.10	Image Samples of Skin Diseases	86
4.11	Proposed Scheme	88
4.12	Different stages of coconut	90
4.13	Cropping Images of different size	91
5.1	Modified Primitive Symbols based on 2×2 Data	97

5.2	Visual Representation of 8x8 SMRT Coefficients	98
5.3	Parallel Distributed Architecture for 8×8 SMRT based on 2×2 Data	101
5.4	M-Spacing Patterns	113
5.5	Parallel Distributed Architecture based on M-spacing Data	115
5.6	Hybrid Architecture based on combination of M- spacing and 2×2 Data	125
6.1	Hybrid Architecture for $N \times N$ SMRT	139
6.2	Sequencies $\{(0,1), (0,2), \dots, (0,2^{v-1})\}, \{(1,0), (2,0),$ $\dots, (2^{v-1},0)\}, \{(M,1), (M,2), \dots, (M,2^{v-1})\}, \{(1,M),$ $(2,M), \dots, (2^{v-1},M)\}$	145
6.3	k and k_{im} values	151
6.4	General Block Diagram of Software Hardware Co- developed System	154
6.5	Co-developed System for Prostate Disease Diagnosis	155
6.6	Sequency distribution in 1-D SMRT for N=4,8,16	162
6.7	Sequency Pattern in 1-D SMRT for N=4	163
6.8	Sequency Pattern in 1-D SMRT for N=8	163
6.9	A and B Matrices	164
A.1	Three level 2-D PSWT decomposition of 128x128 image.	207
A.2	Three level 2-D Wavelet Packet decomposition of 128x128 image	207
C.1	Primitive Symbols based on 2×2 Data	211
F.1	Visual Representation of 4×4 SMRT based on 2×2 Data	224
G.1	Input 8×8 Matrix	230
G.2	Output SMRT Matrix	241
H.1	Sequency Packets in 8x8 SMRT	243

List of Tables

1.1	(k_1, k_2, p) values in placement of 8x8 UMRT Coef- ficients	16
3.1	Computation Times for UMRT and SMRT based Texture Features	57
3.2	Notations of 8×8 UMRT and SMRT Texture Fea- tures	58
3.3	Performance Evaluation of 8×8 SMRT based Tex- ture features	62
3.4	Comparison of Performance of 8x8 SMRT based Texture Descriptors	63
3.5	Image Classification using SMRT(3 features) and Wavelet(3 features) Texture Features and K of K- NN = 1	65
3.6	Comparison of Classification Accuracy of SMRT and Wavelet Texture features with different K val- ues of K-NN classifier	66
4.1	Details of Image Slices	70
4.2	Classification Accuracy of Feature Sets with dif- ferent I and N	77
4.3	Confusion Table for Classes A, B, C and D	78

4.4	SMRT Texture Feature Sets G1, G2 and G3	79
4.5	Confusion Table for G1	80
4.6	Confusion Table for G2	80
4.7	Confusion Table for G3	80
4.8	Comparison Results of SMRT and GLCM texture features	81
4.9	Confusion Table for PNN classifier	82
4.10	Confusion Table for SVM classifier	83
4.11	Confusion Table for BPN classifier	83
4.12	Classification of Prostate Diseases using different classifiers	83
4.13	Comparison of feature sets with different I	87
4.14	Result of GA optimized 16x16 Feature set	87
4.15	Classification Accuracy with different Sub Image Size and Block Size	92
4.16	Classification Accuracy of GA optimized feature set	93
5.1	Primitive Symbols involved in 8x8 SMRT coefficients	99
5.2	8x8 SMRT computations based on 2×2 data . .	111
5.3	Relationship between sequency and frequency parameters	114
5.4	Sequencies and M-spacing Patterns	114
5.5	8x8 SMRT Computations based on M-spacing . .	122
5.6	No. of additions in rows 0, 7 & columns 0, 7 of SMRT matrix for both the architectures	123
5.7	No. of additions in inner rows/ columns of SMRT matrix for both the architectures	124
5.8	8x8 SMRT Computations Based on Hybrid Algorithm	133
5.9	Comparison of the three algorithms based on number of computations	134
5.10	Comparison of the three algorithms based on computation time	134

5.11	Results of Fully Parallel Implementation of 8x8 UMRT	135
6.1	Variables used in the algorithm for different groups in L3G3 - L3G6	148
6.2	Computation time for direct SMRT and SMRT Hardware Algorithms	152
6.3	Comparison of FPGA Implementation for Matrices of different size	153
6.4	Comparison of FPGA Hybrid Architecture Implementation of 8×8 SMRT	153
6.5	GA optimized SMRT Texture feature set	157
6.6	Computation time comparison of SMRT Texture features using Hybrid Algorithm and Direct SMRT algorithm	160
6.7	FPGA Implementation of SMRT texture feature extraction for Prostate Disease Diagnosis	161
6.8	Comparison of Execution time of 1-D SMRT Algorithms	165
6.9	Sign changes and multiplication terms based on sequency in 1-D SMRT inverse computation	167
6.10	Execution time of Inverse 1-D SMRT	169

List of Abbreviations

ASIC	Application Specific Integrated Circuit
ASM	Active Shape Model
AAM	Active Appearance Model
BPN	Back Propagation Neural Network
BPH	Benign Prostate Hyperplasia
CCV	Color Coherence Vector
CT	Computed Tomography
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DIP	Digital Image Processing
DRE	Digital Rectal Examination
FPGA	Field Programmable Gate Array
GA	Genetic Algorithm
GLCM	Gray Level Co-occurrence Matrix
GLRL	Gray Level Run Length Matrix
HT	Haar Transform
KNN	K Nearest Neighbour
MB	Mahalanobis Distance
MRI	Magnetic Resonance Imaging
MRT	Mapped Real Transform
PCA	Principal Component Analysis

PET	Positron Emission Tomography
PNN	Probabilistic Neural Network
PSA	Prostate Specific Antigen
ROI	Region of Interest
SD	Standard Deviation
SMRT	Sequency based MRT
SPECT	Single Photon Emission Computed Tomography
SVM	Support Vector Machine
TRUS	Trans Rectal Ultra Sound
UMRT	Unique MRT
VHDL	VHSIC Hardware Description Language
VR	Visual Representation
WT	Wavelet Transform

Chapter 1

Introduction

Contents

1.1	Introduction	3
1.2	Digital Imaging Techniques	3
1.2.1	Photography	4
1.2.2	Digital Radiography	4
1.3	Digital Image Processing	5
1.4	Image Analysis	8
1.4.1	Image Segmentation	8
1.4.2	Texture based Object Description . .	10
1.4.3	Texture Analysis	11
1.4.4	Sequency based MRT (SMRT)	18
1.4.5	Feature Selection	21
1.4.6	Classification of Images	23
1.4.7	Analysis of Medical and Non Medical Images	24
1.5	Software Hardware Co-development .	26

1.5.1	Parallel Distributed Architecture . . .	26
1.6	Motivation	27
1.7	Organization	29

1.1 Introduction

Human beings are visual creatures, who can see faster than they can think. They derive most of the information about the world through their visual senses. Natural curiosity of human beings to capture the visuals and to know in depth of the things which they cannot see by normal eyes led to the discovery of imaging. In the context of imaging, images can be classified as analog and digital.

Images captured by human eyes are analog in nature. Analog images are also captured using cameras with photographic films processed in dark rooms. These images have various levels of brightness and colors.

In digital imaging, analog information is captured and converted to digital signals by sensors. Digital image obtained is a two dimensional array of small picture elements, termed as pixels, representing a particular location and value in the form of brightness. The images are classified based on the pixel value information as binary, grayscale, color and multispectral images.

There are different imaging techniques to capture, store, manipulate and display images. The imaging techniques used in the thesis are discussed below.

1.2 Digital Imaging Techniques

Digital imaging was developed to overcome the weaknesses of film cameras in the early seventies. Nowadays it is extended to many fields including, photography, medical imaging, robotic vision, remote sensing etc. Some of the imaging techniques [1], [2], [3] are listed below:

1.2.1 Photography

Digital photography captures scenes with a digital camera. The camera contains an array of electronic sensors, which sense visible light, to capture images focused by the lens. The output of the sensors is digitized and stored in memory which can be used for further processing or viewing. Photography normally captures what we can see with our naked eyes. Images which naked eye cannot see have to be captured for diagnostic purposes using techniques like radiography, magnetic resonance etc.

1.2.2 Digital Radiography

Radiography, which uses x-ray imaging techniques, has been a reliable and versatile technology in imaging which is used for diagnosis as well as for industrial applications. X-rays are electromagnetic waves having wavelength greater than visible light. They are emitted by a cathode ray tube and passed through the object to be imaged and received by a detector containing image sensors. There are different types of radiography used for different purposes such as Fluoroscopy, mammography, computed tomography etc. The output of X-ray imaging, fluoroscopy, mammography all are projection images.

Computed Tomography

In computed tomography (CT) x-ray is combined with computing. It creates an image from projection measurements using the inverse Radon transform. In other words, CT images are not acquired directly from an imaging device but is a slice reconstructed from multiple projections taken from different angles. A CT imaging device aims a narrow beam of x-ray, which is quickly

rotated around the object under inspection. The signals thus produced are captured by sensors and are processed by the computing part of the device to generate cross-sectional images termed as slices. The processing device digitally stacks these slices to create a three-dimensional image of the object's internal parts. CT has medical as well as industrial applications. In industry, CT is used for inspecting the internal parts and components of products.

In medical imaging, CT is used to get detailed images of internal organs, bones, soft tissue and blood vessels for diagnosis and therapeutic purposes. Compared to ordinary x-ray images, CT images avoid overlapping structures making the internal anatomy more clear.

Images captured using various imaging techniques discussed above, has to be processed and analyzed to get the relevant information. Processing and analysis of digital images are termed digital image processing.

1.3 Digital Image Processing

Digital image processing (DIP) is a rapidly growing field having wide applications in almost all fields of science and engineering. It focuses mainly on processing the output of different digital image sensors for human interpretation, storage or autonomous machine perception. Nowadays almost all technical fields use image processing. A few major applications of Digital Image processing [1], [2], [3] are listed below:

- **Communication:** Images and videos used in communication have to be captured, enhanced, compressed and secured before transmitting. Different compression algorithms are used for faster communication. Steganography, watermark-

ing etc. are used for secure information transmission and reception.

- **Industry:** Image processing is used in industry for automatic inspection of items on the production line and to separate different items automatically.
- **Remote sensing:** Remote sensing deals with the acquisition of images without making physical contact. It is used in urban monitoring and planning, precision agriculture, defense and security issues. Spectral imaging is mainly used in remote sensing.
- **Robotic Vision:** Vision in robotics is helpful for its navigation. Also, it can be used in industry for inspection and assembly of parts of equipments. The images are captured using high-end digital cameras or video recorders. The processes involved in robotic vision are sensing, preprocessing, segmentation, description, recognition, interpretation, etc.
- **Medical Imaging:** Medical image processing is a powerful application of DIP which includes image acquisition, computer processing and analysis of medical images. Medical imaging techniques include digital radiography, digital sonography, nuclear imaging, etc.

The major processes used for improving pictorial information for human interpretation includes image restoration, image enhancement and image compression. In autonomous machine perception, the major task is image analysis in which the image processing tools required are image segmentation, object description and representation, pattern classification etc. Important processes involved in DIP [1], [2], [3] used in almost all applications are briefed below:

1. **Image Restoration:** Image restoration, an objective process is the operation which reconstructs or recovers a de-

graded image. Blur, noise, camera misfocus are some causes of degradation. The degradation factor is identified and modeled and the reverse process is applied for restoration.

- 2. Image Enhancement:** Image enhancement is a subjective process that emphasizes the image features such as edges, boundaries or contrast which will make the image more pleasing to the observer. The digital image is modified so that it is more suitable for analysis or display. Spatial domain and frequency domain methods are used for enhancement.

- 3. Image Compression:** Image compression refers to a type of data compression applied to digital images for efficient storage or transmission. Image compression technique removes redundant and irrelevant information encodes the relevant information. Lossy and lossless compression techniques are used. Lossy compression is used where degradation of decompressed image is acceptable, as it offers high compression ratio. Lossless compression techniques are used in case of medical and archival images where the original image has to obtain by decompression.

- 4. Image Registration:** Image registration is the process in which two images having different coordinate system is transformed into a single coordinate system. After registration, it is possible to compare or combine two images.

Image analysis is the major image processing task used in the present work which is introduced in the coming section.

1.4 Image Analysis

Image analysis methods use information extracted from images for understanding, recognizing and differentiating diverse types of images. The output of image analysis is not an image, but numerical values. The block diagram of the image analysis system is shown in Fig. 1.1.

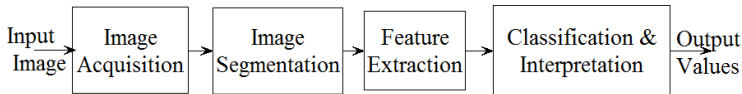


Fig. 1.1: Block diagram of Image Analysis System

After obtaining the images for analysis the first step is to segment the region of interest.

1.4.1 Image Segmentation

Segmentation [1], [2], [3], the first stage in image analysis, is the process of dividing an image into parts called segments. The segmented image is a simplified and meaningful representation of the region of interest (ROI), which can be analyzed easily. Segmentation is applied on almost all types of images.

In medical images, segmentation is done to locate tumors, disease diagnosis, assist surgery etc. In industry, it is used in the automatic traffic control system, biometric identification system etc. for object detection and recognition tasks. Segmentation also used in classification of terrains in satellite images.

There are a number of segmentation methods and the choice of the method is based on the application. Segmentation techniques

are broadly classified into layer based and block based methods. Layer based method is applied for an image, by dividing it into the foreground, mask and background layers. Block based segmentation is done based on various features found in the image. The block based segmentation can be further classified [3], [4] into edge based, region based, thresholding based, clustering based on pixel features etc.

Edge based segmentation relies on the notion that edges often occur at object boundaries. Edges represent the difference in gray level values identified based on feature dissimilarities. In monochrome images, dissimilarities can be identified by taking the first or second derivatives of the images or from the histogram. In edge based segmentation boundary of ROI is identified, where as in region based segmentation ROI as such is identified.

Region based segmentation depends on the similarity between image pixels. There are different segmentation techniques in this category namely Region Growing, Region splitting/merging etc. In region growing algorithm a seed is identified and the neighboring pixels are examined and added to the seed based on similarity. For region splitting and merging many different algorithms are used and one among them is the quad tree algorithm. Quad tree is a data structure used to split two dimensional images by iteratively splitting into four quadrants or regions. Based on similarity between different quadrants they can be merged also. Another segmentation technique used for identifying ROI is based on clustering of image features.

Clustering based segmentation depends on rearranging the image by grouping similar pixels together. Similarity of pixels measured based on texture, color etc. K Means, Fuzzy C means etc. are the commonly used clustering algorithms in image segmentation. Another approach in segmentation is based on deformable models.

Deformable models are used when the ROI has insufficient boundaries or when there is a lack of contrast between the region and background. They are generally classified into parametric and geometric models. Parametric models are also termed as active contours [5] which can be moved to the boundary of ROI under the influence of some internal and external forces. The evolution of the model is based on the energy minimization of the curve due to these two forces. Geometric models, on the other hand, are based on level set based shape representation and curve evolution. Statistics based geometric deformable model termed as active shape model [6], is used in the work.

Once the segmentation is done, the resulting set of pixels have to be represented for classification. Representation can be based on external or internal properties of the segmented pixels. External property is the shape of the segmented region. Internal properties can be color and texture. Based on the suitable representation different object descriptors are defined.

1.4.2 Texture based Object Description

Object Description or feature extraction deals with extracting attributes from images that result in some quantitative information of interest which are used for further analysis. Feature descriptors are a set of values obtained from each image, in order to identify the relationship among a collection of images. The description can be based on a single representation or many taken together. Feature descriptors are used for image registration, image matching and recognition, image classification etc.

The segmented image will have a boundary and the contour representing the boundary is termed its shape. So an object can be described in relation to its shape. The most commonly used boundary descriptors [3] are chain codes and Fourier descriptors.

In chain code, the boundary pixels are represented as a connection between the pixels. In Fourier descriptor, the boundary is represented in terms of frequency of the contour.

A counterpart of boundary descriptor is regional descriptor which characterizes the geometrical properties or density of the region. The descriptors based on geometrical properties are termed as basic regional descriptors and descriptors based on the density are termed as moments. Basic descriptors that characterize the region are area, perimeter, compactness, dispersion (irregularity) etc.

Color is also an important property used for image representation, which is invariant to scale, rotation and translation. The important color descriptors [7], [8], [9] used are color moments, color histograms and color coherence vectors. The mean, variance and standard deviation of the image are collectively termed as color moments. The color histogram represents the distribution of colors in an image. It can be developed for RGB or HSV color space. Color coherence vector (CCV) separates coherent and incoherent pixels with respect to each color. It distinguishes images better compared to the color histogram.

Another commonly used object descriptor is based on texture, known as texture descriptors or texture features. The method to identify this descriptor set is termed texture analysis.

1.4.3 Texture Analysis

Gray image texture can be defined as a function of the spatial variation of pixel intensities. Texture is characterized by a given pixel and the pattern in a local area around the pixel. This can be perceived in two-dimensional images as homogeneous visual patterns which represent the surface composition being imaged.

Hence it plays a very important role in analysis images like remote sensed data, biomedical modalities and natural scenes.

Texture features used for image analysis can be broadly classified into statistical features and transform based features.

Statistical Texture Features Statistical methods is based on spatial distribution analysis of pixels, by computing distribution of some localized features. First order and second order statistics are used for obtaining texture features.

First Order Statistics

First order statistics of image is computed from the histogram of pixel intensities [10], which represents the probability density function of pixels. Features such as mean, variance, skewness, kurtosis, energy, entropy etc. of histogram are used as features. Mean gives the average level of intensity of the texture. Variation intensity around the mean is termed variance, measures the similarity of intensities within a region. Skewness is an indication of symmetry. If skewness is negative, the data are spread more to the left of the mean than to the right. If this is positive, the data are spread out more to the right. Kurtosis is a measure of flatness of histogram. All these first order statistics based features do not provide any information about the relative pixel positions, which describes the texture characteristics. Second order statistics based features overcome these disadvantages.

Second Order Statistics

Second order statistical methods give information about tonal and spatial dependencies of pixels. Most popular second order

statistical methods are based on the gray level co-occurrence matrix (GLCM) and gray level run length (GLRL) matrix.

Gray Level Co-occurrence Matrix (GLCM) [11], [12] is termed as the second order histogram of an image. The elements of the GLCM matrix give the distribution of co-occurrence of pixel values at an offset. The image is scanned in different angles for finding the GLCM matrix. The size of GLCM matrix depends on the gray level and size of the image.

Many times GLCM matrix will be very large owing to its computational cost. Because of its large size, GLCM cannot be used as such as a texture feature. Hence metrics based on GLCM matrices are taken as texture features. Normally texture features based on GLCM matrices are termed as Haralick features, and is given in Appendix A, found to be very efficient for many class of textures.

Gray level run length matrix [13] is another approach to identify spatial dependencies of pixels. Gray level run is defined as a set of consecutive, collinear picture points having the same gray level value. The length of the run is the number of pixels in the run, i.e. it gives the information about number of connected pixels in a run. Similar to GLCM, GLRL matrix size depends on gray level and image size. After computing the gray level run length matrices, the texture features can be calculated similar to GLCM as given in Appendix A.

Transform Based Features Almost all naturally occurring textures exhibit many regularities like approximate periodicity and variation which are difficult to model using traditional statistical techniques. But this can be easily modeled using transform based techniques. Researchers started working on Fourier transform based texture features from very early period.

Fourier Transform

The Fourier transform decomposes a function of time (a signal) into the frequencies that make it up. Literature reveals that Fourier Transform based texture features were proposed from the early days. Weszka *et al.*[14] proposed Fourier power spectrum based features. Fourier transform based texture analysis methods utilizes only the magnitude spectrum and ignores the phase spectrum. But the phase spectrum gives much of the information about the spatial structure of textures. That is why many Fourier transform based methods failed in texture analysis. Later Local Fourier Transform or short time Fourier Transform based features are proposed which performed better because it utilizes the phase information in a local window along with magnitude spectrum.

Gabor Transform

The Gabor transform, named after Dennis Gabor, is a special case of the short-time Fourier transform. It is used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. The function to be transformed was first multiplied by a Gaussian function, which can be regarded as a window function, and the resulting function was then transformed with a Fourier transform to derive the time-frequency analysis. But it is difficult to characterize different scales of same texture using Gabor transform. This drawback was rectified using wavelet transform for texture analysis.

Wavelet Transform

Fourier transform analysis is done by decomposing a signal into its component sine waves, whereas in wavelet transform decompo-

sition, the signal is decomposed into scaled and shifted versions of the mother wavelet. Hence, wavelet transform offers good localization in the time frequency domain. It can be used for texture analysis as it provides both frequency and spatial information. Both pyramid structured algorithm and tree structured algorithm based coefficients were used for obtaining texture features Appendix A. The first order and second order statistical methods explained for pixel values are also applied for wavelet coefficients of pixels to find texture features.

Similar to first order statistical methods many transform based methods does not provide spatial information of pixels. Hence, researchers are exploring on transform based methods that will provide the phase information along with frequency information. Mapped Real transform (MRT) is an emerging transform which can provide spatial as well as frequency information about data.

Mapped Real Transform(MRT)

MRT (Mapped Real Transform, originally M-dimensional Real Transform) is an evolving transform [15] [16] [17] that can be used for the frequency domain analysis of signals. It is evolved from the modification of DFT computation using 2×2 DFT which involves only real additions and exploiting the symmetry and periodicity properties of the twiddle factor. The MRT coefficients $Y_{k_1, k_2}^{(p)}$ for any signal X_{n_1, n_2} , $0 \leq n_1, n_2 \leq N - 1$, are given by

$$Y_{k_1, k_2}^{(p)} = \sum_{\forall(n_1, n_2)|z=p} x_{n_1, n_2} - \sum_{\forall(n_1, n_2)|z=p+M} x_{n_1, n_2} \quad (1.1)$$

where, $0 \leq k_1, k_2 \leq N - 1$ and $0 \leq p \leq M - 1$,

$M = \frac{N}{2}$, $z = ((n_1 k_1 + n_2 k_2))_N$.

MRT has $\frac{N^3}{2}$ coefficients, of which many are redundant and only N^2 coefficients are unique. Methods were proposed to eliminate

the redundant coefficients and retain only the unique MRT coefficients. An algorithm for finding all the MRT coefficients and identifying the unique MRT coefficients by removing redundancy and placing it is explained in [18]. In [17], an algorithm was proposed to identify the unique MRT coefficients and placing it which is termed as UMRT algorithm. In the proposed algorithm, a group of DFT coefficients was identified which uniquely represent the MRT coefficients which are termed as the basic DFT coefficients. These $(3N-2)$ basic DFT coefficients with different p 's are placed in an $N \times N$ matrix. The scheme places these coefficients where it actually duplicates. UMRT algorithm is faster than the earlier algorithm [18] as there is no need to find all the MRT coefficients. The placement of an 8×8 UMRT is shown in Table1.1.

Table 1.1: (k_1, k_2, p) values in placement of 8x8 UMRT Coefficients

0, 0, 0	0, 1, 0	0, 2, 0	0, 1, 1	0, 4, 0	0, 1, 2	0, 2, 2	0, 1, 3
1, 0, 0	1, 1, 0	1, 2, 0	3, 1, 1	1, 4, 0	5, 1, 2	3, 2, 1	7, 1, 3
2, 0, 0	2, 1, 0	2, 2, 0	6, 1, 1	2, 4, 0	2, 1, 2	6, 2, 2	6, 1, 3
1, 0, 1	3, 1, 0	3, 2, 0	1, 1, 1	1, 4, 1	7, 1, 2	1, 2, 1	5, 1, 3
4, 0, 0	4, 1, 0	4, 2, 0	4, 1, 1	4, 4, 0	4, 1, 2	4, 2, 2	4, 1, 3
1, 0, 2	5, 1, 0	1, 2, 2	7, 1, 1	1, 4, 2	1, 1, 2	3, 2, 3	3, 1, 3
2, 0, 2	6, 1, 0	6, 2, 0	2, 1, 1	2, 4, 2	6, 1, 2	2, 2, 2	2, 1, 3
1, 0, 3	7, 1, 0	3, 2, 2	5, 1, 1	1, 4, 3	3, 1, 2	1, 2, 3	1, 1, 3

The visual patterns of UMRT coefficients were analyzed and found that there exists a specific pattern for each k_1, k_2, p , which represents the addition of certain elements and subtraction of certain other elements [19].

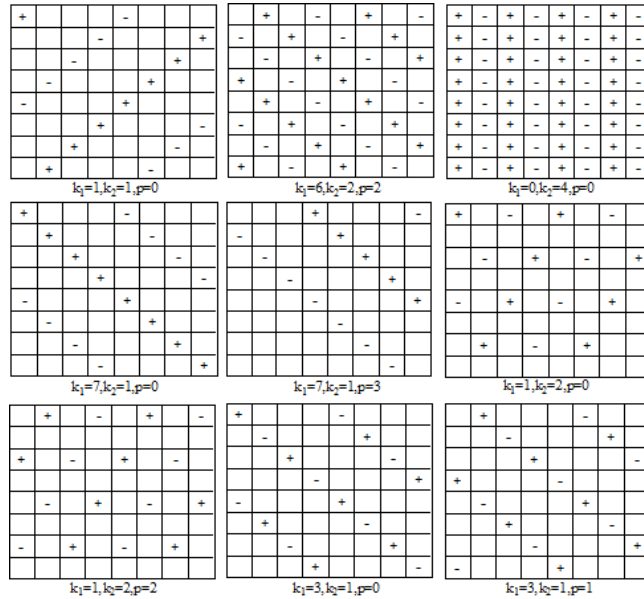


Fig. 1.2: Pixel Pattern of UMRT Coefficients

The +, - and blank symbols in the visual representation indicate that the data in that position is to be added, subtracted or ignored to obtain the MRT coefficients, as shown in Fig. 1.2. From the Fig. 1.2 it is clear that various MRT coefficients measure the gray level differences of the pixels. i.e., coefficients represent texture in terms of local gray level differences on various pixel distances and orientations.

UMRT based texture features were found out from the above observations and the expression for MRT given in Eqn.1.1. A texture feature is formed from the absolute sum of the coefficients corresponding to different p values of a particular k_1, k_2 . A 2-D

UMRT texture feature as in [19] is defined as,

$$f_{k_1, k_2} = \frac{\sum_{i=1}^{N_b} \sum_p |Y_{k_1, k_2}^{(p)}|}{I \times I} \quad (1.2)$$

where $I \times I$ - size of image, $N \times N$ - size of image block and N_b - No. of blocks = $\frac{I^2}{N^2}$. The total number of features for a particular size of image block, N is $3N - 2$. UMRT texture features [19] were found better than GLCM and GLRL based features in image classification.

Visual representation of the DFT coefficients was derived in [15] and was extended to MRT coefficients in [17]. In [20] visual representation of UMRT coefficients was analyzed. Analysis of visual pattern revealed the presence of a constant number of '+' and '-' symbol pairs for each row and column among the UMRT coefficients. The number of '+' and '-' symbol pairs in a spatial direction is defined as sequency, c . When the coefficients are rearranged based on this sequency pattern, Sequency based MRT was formed.

1.4.4 Sequency based MRT (SMRT)

SMRT can be considered as an arrangement of UMRT coefficients according to row-wise and column-wise sequencies. The (k_1, k_2, p) placement of 8×8 SMRT coefficients is shown in Fig. 1.3

000	010	011	012	013	020	022	040
100	110	310	510	710	120	320	140
101	111	311	511	711	121	321	141
102	112	312	512	712	122	322	142
103	113	313	513	713	123	323	143
200	210	211	212	213	220	620	240
202	610	611	612	613	222	622	242
400	410	411	412	413	420	422	440

Fig. 1.3: (k_1, k_2, p) placement of 8×8 SMRT Coefficients

In other words, SMRT can be considered as an ordered arrangement of sequency packets, having elements with same row and column sequencies in each packet as shown in Fig. 1.4 .

0,0	0,1	0,2	0,4
1,0	1,1	1,2	1,4
2,0	2,1	2,2	2,4
4,0	4,1	4,2	4,4

Fig. 1.4: Sequency Packets in 8×8 SMRT Coefficients

Jaya [21] also derived the expression for direct computation of SMRT as explained below:

The SMRT coefficients $S_{c_1, c_2}(i_1, i_2)$ for any data $X = [x(n_1, n_2)]$, where $0 \leq n_1, n_2 \leq N - 1$, $c_1, c_2 = 0, 2^0, 2^1, 2^2, \dots, M$; $i_1 =$

$0, 1, 2, \dots, \frac{M}{c_1} - 1$ and $i_2 = 0, 1, 2, \dots, \frac{M}{c_2} - 1$
 can be represented in terms of a kernel, $A_{c_1, c_2, i_1, i_2}(n_1, n_2)$, as

$$S_{c_1, c_2}(i_1, i_2) = \langle X, A_{c_1, c_2, i_1, i_2} \rangle = \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} x(n_1, n_2) \cdot A_{c_1, c_2, i_1, i_2}(n_1, n_2) \quad (1.3)$$

where for $c_1 \leq c_2$ or $c_2 = 0$,

$$A_{c_1, c_2, i_1, i_2}(n_1, n_2) = \begin{cases} +1, & \text{if } ((n_1 \cdot c_1 \cdot (1 + 2 \cdot i_2) + n_2 \cdot c_2))_N - c_1 \cdot i_1 = 0 \\ -1, & \text{if } ((n_1 \cdot c_1 \cdot (1 + 2 \cdot i_2) + n_2 \cdot c_2))_N - c_1 \cdot i_1 = M \\ 0, & \text{otherwise} \end{cases} \quad (1.4)$$

and for $c_1 > c_2$ or $c_1 = 0$,

$$A_{c_1, c_2, i_1, i_2}(n_1, n_2) = \begin{cases} +1, & \text{if } ((n_1 \cdot c_1 \cdot (1 + 2 \cdot i_1) + n_2 \cdot c_2))_N - c_2 \cdot i_2 = 0 \\ -1, & \text{if } ((n_1 \cdot c_1 \cdot (1 + 2 \cdot i_1) + n_2 \cdot c_2))_N - c_2 \cdot i_2 = M \\ 0, & \text{otherwise} \end{cases} \quad (1.5)$$

$\langle X, A_{c_1, c_2, i_1, i_2} \rangle$ denote the inner product of the two $N \times N$ matrices X and A_{c_1, c_2, i_1, i_2} . Some of the general properties of SMRT were also discussed in [22]. SMRT in its course of evolution were applied in many image processing applications [23], [19], [24] and [21].

Feature set obtained by many feature extraction techniques explained in section 1.4.3 is very large in size and all the elements in the set will not contribute to the classification of images. Optimization is done to minimize the size of feature set giving maximum classification accuracy. There are different feature selection techniques as explained in the following section.

1.4.5 Feature Selection

The process of removing irrelevant or redundant elements, that won't serve the purpose for which the features are extracted, is called feature selection. It is an important task in texture analysis. In classification problems some feature selection techniques depends on the classifier and some are classifier independent.

Fisher criterion and Principal component analysis are two statistical based feature selection methods independent of the classifier termed as filter methods for feature selection. Some methods do up to classification for getting the optimum feature set and such methods are termed wrapper methods. Examples of such methods are recursive feature elimination methods and the optimization techniques like genetic algorithm, particle swam optimization etc.

- 1. Filter Methods:** Linear Discriminant Analysis (LDA) based on Fisher criteria is mainly used to select optimum features based on some ranking, whereas in PCA feature set size is reduced based on eigen values.
- 2. Wrapper Methods:** Forward selection, backward elimination, recursive feature elimination, genetic algorithm (GA) optimization, ant colony optimization (ACO), particle swarm optimization (PSO), simulated annealing (SA) etc. comes under the category of wrapper method for feature selection. Forward selection starts with zero features and adds up elements iteratively in the feature set by checking for classification performance. Backward elimination starts with the complete feature set and iteratively eliminates elements until a stable classification performance is reached. Recursive feature elimination comes in the class of greedy optimization techniques, iteratively eliminates the least performing elements in the feature set and repeatedly changes the ob-

jective function to find the best feature set. Almost all optimization techniques like GA, PSO, ACO etc. can also be used along with the classifier to optimize the feature set.

In this thesis the wrapper feature selection based on GA optimization is used.

GA Optimization

Genetic Algorithms [25], [26] are adaptive heuristic search algorithms evolved by mimicking the same process mother nature uses, inspired by Darwin's theory of evolution - 'Survival of the fittest'. It uses the combination of selection, crossover and mutation to find the fittest. Crossover is the process in genetic evolution which combines parent chromosomes to produce off springs from selected parents. Mutation introduces new characteristics to the population. Solution to a problem is represented as chromosomes comprising of a set of parameters termed as genes. Each chromosome has a fitness score which is evaluated based on a fitness function. A set of chromosomes/individuals is termed as population. Based on the fitness function, certain individuals get selected to form the population of the next generation. After selecting fit individuals, they undergo genetic operations like crossover or recombination, mutation etc. In GA, recombination combines parts of two individual solutions to form new solution. Mutation introduces new information to the population, by changing the parameters.

When the Genetic Algorithm approach is used for feature optimization [27], [28], [29], selection and omission of features are coded as an individual. The fitness of the individual is determined from its ability to have high classification accuracy with a minimum number of features. The optimum feature subset is used for image classification.

1.4.6 Classification of Images

Image classification analyses the properties of feature subset and put the images into different categories. Classification algorithms usually involves training and testing. In the initial training phase, characteristic properties of the feature subset are identified. In the subsequent testing phase, these feature-space partitions are used to classify image features. The classification techniques can be broadly classified into supervised and unsupervised techniques. The performance of the classifier is evaluated using three measures:

Accuracy, which measures the percentage of objects correctly classified in each class.

Error Rate, which measures the percentage of objects incorrectly classified in each class.

Sensitivity, overall percentage of the objects correctly classified.

Different classifiers [30], [31] used in image processing for object identification are:

Minimum distance classifier uses distance functions to measure the similarity between images in the feature space based on class mean vectors. K-Nearest Neighbor (K-NN) classifier is a supervised non parametric classification algorithm based on distance functions.

Bayesian classifier [30], [32] classify images by maximizing the decision function defined based on the Gaussian PDF of the feature set.

Support Vector machines [33] are supervised learning models with associated learning algorithms that analyze data used for classification [34]. SVM is basically a binary classifier usually perform linear classification. But they can perform non linear classifica-

tion using kernel trick, by which inputs can be mapped to high dimensional feature space.

Neural networks [35], [36] are non linear models, mimicking the functions of human brain, that can adapt with input data. Supervised or unsupervised learning techniques are used to train the neural networks to find a functional relationship among image features and their class.

Multilayer neural networks with back propagation training algorithm is the most popular neural network. Radial basis function neural network structure is similar to BPN, but has a nonlinear Gaussian activation function. Probabilistic neural network PNN is mainly used in classification problems, uses a non linear activation function.

Image analysis was performed on different categories of images for a variety of applications.

1.4.7 Analysis of Medical and Non Medical Images

Image analysis is used in both medical and non medical applications.

Non Medical Images

Aerial photographs and satellite images are analyzed to study about man-made objects and natural scenes. Such images are also analyzed for planning agriculture developments. Spectral analysis of satellite images is done to know the mineral potential of a particular area. Texture photographs termed Brodatz images [37] is widely used as a standard image processing dataset. Image analysis have applications in different industries. Multispectral

and hyperspectral images analysis methods were developed for the quality assessment in different food processing industries [38], [39].

Medical Images

Similar to non medical image analysis, in analysis of medical images also different imaging and analysis techniques can be used. Medical problems that can be addressed using image analysis techniques are diagnostics, treatments, assisting surgery etc.

Analysis of Brain MRI/CT images were used in the diagnosis of brain tumor. Abdomen CT images were analyzed in the diagnosis of liver diseases. Trans-rectal ultrasound (TRUS) image-guided biopsy was done to prostate cancer diagnosis. When medical images are used in assisting surgery, it is termed as image-guided surgery. PET/CT image-guided radiotherapy was used to treat prostate, bladder [40], neck or head cancer [41].

The image analysis methods can be extended to many other medical and non medical applications. In medical applications the method can be extended to the early diagnosis of many deadly diseases in an effective and economic way. Similarly, in non medical applications, it can address many socially and economically relevant applications.

The above mentioned applications can be implemented in software or hardware platform. But, out of the different techniques, some may be easier/faster to implement in hardware and some others may be easier/faster to implement in software. Hence, if software and hardware can work together, the applications can be implemented efficiently.

1.5 Software Hardware Co-development

Real time image processing tasks can be made easier and faster by using software hardware co-development. Also, the flexibility of the system increases with this approach. FPGA based hardware design is used in such systems, which makes the system compatible. The difficult task in texture analysis is to obtain the feature set, which can be implemented in hardware. The remaining processing steps can be implemented in software. Thus the whole system can be implemented in an efficient manner. FPGA implementation of the feature set extraction will work faster than its software version. The speed of the system can be improved by choosing a parallel distributed architecture.

1.5.1 Parallel Distributed Architecture

Parallel distributed architecture discusses about the simultaneous operation of multiple processing elements to finish a task as fast as possible. Literature review suggests that many transforms are implemented in parallel distributed architecture. One has to choose an optimum design from different design approaches. Also to get the best performance, it has to be implemented in a custom hardware. ASIC and FPGA are commonly used for such implementations. Nowadays, FPGA is commonly used in custom hardware implementations as it is cheaper and easily available compared to other technologies. Implementation of transforms using parallel distributed architecture enables real time processing. A hierarchical parallel distributed architecture mimicking neocognitron structure was developed for the computation of DFT coefficients [15]. Discrete cosine transform (DCT), Wavelet transform etc. was also implemented in different architecture styles.

1.6 Motivation

Medical image analysis has been a critical area of research nowadays. Discussions with doctors suggest that prostate cancer is a deadly disease which can be cured if diagnosed in an early stage. Since other prostate diseases also have similar symptoms, early stage diagnosis of prostate cancer is difficult. The other common prostate diseases are Benign Prostate Hyperplasia (BPH) and Prostatitis. BPH is a noncancerous enlargement of prostate gland. Prostatitis is an inflammation or infection of the prostate gland. Prostate specific Antigen (PSA) screening and Digital Rectal Examination (DRE) methods are currently available for early stage detection of prostate cancer. PSA screening is not completely reliable since BPH and Prostatitis can also cause an increase in PSA level. Also, a 'normal' PSA does not completely rule out prostate cancer. If PSA screening result is positive, Digital Rectal Examination is done. DRE detects the tumor only when it reaches a volume, suggesting aggressive biological activity, though DRE is inexpensive and less time consuming to get the results.

Generally, imaging techniques such as Magnetic Resonance Imaging (MRI) and Trans Rectal Ultra Sound (TRUS) imaging are suggested only if carcinoma is suspected. Even though TRUS guided biopsy provides correct diagnosis, it is painful and expensive. MRI, being expensive, is done to locate and quantify the carcinoma. CT images are cheaper compared to TRUS and MRI images, but they are used only for prostate cancer treatments. There are other medical problems, like skin cancer whose early detection is very difficult. Cancer occurring on skin can be broadly classified as melanocytic and non-melanocytic. Melanoma is a malignant tumor of melanocytes. The most common non-melanocytic skin cancers are basal cell carcinoma (BCC) and squamous cell carcinoma (SCC). Malignant Melanoma is the

deadliest of all skin cancers and must be diagnosed early for effective treatment. It is very difficult even for experienced doctors to make correct diagnosis seeing the lesions, as appearance of many malignant melanoma lesions are similar to non-malignant melanoma lesions. Dermoscopic images of skin lesions are analyzed to diagnose the different skin diseases. Researchers are trying to find tools which help doctors to diagnose the disease correctly.

Also Mapped Real transform (MRT) has been proved as an efficient tool for texture analysis [19]. In [19], texture analysis of CT images was used to predict the fragmentation of renal stones. Hence it is useful to explore the possibilities of SMRT based texture analysis to develop reliable and economical methods for the diagnosis of deadly disease like prostate cancer and malignant melanoma.

Manual coconut harvesting is a big problem faced by people in Kerala. Nowadays only very few experienced people are available for doing the task. Developing an automation system of coconut harvesting is a must for a region like Kerala, where there are plenty of coconut trees. The first step to be done is to develop an automated coconut harvesting system, is to identify the growth stage of coconut. So there is a definite requirement to perform image analysis to find the coconut maturity.

The above discussed problems motivated to explore the scope of texture analysis of images in different applications.

Another major issue associated with image processing algorithms are the implementation speed. Faster implementations are necessary for real-time applications. Hence methods are to be developed to improve the speed of texture analysis implementations when they are to be used in real time processing.

The fastness requirement of texture analysis system gives the motivation to explore the co-development of software and hard-

ware for such systems. This also motivated to evaluate the scope of parallel distributed architecture for further improvement in speed.

1.7 Organization

A detailed literature review on relevant topics is presented in chapter 2. Various texture analysis methods are reviewed. Image classification methods, texture analysis of medical images etc in literature are discussed. Different parallel architecture algorithms in the literature are reviewed.

The work described in the thesis is presented as two parts. Chapter 3 and 4 emphasizes the software development of texture analysis. Hardware development is discussed in chapter 5 and 6.

Chapter 3 focuses on the optimal SMRT texture feature extraction and comparison with other popular texture features. All the studies in this chapter are done using images from Brodatz data base.

Chapter 4 explains the texture analysis of medical and non medical images using SMRT features. Abdomen CT images are analyzed for the diagnosis of prostate diseases.

The chapter also presents the use of SMRT texture features for social relevant application other than medical field. The application of SMRT texture features to classify different stages of coconut is explained.

Chapter 5 discusses the development of a distributed parallel architecture algorithm for computing 8×8 SMRT coefficients. A general algorithm for $N \times N$ SMRT, N a power 2 is discussed in chapter 6. FPGA implementation of the algorithm for different N

is also discussed. The chapter also discusses a similar algorithm for 1-D SMRT.

Overall summary of significant work and the major conclusions are given in Chapter 7. Important Research contributions and further scope of the work is also discussed in the chapter.

Chapter 2

Literature Survey

Contents

2.1	Introduction	32
2.2	Texture Analysis	32
2.3	Image Segmentation	38
2.4	Feature Selection	41
2.5	Classifiers	43
2.6	Applications of Texture Analysis . . .	47
2.7	Software Hardware Co-development .	50
	2.7.1 Parallel Architectures on FPGA . . .	51
2.8	Conclusion	52

2.1 Introduction

An exhaustive literature survey is conducted in all the related fields before carrying out the work. Statistical and transform based texture analysis methods in the literature is reviewed in detail. A detailed survey is carried out on different image segmentation techniques. Feature selection procedures described in the literature is studied and a detailed literature survey on GA based feature extraction is done. Different classifiers explained in the literature are reviewed. Previous works on texture analysis of medical and non medical images are surveyed. Finally, literature survey on software hardware co-simulation and parallel distributed architectures of transforms are carried out.

2.2 Texture Analysis

Wezcka *et al.* did a comparative study of different texture measures based on Fourier power spectrum and statistical approaches. A theoretical comparison of textural features was explained in [42].

In [10], different texture analysis methods including statistical and transform based methods were discussed.

Texture Analysis based on Statistical methods

Seminal paper on texture analysis was by Haralick. Haralick *et al.*[11] presented a general procedure for extracting textural properties of blocks of image data. They assume that the texture content information in an image is contained in the overall or average spatial relationship which gray tones in image have to one another. This texture content information was specified by

the co-occurrence matrix of relative frequencies, computed as a function of the angular relationship between the neighbouring resolution cells as well as a function of the distance between them. A set of 14 features was extracted from these matrices.

M. M. Galloway [13] proposed a set of texture features based on gray level run length matrices.

In [12] Haralick reviewed various approaches and models, investigators used for texture analysis. He concluded that for micro textures, the statistical approach seems to work well. The histograms of primitive properties and co occurrence of primitive properties were used for macro textures,.

A new approach to textural features based on co-occurrence matrices was explained in [43]. Gelzinis, Verikas and Bacauskine [44] were concerned with an approach to exploit information available from the co-occurrence matrices computed for different distance parameter values. A polynomial of degree n was fitted to each of 14 Haralick's coefficients computed from the average co-occurrence matrices, characterized the tendencies of variation of the coefficients with the variation of the distance parameter value.

The fractal dimension co-occurrence matrix (FDCM) method, incorporated the fractal dimension and the gray level co-occurrence matrix (GLCM) method, was presented for texture classification by Kim *et al.*[45].

Fuan Tsai *et al.*[46] extended GLCM in three dimensional form to analyze hyperspectral image cubes as volumetric data sets.

A fast algorithm for calculating the textural descriptors based on co-occurrence matrices was given in [47].

An approach which used regional entropy measures in the spatial frequency domain for texture discrimination was presented in [48].

In [49], a new feature set from GLRL matrix was introduced. A geometrical approach based on Texture classification was presented by Hamdan and Larson [50].

Epifanio and Ayala [51] proposed a global framework for texture classification based on random closed set theory.

Phillippe Maillard [52] did a comparative study between two new texture analysis methods, variogram and Fourier spectra method and the GLCM method.

A new algorithm for texture analysis was proposed [53] which uses Regularized Simultaneous Autoregressive (RSAR) method, for classification of tumor on the skin.

In [54], a statistical approach based on joint probability distribution of filter responses was developed. This distribution represented the frequency histogram of filter response cluster centers (textons), used as a feature for classification.

In [55], a feature extraction method invariant to rotation and histogram equalization was proposed based on the concept of Advanced Local Binary Patterns (ALBP), reflected the local dominant structural characteristics of different kinds of textures.

In [56], textures were modeled statistically by the full joint PDF of local fractal dimensions.

In [57], Independent Component Analysis (ICA) of textured images were introduced as a technique for creating a new data dependent bank called as basis functions of textured images, used for texture classification.

Transform based Texture Descriptors

In [14], features derived from the Fourier power spectrum was compared with GLCM and GLRL methods. Connors *et al.*[42]

did a theoretical comparison of the three methods explained by Weszka.

Tau-I Hsu, Calway and Roland Wilson [58] presented a frequency domain approach to texture analysis based on Multiresolution Fourier Transform (MFT).

Fourier transform treats the input picture as periodic. As picture is not periodic, the transform is affected by the discontinuities. This causes very high value in the power spectrum. The Fourier features are sensitive to either spatial frequency or orientation only, not both. Therefore Fourier transform is not widely used in texture analysis. Wavelet methods of analysis and representation are having a significant impact on the science of medical imaging and the diagnosis of disease and screening protocols. Because of a powerful underlying mathematical theory, they offer exciting opportunities for the design of new multiresolution image processing algorithms, and novel acquisition methods.

Lonnestad [59] has formulated a set of features for texture classification based on Haar transform. Experiments performed on Brodatz images based on Haar features provided better classification compared to GLCM and GLRL features.

Mokji *et al.*[60] proposed a new GLCM based on Haar Wavelet transform.

T. Chang and C.C.J. Kuo [61],[62] developed a texture classification algorithm based on tree structured wavelet transform as textures were quasi periodic signals with dominant frequencies in the middle frequency channels.

Unser [63] proposed a variation of the discrete wavelet transform for characterizing texture properties. Unlike other wavelet-based approaches, the method was invariant to translations in the input signal.

Vautrot [64] proposed a texture classification method based on

features extracted by one of the three techniques with Gabor filters, dyadic 1D wavelet decomposition or wavelets packets.

Van de Wouwer *et al.*[65] proposed statistics of the wavelet detail coefficients for texture characterization and introduced two feature sets namely, wavelet based histogram signatures and wavelet co-occurrence signatures.

Wang *et al.*[66], [67] proposed a low-level evolutionary feature extraction method for texture classification based on Gabor wavelets and a high-level neural network based pattern recognition.

Porter *et al.*[68] proposed texture analysis scheme based on Gabor Filter, GMRF model and wavelet transform with less number of features and provided high performance rotation invariant classification.

Dani Kramer and Farzin Aghdasi presented [69] a multi scale statistical approach to texture analysis used in classification of micro calcifications in digital mammograms as benign or malignant. In this method a combination of multi scale texture signatures and wavelet based texture signature were used.

Arivazhagan *et al.*[70] described feature extraction methods for characterization and segmentation of texture at multiple scales based on block by block comparison of wavelet co-occurrence features.

K. Mala and Sadasivam [71], [72], [73], [74] proposed a CAD system for the classification of diffused liver CT images using wavelet based texture analysis. Orthogonal Wavelet transform was applied to the liver images to get horizontal, vertical and diagonal details and statistical texture features were extracted from these details.

Hiremath and Shivshankar [75] proposed a texture classification method for digital images based on the co-occurrence features obtained from the two-level wavelet packet decomposition. Since

the most significant information about texture appeared in the high frequency channels which were further decomposed and used for feature extraction. In [76] they proposed a feature extraction algorithm based wavelet decomposition. The features were constructed from the different combination of sub band images.

Z.Z. Wang and J.H. Yong [77] used 2-D wavelet packet transform for texture based classification. The linear regression model was used for the analysis of correlation and extract texture features for classification.

Ramana Reddy et. al. [78] presented a feature extraction method for classification of textures using GMRF model based on linear wavelets. Seven features were extracted based on the least square error estimation method on third order Markhov neighborhood. Rajesh Cherman proposed [18], [16] a new transform for the frequency domain analysis of two dimensional signals in terms of real additions only. The transform named MRT, mapped real transform maps data to the twiddle factor axis. The coefficients were obtained using real addition and carried frequency as well as phase information.

In [18], a comparison was made between MRT and DFT. 1-D MRT was proposed in [79]. Lean MRT an invertible transform used for image representation was proposed in [80]. Lean MRT representation was effectively used to compress images [81]. A pictorial representation of MRT in terms of 2 x 2 data was proposed in [82]. Bhadran [17] proposed a suitable scheme for placement of UMRT coefficients in the $N \times N$ space, N a power of 2. Meenakshy in her Thesis had shown the feasibility of texture descriptors based on 2-D MRT for prediction of kidney stone fragmentation.

Texture is concluded as an effective tool for analysis of images from the survey paper. Research is continuing on both statistical and transform based texture descriptors.

2.3 Image Segmentation

Segmentation is the first step in image analysis.

Vautrot [64] proposed an image segmentation method based on wavelet transforms. The feature extraction schema was performed based on Gabor filters, dyadic 1D wavelet decomposition, wavelets packets. The feature reduction step was performed by Karhunen-Loeve expansion. The clustering was performed with classical Kmeans (Euclidean or Mahalanobis distance) or Fuzzy C-means techniques.

A method for automatic segmentation of computed tomography (CT) head images of patients having spontaneous intra-cerebral brain hemorrhage was presented in [83].

Nawaz [84] used active contour (snake) method for segmentation.

In [85], an algorithm was developed for anatomical structures segmentation from CT of head images based on the combination of intensity rescaling, threshold algorithm, region growing method, fast-marching method (FMM) and mathematical morphology.

Varshney [86] reviewed different methods used for automatic extraction of organ regions from abdominal CT images. The techniques included mathematical morphology, neural networks, model-fitting, levelset methods, thresholding, edge-detection, and knowledge based classification.

Wei *et al.*[87] used gabor wavelet features of textures derived from gabor filters for the segmentation of remotely sensed LIDAR data.

Luo *et al.*[88] introduced an automatic liver parenchyma segmentation algorithm that can delineate liver in abdominal CT images.

The prostate is difficult to distinguish from its surrounding tissues and hence model based segmentation have to be used.

Model based segmentation

Cohen [89] proposed a deformation model termed as active contour which solved many problems encountered by the snake model. In [90], [91] discussed on a deformable model whose iteration is similar to active contour model, but the model deforms based on the class which it represents.

Nicolae Duta and Milan Sonka [92] described an improvement in Active shape model for segmentation Magnetic resonance image of brain. The knowledge based point distribution model was developed.

In [93], a new search method was proposed for the improved the performance of ASM model in [90].

Cootes *et al.* [94], [95] demonstrated an iterative image search method using statistical appearance model used for any class of objects which exhibit shape and texture variability. The method was applied to face and medical image segmentation.

Leventon *et al.* [5] presented an improvement in the geodesic active contour method of medical image segmentation by incorporating shape representation using PCA.

In [96] presented a report on different shape and appearance models and their application in interpreting images.

View based active appearance model represented both shape and texture variability was explained in [96], [6]. The active appearance model discussed was used to track faces of a person through wide angle changes, and also to predict appearance from new viewpoints given a single image.

An algorithm was proposed in [97] for volumetric segmentation, by extending ASM. Based on this kidney was modeled.

Ginneken *et al.* [98] proposed an improvement in ASM segmen-

tation, in which the normalized first order derivative profiles was replaced by selecting optimal local features using sequential feature forward and backward selection. Non linear K-NN search was used instead of iterative search based on Mahalanobis distance.

Three dimensional segmentation method using 3D-ASM for cardiac CT images was explained in [99]. In the work the positions were updated for model generation based on fuzzy-inference method, performed better compared to convolution based methods.

The paper [100] described the modification of shape and appearance model of ASM for 3D segmentation of structures in medical images. The method was applied for segmenting thrombus in abdominal aortic aneurysms.

In [101] an ASM based face segmentation approach was explained. A texture profile was created based Log-gabor wavelet characteristics.

Tian *et al.*[102] presented 3-D volumetric models for 3-D medical image segmentation based on deformable modes and finite element segmentation. The elastic solid like model has the capability to deal with topology changes and incorporate human interactions.

Payel *et al.*[103] proposed an automatic segmentation of prostate based on combination of GA with level set methods.

In [104], [105] Soumya *et al.* proposed prostate segmentation based on active appearance model in TRUS images. The appearance model was based on texture features from approximation coefficients of Haar wavelet transform. A modification based on local phase information of log-Gabor quadrature filter extracted texture was presented in [106].

Akbari *et al.*[107] presented an automatic model based segmenta-

tion of prostate gland in 3-D trus images, based on wavelet based SVMs and shape model.

A fully automated technique based on ASM for lung segmentation in CT image was presented in [108].

Methods to improve the accuracy of ASM based segmentation was discussed in [109]. In the proposed algorithm landmark features were extracted using Haar filters and K-NN algorithm was used instead of iterative search based on Mahalanobis distance.

Sung-Il Joo *et al.*[110] used Active shape model for detecting harmful images in social network services.

2.4 Feature Selection

In texture analysis, the number of features extracted by a particular method is very large in number. So it is necessary to select the relevant features essential for classification. For this purpose, an automated feature selector has to be included in the system. The feature selector algorithm may be implemented using any of the clustering algorithms or optimization algorithms.

Dejong [25] provided a brief overview of genetic algorithms as a key element in learning systems.

Haleh Vafaie and Ibrahim F.Imam [111] presented a comparison between two feature selection methods, the Importance Score (IS) which was based on a greedy-like search and a genetic algorithm-based (GA) method, in order to understand better their strengths and limitations and their area of application. The results of their experiments show a very strong relation between the nature of the data and the behavior of both systems. The Importance Score method was more efficient when dealing with little noise and small number of interacting features, while the genetic al-

gorithms provided a more robust solution at the expense of increased computational effort.

Stolpmann and Dooley [112] described the usage of genetic algorithm feature selectors in texture classification.

F.C.H. Rhee and Y.J.Lee [27] presented an unsupervised feature selection using a fuzzy genetic algorithm. A fuzzy membership function was employed to determine the degree of closeness of pattern pairs and genetic algorithm was applied to find an optimal set of weighting coefficients that minimizes the evaluation index.

In [113], texture features based on first order statistics, gray level difference method, Laws texture energy measures and fractal dimension measurements were computed. When the feature size is greater than a predefined threshold GA optimization was performed to obtain the optimum feature set. Neural network classification was performed.

In [114], Liu *et al.* proposed a feature selection technique named filtered and Supported Sequential Forward Search (FS-SFS) along with SVM classifier. The method reduced the computation time of SVM.

In [115] GA was used to optimize the feature selection in object based classification. The classifier used was Back Propagation Neural network classifier.

Feng Tan *et al.* [29] proposed a genetic algorithm based framework for feature selection that combines different existing feature selection techniques like entropy based feature ranking, T-statistics and SVM-RFE.

In [116] SGLDM features computed from wavelet coefficients of brain MRI images were used as texture features. Optimization of feature set was done using GA and classification was performed using SVM classifier.

Ant colony optimization was used for feature selection [117] in classification of biomedical images. Gabor wavelet was used to obtain the features using SVM classifier.

Genetic algorithm optimization was done to obtain the wavelet feature subset in [118]. The optimized subset was used in classification using SVM and BPN and the classification accuracy improves tremendously.

In [119] GA based feature selection was performed among GLRL, wavelet and SGLDM texture features. Adams *et al.*[120] proposed that genetic algorithm optimized texture features based on local binary patterns served as an excellent tool for personal identification using an individual's periocular skin texture.

Irfan Alam *et al.*[121] proposed an optimization technique for numerical computation of Haralick features.

After optimization of feature set, classifier has to be used. Examples of classifiers are KNN classifier, SVM classifier etc. Classification can also be implemented using Neural Networks.

2.5 Classifiers

Different classifiers based on supervised and non supervised learning techniques were discussed in the literature.

K-NN Classifier

Michael Peterson *et al.*[122] proposed weight and offset optimization for K-NN with varying similarity measures using Genetic algorithm. Meenakshy *et al.*[123] used KNN classifier for classification of renal stones.

SVM Classifier

Different implementations of multi class SVMS are compared in [124]. Mohamed Aly [125] presented a survey on multiclass classifiers like Neural network classifier, Decision tree based classifier, K-NN classifier, Nave Bayes, SVM etc. He also described the conversion methods of binary classifiers to multi class. The basics of both linear and non linear SVMs were explained in [33], [126], [127], [128], [129], [130], and their application in multiclass classification based on kernel method was discussed. In [131], multiclass SVM with Radial Basis Function (RBF) kernel was used in text classification.

Nave Bayes Classifier

Rennie [132] proposed an improvement for Nave Bayes classifier and used it in text classification. In [32], Murphy explained on statistical multiclass classifiers, especially Nave Bayes classifier.

Neural Network Based Classifiers

Neural network based classifiers with different architectures and learning algorithms are reviewed.

In [133], multi resolution image analysis was performed and classification was done using hierarchical neural networks. Also fuzziness was introduced in the classification techniques. In [134] the non parametric characteristics of back propagation neural network classifiers were tested for image classification. Shah *et al.*[135] discussed Back propagation based Neural network classifier for land cover classification using Haralick features. In [136], features for image classification were derived from shape based

features using wavelet transform. Content based image classification was performed using neural network based on back propagation learning algorithm. Wing *et al.*[137] discussed an improved RBF neural network for image classification which minimizes the localized generalization error. Its performance was compared with multi class SVM. Zhihong Liu and Zhizeng Luo [138] discussed a Learning vector quantization (LVQ) NN based classifier for hand motion pattern classification based on EMG signal. The method achieved good recognition accuracy.

In [139] texture images were classified using feed forward neural network classifiers and SVM classifiers.

Six different LVQ algorithm implementations were discussed in [140], for arrhythmia from the Electrocardiogram (ECG) dataset. The performance of the algorithms varied with initialization positions of code book vector. In [141], a sigmoid radial basis function neural network with growing and pruning algorithm was used for the classification of liver diseases. The performance of the network was compared with basic RBF network with growing and pruning algorithm and found to be better. In [142], features used for classification were extracted based on Daubechies 4 wavelet transform and first order color moments. Classifier used was Back propagation neural network with one hidden layer. Aircraft images were classified with 98% accuracy. In [143], back propagation neural network classifier was used to classify images based on color image histogram and mean value. The method was used to identify image Spam in emails. Jiang *et al.*[144] discussed methods to improve the classification accuracy of high resolution remote sensing images using Back propagation neural network. Taskin *et al.*[145] discussed on classification of remote sensed images using multilayer perceptron model trained with back propagation learning algorithm. They suggested a method for improvement of classification accuracy using data refining. In [146], described neural network based classification of images

by fusing input data, which improved the accuracy than with a single feature set.

Donald F. Specht [147] discussed about PNN and polynomial Adaline classifier based on Bayes strategy. He compared the classifiers with BPN classifier. He concluded that PNN is fast and easy compared to other algorithms. In [148] Specht studied about the disadvantages of PNN classifier and suggested modifications to improve the performance of the classifier. He suggested clustering techniques to reduce the neurons in basic PNN.

Spyridinos *et al.*[149] designed a prognostic classification system for predicting the recurrence of urinary bladder cancer. They used four layered probabilistic NN for performing binary classification. In [150], discussed modified PNN classifier for hepatic cancer diagnosis. They compared three different smoothing parameter search approach of modified PNN methodology based on two sided incremental, Monte Carlo and Genetic Algorithm. Mala *et al.*[71] used PNN classifier for classifying diffused liver diseases using CT images based Wavelet texture features.

B.S.Rani [151] discussed the development of an unsupervised pattern classification technique based on the searching capability of Genetic algorithm for automatic clustering of a given set of feature vectors into an appropriate number of clusters using Kohonen Self Organizing Map (SOM) and Adaptive Resonant Theory network (ART).

K. Mala and Sadasivam [71], [72], [73] proposed a CAD system for classifying diffused liver images from CT images using wavelet based texture analysis and Probabilistic Neural Networks (PNN).

Fallahi [152] did a comparison of classifiers SVM, MLPNN and RBFNN for classifying normal and abnormal CT images. Nawaz [65] proposed a classification method for hepatic lesions based on GLCM.

Wang *et al.* [66], [66] proposed a new Gabor wavelet based texture classification scheme based on Fuzzy ART network for high level decision making and recognition.

2.6 Applications of Texture Analysis

Texture analysis of images is performed in medical and non medical applications.

Mitchell *et al.* [153] introduced a maximum minimum measure for texture analysis. Statistical texture analysis techniques were used for analysis of many medical images [154].

Skorton, Collins and others [155] characterized myocardial tissue using gray level run length statistics.

Statistical texture analysis techniques were used to assess the ability of intra vascular ultrasound (IVUS) data, in raw and scan-converted form, to characterize intra coronary thrombus in [156].

Q. Wei and Y. Hu [157] studied the effectiveness of using texture analysis methods such as GLCM and GLRL for identifying fissure regions of lungs from CT images. They used the feed forward back propagation neural network for classification of four types of regions in the lungs such as fissure, air, bronchial and tumor. They have shown GLRL is a better method considering the computation time and classification accuracy.

An efficient, integrated image textural analysis and classification of transrectal prostate ultrasound images into clusters potentially representing cancerous or normal tissue areas were presented in [158].

In [152], computer aided diagnosis was applied to the brain CT image processing. They compared the performance of morpho-

logical operations in extracting three types of features namely, grayscale, symmetry and texture. Nailon *et al.*[40] investigated the classification of bladder, rectum and a clinically relevant region defined as other using computational methods based on statistical and fractal texture analysis. The statistical methods used were first-order (FOS) based on analysis of the image histogram, second-order based on gray-tone spatial dependence matrices (GTSDM) and higher-order based on gray-tone run length matrices (GTRLM).

Al-Kadi [159] presented the potential of fractal analysis of time sequence contrast-enhanced (CE) CT images to differentiate between aggressive and non aggressive malignant lung tumors.

Carl Philips *et al.*[160] analyzed the directional invariance of Co-occurrence matrices for the purpose of reducing their runtime by minimizing the number of directions analyzed without negatively affecting the quality of the texture data extracted.

Kale [161] presented a new computer-aided multispectral image processing method used in three spatial dimensions and one spectral dimension. The method was based on co-occurrence analysis using a 3-D window of observation for automated identification of suspicious lesions.

Prostate Cancer Diagnosis and Treatment

In [162], proposed a combination of expert system with remote sensing to diagnose prostate cancer. It was developed based on the clinical details of the patient.

Mohamed *et al.*[163] presented a texture analysis method based on multi-feature for prostate cancer diagnosis from TRUS images. GLDM and GLDV features were used and the features were selected based on Mutual information feature selection (MIFS)

algorithm. In [164], histological images were analyzed for diagnosis of prostate cancer and gleason grading. Eleni *et al.*[165] proposed GLCM based texture analysis of histopathological images for Gleason grading of prostate cancer. In [166], a method combining image analysis and optimization techniques was proposed to improve the probability of prostate cancer detection. The fact sheet [167] described all the known facts about prostate cancer diagnosis. The booklet [168] has presented the problems and difficulties faced by people suffering from prostate cancer and the treatment methods. Liao *et al.*[169] proposed a method for localization of prostate in CT images to assist image guided radiotherapy.

Skin Cancer Diagnosis

Maryam [170] proposed a method for detection of pigment networks and streaks in dermoscopy images. The structures were analyzed to find any irregularity which predicts malignant melanoma.

Radu *et al.*[171] presented combined texture and fractal analysis for skin cancer diagnosis from dermoscopy images and photographic images. Haralick texture features were used.

Ballerini *et al.*[172] proposed a content based image retrieval system for diagnosis of different skin diseases. Color and texture features extracted from the images were optimized using evolutionary algorithms.

In [173], a skin cancer detection system was presented. The system used dermoscopy images and features were extracted using wavelet transform and the classification was performed with BPN classifier.

Coconut Harvesting Systems

In [174], had discussed on the health of coconut tree climbers of Kerala. Disabilities and fatalities were very common among them. Permanent cosmetic disorders to skin also happens among the coconut tree climbers.

In [175], [176], Rajesh *et al.* discussed a kinect based coconut harvesting system which is capable of climbing trees, cutting down coconuts, cleaning the tree tops and spraying pesticides.

Giby *et al.*[177] proposed a sparse coding based algorithm for identification of coconut among the different parts of coconut tree. The sparse representation based classifier detected the possible coordinates of coconut with 92 % accuracy. But the method has many limitations based on how the image is captured.

In [178] a method was proposed to identify coconut from images using color image processing, filtering and Circular Hough Transform.

Wibowo [179] discussed the implementation of an autonomous coconut harvesting robot which detects the position of coconut using a genetic algorithm based approach.

2.7 Software Hardware Co-development

Wayne H. wolf [180], discussed about the advantageous, problems and challenges in the co-development of an embedded system. He described the relationship between hardware and software architectures, analysis approaches in software-hardware co-design etc. Takalo *et al.*[181] did a detailed survey on the activities involved and the challenges in co-design of hardware and software. In [182], software hardware development of an automated

surveillance system. The system was modeled and simulated in simulink software.

Prasad *et al.*[183] developed morphological image processing blocks in FPGA and had communication with PC through ethernet.

2.7.1 Parallel Architectures on FPGA

In [184], FPGA implementation of Hough transform was presented which used a combination of incremental method with usual Hough transform expression. Fleury *et al.*[185] developed a parallel Karhunen-Loe‘ve transform prototyped in FPGA for remote sensing applications.

The computation of Discrete Fourier Transform (DFT) coefficients was developed as picture patterns, termed visual representation(VR) [15], based on a set of 36 primitive symbols. By doing a detailed analysis of the visual representation of DFT coefficients, parallel distributed architecture for DFT computation was developed [15].

Visual representation for DFT developed in [15] was modified in [82], in terms of 2×2 data, which gives direct relationship between time domain data and the frequency domain representation in terms of pictures. The primitive symbols based on 2×2 data were also used for visual representation of UMRT coefficients. M-spacing based data availability theorem was developed by analyzing the visual representation. Based on the M-spacing data availability theorem parallel distributed architecture for UMRT computation was developed. Maamoun *et al.*[186] had proposed a parallel architecture for high speed computing of discrete wavelet transform.

In [187], parallel architecture for affine transform, used for high speed image processing, was presented. The architecture was

implemented in FPGA.

Lin *et al.*[188] presented the implementation of parallel computing for discrete Gabor transform. Different parallel computing algorithms and their comparison were presented.

A parallel implementation of image processing on compute unified device architecture (CUDA)-accelerated CPU/GPU system is suggested in [189]. They implemented image processing with C programming language.

2.8 Conclusion

The literature survey discussed in the present chapter created the background for the work reported in the thesis. This aspect is demonstrated in the following chapters.

Chapter 3

GA based optimization of SMRT Texture Features using K-NN classifier

Contents

3.1	Introduction	54
3.2	SMRT based Texture Features	54
3.3	GA based Optimization of 8×8 SMRT Texture Features	57
3.3.1	Fitness Function	59
3.3.2	Genetic Operators	59
3.3.3	Simulation Studies	60
3.4	Texture Features based on SMRT and Wavelet Transform: A Comparison . .	63
3.5	Conclusion	67

3.1 Introduction

Texture in gray level images can be regarded as a spatial distribution of intensity variations. Texture feature is a measure of this gray level distribution. A number of statistical and transform based texture features [10] are commonly used. Gray level co-occurrence matrix (GLCM), wavelet transform and SMRT based features are used in the thesis. The features are evaluated in terms of classification accuracies.

3.2 SMRT based Texture Features

The work explained in the chapter is a continuation of work [19] done using UMRT algorithm to extract texture features. In the algorithm [17] corresponding to (k_1, k_2) values of basic DFT coefficients, the UMRT coefficients were placed in different (k_1, k_2, p) positions as shown in Table 1.1. The positions of particular (k_1, k_2) for different p values have to be identified in UMRT matrix and added together to get a particular feature. This requires a good knowledge of the placement algorithm which is complicated.

SMRT placement is simple compared to UMRT placement scheme. Analysis of the SMRT placement scheme given in Table 1.3 shows that particular (k_1, k_2) for different p values are placed in a row or column. This clearly shows that SMRT texture features can be computed using row wise or column wise addition of SMRT matrix elements. The expression for SMRT texture feature is derived from the UMRT texture equation 1.2 and the SMRT equation based on SMRT kernel, as given below:

For $c_1 \leq c_2$ or $c_2 = 0$,

$$f_{i_2}(c_1, c_2) = \frac{\sum_{i=1}^{N_b} \sum_{i_1=1}^{\frac{M}{c_1}} |S_{c_1, c_2}(i_1, i_2)|}{I \times I} \quad (3.1)$$

and for $c_1 > c_2$ or $c_1 = 0$,

$$f_{i_1}(c_1, c_2) = \frac{\sum_{i=1}^{N_b} \sum_{i_2=1}^{\frac{M}{c_2}} |S_{c_1, c_2}(i_1, i_2)|}{I \times I} \quad (3.2)$$

where $S_{c_1, c_2}(i_1, i_2)$ is the SMRT coefficients

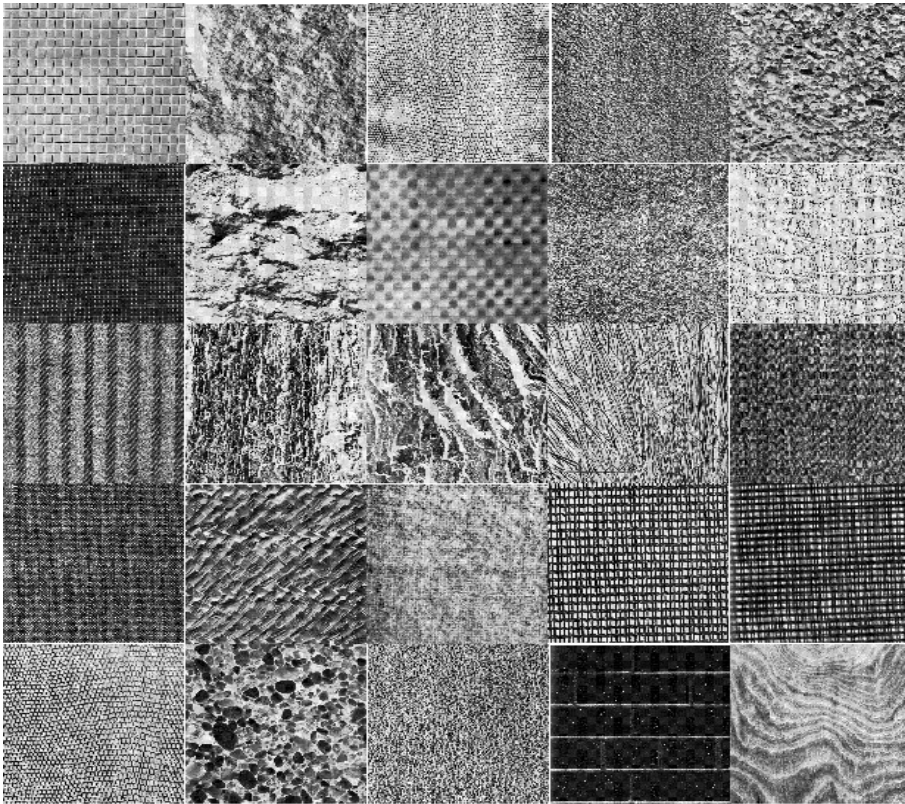
Sequencies, $c_1, c_2 = 0, 2^0, 2^1, 2^2, \dots, M$

Sequency indices, $i_1 = 0, 1, 2, \dots, \frac{M}{c_1} - 1$ and $i_2 = 0, 1, 2, \dots, \frac{M}{c_2} - 1$

The values of UMRT and SMRT texture features are same, only difference is in the computation time.

Comparison of Computation Time for UMRT and SMRT Texture Features

Experiments are performed on 12 images of size 512×512 from Brodatz album [37]. Sample images from brodatz album used in the study are shown in Fig. 3.1. Sub images of size 128×128 are cropped from each image. The sub images are divided into blocks of size, 8×8 . Texture features are found out for the different images based on both UMRT and SMRT algorithms. The features for each sub image is calculated using equation 1.2 based on UMRT and using equations 3.1 & 3.2 based on SMRT. Calculation time for UMRT and SMRT texture feature extraction are derived. The result is verified by changing the sub image size to 256×256 and keeping the block size, 8×8 . The results are tabulated in Table 3.1.



Row 1: D1, D2, D3, D4, D5

Row 2: D6, D7, D8, D9, D10

Row 3: D11, D12, D13, D15, D16

Row 4: D17, D18, D19, D20, D21

Row 5: D22, D23, D24, D25, D71

Fig. 3.1: Sample Images from Brodatz Data Base

Table 3.1: Computation Times for UMRT and SMRT based Texture Features

Image Size =128			Image Size =256		
Images	Time in sec		Images	Time in sec	
	UMRT	SMRT		UMRT	SMRT
D1	75.78	11.37	D1	76.39	11.32
D2	74.95	11.32	D2	76.28	11.48
D3	80.95	11.29	D3	76.2	11.28
D4	74.75	11.47	D4	76.19	11.33
D5	76.11	11.28	D5	75.86	11.33
D6	75.14	11.34	D6	76.3	11.9
D7	76.04	11.31	D7	75.85	11.35
D8	74.29	11.34	D8	76.12	11.3
D9	76.01	11.31	D9	76.2	11.28
D10	74.04	11.35	D10	75.04	11.38
D11	75.43	11.37	D11	74.8	11.39
D12	73	11.33	D12	74.12	11.35
Average time	75.54	11.34	Average time	75.78	11.39

The Table 3.1 clearly shows that SMRT algorithm is approximately six times faster than UMRT algorithm. As SMRT texture feature extraction is found to be simpler and faster compared to UMRT algorithm, now onwards only SMRT is being used to find texture features.

3.3 GA based Optimization of 8×8 SMRT Texture Features

There are $3N - 2 = 22$ texture features for 8×8 SMRT/UMRT as discussed in section 1.4.3 The notations of 8×8 UMRT and

SMRT texture features are as shown in Table 3.2.

Table 3.2: Notations of 8×8 UMRT and SMRT Texture Features

Sl. NO.	$f(k_1, k_2)$	$f_{i_1}(c_1, c_2)/f_{i_2}(c_1, c_2)$
1	$f(0, 0)$	$f_1(0, 0)$
2	$f(0, 1)$	$f_1(0, 1)$
3	$f(1, 0)$	$f_1(1, 0)$
4	$f(0, 2)$	$f_1(0, 2)$
5	$f(2, 0)$	$f_1(2, 0)$
6	$f(0, 4)$	$f_1(0, 4)$
7	$f(4, 0)$	$f_1(4, 0)$
8	$f(1, 1)$	$f_1(1, 1)$
9	$f(3, 1)$	$f_2(1, 1)$
10	$f(5, 1)$	$f_3(1, 1)$
11	$f(7, 1)$	$f_4(1, 1)$
12	$f(1, 2)$	$f_1(1, 2)$
13	$f(2, 1)$	$f_1(2, 1)$
14	$f(3, 2)$	$f_2(1, 2)$
15	$f(6, 1)$	$f_2(2, 1)$
16	$f(1, 4)$	$f_1(1, 4)$
17	$f(4, 1)$	$f_1(4, 1)$
18	$f(2, 2)$	$f_1(2, 2)$
19	$f(6, 2)$	$f_2(2, 2)$
20	$f(2, 4)$	$f_1(2, 4)$
21	$f(4, 2)$	$f_1(4, 2)$
22	$f(4, 4)$	$f_1(4, 4)$

In classification problems, the optimum feature subset has to be found out. In [19], all possible combinations of features were considered to find the feature subsets with highest classification accuracy, in classifying 30 images from Brodatz database [37]. The following four feature subsets with four features each were found out as the best sets that provide 100% classification accu-

racy:

- $G1 - -f(2, 0), f(5, 1), f(7, 1), f(4, 4)$
- $G2 - -f(0, 2), f(2, 0), f(7, 1), f(4, 4)$
- $G3 - -f(1, 1), f(1, 2), f(6, 2), f(4, 4)$
- $G4 - -f(3, 1), f(1, 2), f(6, 2), f(4, 4)$

$G1, G2, G3, G4$ were the four feature subsets for different k_1, k_2 values.

A genetic algorithm based approach is proposed in the chapter, to find out the optimum feature set from the 22 SMRT features instead of taking different combinations.

GA based feature optimization is introduced in section 1.4.5. In each population an individual represents a feature subset and is encoded as an n bit binary vector, where n is the number of features. When a feature is being selected it is represented as bit 1 in the binary vector and represented as bit 0 when the individual is not selected. Decoding can be done by reconstructing a feature subset for the best individual in terms of indices.

3.3.1 Fitness Function

Fitness function is defined based on the objectives of GA optimization. In this work, the objectives are to maximize the classification accuracy of the feature subset and to minimize the number of features selected.

3.3.2 Genetic Operators

Various operations are performed on the individuals in the population iteratively. The fitness function of individuals is evaluated

in each iteration to find the best individual. The genetic operators used are:

Selection: Roulette wheel selection method is used for selection of chromosomes from the present population to add to the next generation. In this selection method, probability that an individual is selected is proportional to its own fitness and is inversely proportional to the fitness of other competing hypotheses in the current population.

Crossover: Traditional Single Point crossover is used. The crossover point n is chosen at random so that child is formed by taking the first n bits from the first parent and remaining bits from the second parent.

Mutation: Mutation operation encourages the search process to unknown regions of search space. It complements a randomly selected bit of feature vector.

3.3.3 Simulation Studies

The classification capability of different SMRT texture feature sets are evaluated by experiments performed on 29 images from Brodatz album [37]. Simulation is performed in MATLAB 7.12 (R2011a) software package. It is done in Intel core i5 machine with 4GB RAM and clock speed 2.4 GHz.

Each image is divided into 16 non overlapping sub images of size 128×128 to form a 29×16 image database. Test dataset is created by extracting a 128×128 sub image from the middle of each image. The sub images are divided into blocks of size 8×8 . SMRT based texture features are calculated using equations 3.1 & 3.2. Parameter settings of GA to find the optimum SMRT features are:

Population size: 1000

Maximum No. of Generations: 50

Selection: Roulette Wheel

Probability of Cross Over: 1

Probability of Mutation: 0.001

The feature set obtained by GA optimization with 100 % classification accuracy has five features, named as Ga is given below:

$Ga = f(0, 2), f(2, 0), f(3, 1), f(7, 1), f(4, 4)$

Performance Evaluation of Different 8×8 SMRT texture feature sets

The performance of different SMRT features is evaluated with the same 29 images from Brodatz album is used. Feature sets used for evaluation are $G1, G2, G3, G4$ (section 3.3) and Ga . From the 512×512 images, 128×128 overlapping sub images with different percentages of overlap is taken to create the data set. When overlap percentage is zero, it means no columns overlap. When overlap percentage is 25, it means that 25% columns overlap, ie., 32 columns of pixels overlap and so on. The test set is created by taking test samples from different portions of the image.

Table 3.3 shows the classification accuracy for different test samples from different regions and training samples with different degrees of overlap.

The results of evaluation are concluded in Table 3.4 which shows that GA optimized feature sets have higher classification accuracy compared to other feature sets. It has higher classification accuracy for 15 images. The feature with next higher classification accuracy is $G2$ with 9 images. Actually, Ga is a modification of $G2$ with an addition of one feature. The addition of that one feature causes improvement in classification accuracy.

Table 3.3: Performance Evaluation of 8×8 SMRT based Texture features

Overlap %	Test Samples	G1(%)	G2(%)	G3(%)	G4(%)	Ga(%)
0%	Middle	100	100	100	100	100
0%	top right	82.76	89.66	79.31	75.86	93.1
0%	top left	86.21	82.76	72.41	75.86	82.76
0%	bottom right	89.66	93.1	68.97	79.31	89.66
0%	bottom left	79.31	82.76	72.41	89.66	82.76
0%	random	100	96.55	89.66	93.1	96.55
25%	middle	96.55	100	93.1	96.55	100
25%	top right	82.76	93.1	82.76	82.76	100
25%	top left	82.76	82.76	75.86	75.86	82.76
25%	bottom right	89.66	93.1	79.31	79.31	93.1
25%	bottom left	75.86	82.76	82.76	82.76	86.21
25%	random	96.55	100	89.66	93.1	100
50%	middle	100	100	93.1	89.66	100
50%	top right	89.66	93.1	86.21	72.41	96.55
50%	top left	82.76	79.31	72.41	82.76	86.21
50%	bottom right	89.66	89.66	82.76	79.31	89.66
50%	bottom left	89.66	93.1	82.76	89.66	96.55
50%	random	93.1	96.55	93.1	96.55	100
75%	middle	96.55	100	93.1	100	100
75%	top right	89.66	96.55	93.1	75.86	96.55
75%	top left	86.21	86.21	86.21	82.76	96.55
75%	bottom right	93.1	93.1	75.86	82.76	100
75%	bottom left	93.1	93.1	96.55	86.21	96.55
75%	random	96.55	96.55	96.55	93.1	100

Table 3.4: Comparison of Performance of 8x8 SMRT based Texture Descriptors

Performance Measures	G1	G2	G3	G4	Ga
No. of times 100% class accuracy occurs	3	5	1	1	9
No. of times highest class accuracy occurs	4	9	2	2	15
No. of times second high class accuracy occurs	8	7	3	2	6

The SMRT texture features classify Brodatz images with 100 % accuracy. In the next section, SMRT texture features are compared with the wavelet transform based texture features, the most popular among transform based texture features.

3.4 Texture Features based on SMRT and Wavelet Transform: A Comparison

SMRT texture features are compared with wavelet texture features based on classification accuracy. The sub image size for both wavelet transform and SMRT based texture analysis is chosen as 16.

SMRT based Texture Features

When the sub image size is 16, there will be 46 different texture features, as explained in section 1.4.3. GA optimization

explained as in section 1.4.5, is performed to obtain feature set with minimum number of features having maximum classification accuracy.

Wavelet Transform based Texture Features

Wavelet based texture features are explained in Appendix A, as described in [62]. The texture image is decomposed using tree structured wavelet transform and energy signature of each sub image is calculated using equation A.5, to obtain the energy map. Dominant frequency channels with large energy value serve as a good candidate for texture representation.

Simulation Study

The classification capability of SMRT and wavelet based texture features are studied based on the experiments performed on 29 images from Brodatz album [37]. Each image of size 512×512 was used. The image is divided into 16 non overlapping sub images of size 128×128 forming a 29×16 image database. Test dataset is created by extracting a 128×128 sub image from the center of each image. SMRT based texture features are computed using equations 3.1 & 3.2, by dividing the image into blocks of size 16×16 , results in 46 features. GA is applied to obtain an optimum SMRT feature sub set. The values corresponding to the features in the optimized feature set are $f_1(2, 0)$, $f_1(0, 8)$ and $f_1(1, 2)$.

Tree structured wavelet decomposition is applied to obtain a sub image of size 16×16 , in three levels and the corresponding energy map of the image is computed. The energy map of dominant frequency channels is taken as feature set. The classification accuracy with different number of dominant channels is obtained.

The classification results of test data with K-NN classifier for SMRT (GA optimized feature set) and wavelet transform based texture features (three top dominant frequency channels) is given in Table 3.5

Table 3.5: Image Classification using SMRT(3 features) and Wavelet(3 features) Texture Features and K of K-NN = 1

Images	Actual Class	Test Result		Images	Actual Class	Test Result	
		SMRT	Wavelet			SMRT	Wavelet
D1	1	1	3	D17	16	16	16
D2	2	2	2	D18	17	17	9
D3	3	3	3	D19	18	18	1
D4	4	4	4	D20	19	3	19
D5	5	5	5	D21	20	20	20
D6	6	6	6	D22	21	21	21
D7	7	7	21	D23	22	22	13
D8	8	24	25	D24	23	23	23
D9	9	9	9	D25	24	24	24
D10	10	10	13	D71	25	25	25
D11	11	11	11	D72	26	26	18
D12	12	12	12	D73	27	17	27
D13	2	9	27	D74	28	28	28
D15	14	14	14	D75	29	29	29
D16	15	15	27	Total Images	29	25	19

SMRT based texture features gives high classification accuracy of 86.2% compared with 65.5% classification accuracy of wavelet based texture feature set.

The results of comparison of K-NN classification using SMRT (GA optimized feature set) and Wavelet texture features with varying number of dominant frequency channels for different K in K-NN classification are given in Table 3.6.

Table 3.6: Comparison of Classification Accuracy of SMRT and Wavelet Texture features with different K values of K-NN classifier

Method	No. of Features	K=1	K=2	K=3	K=4	K=5
SMRT	3	86.2	86.2	96.55	96.55	100
Wavelet	3	65.5	68.97	68.97	72.41	72.41
Wavelet	4	68.97	68.97	72.41	75.86	75.86
Wavelet	5	75.86	82.76	79.31	75.86	79.31
Wavelet	6	75.86	75.86	82.76	79.31	79.31
Wavelet	7	82.76	86.2	86.2	79.31	82.76
Wavelet	8	82.76	82.76	82.76	86.2	82.76

Highest classification accuracies are obtained with SMRT texture features. It is clear that 8 wavelet texture features show a classification accuracy of 82.76% only, whereas with 3 SMRT texture features classification accuracy of 100% is obtained with 5 nearest neighbors.

The inherent gray level difference patterns present in the visual representation of SMRT coefficients enable them as a strong candidate for texture analysis. The studies show that only three SMRT texture features are sufficient to get 100% correct classification compared to wavelet based features, where even with 8 features only 82.76% correct classification is achieved. Wang [17] proposed a progressive texture classification algorithm which is computationally complex to improve the classification accuracy of wavelet texture features. SMRT texture features give 100% classification accuracy with fixed number of three features, which makes computation simple. Thus the experimental results prove that SMRT based texture features are very powerful in classifying texture images compared to Wavelet based texture features in both classification efficiency and computational simplicity.

3.5 Conclusion

Visual representation of SMRT coefficients shows that it measures the spatial distribution of gray level intensities. Texture features are computed by the addition of SMRT/UMRT coefficients corresponding to a particular frequency with different phase values. Hence, feature computation is easier and faster for SMRT than for UMRT. GA based optimization of SMRT texture features using K-NN classifier gives good accuracy for image classification and are compared with Wavelet texture features, suggested by Chang [62]. Top energy signatures computed for tree structured wavelet transform are used as wavelet texture features. SMRT texture features give a classification accuracy of 100 % compared with that of 82.76 % for wavelet features using images from Brodatz database [37]. This proves the usefulness of SMRT texture features and can be applied for the classification of different categories of images.

Chapter **4**

Texture Analysis of Medical and Non Medical Images

Contents

4.1	Introduction	69
4.2	Prostate Disease Diagnosis	69
4.2.1	Abdomen CT Images	70
4.2.2	Segmentation of Prostate Gland	70
4.2.3	Optimization of Texture Features	77
4.2.4	Selection of Classifiers	82
4.3	Skin Cancer Detection	84
4.4	Coconut growth stage identification	88
4.5	Conclusion	94

4.1 Introduction

SMRT texture features optimized using GA and K-NN classifier for Brodatz images gave 100 % accuracy, as discussed in chapter 3. Hence, it is proposed to extend the method for the classification of medical and non medical images. As discussed in section 1.6, early stage diagnosis of prostate cancer is very difficult. If the disease can be diagnosed from images at an early stage, it will be very helpful for the experts as well as for the patients. In this chapter, the possibility of SMRT based texture analysis is explored for the diagnosis of prostate diseases from abdomen CT images. Another category of medical image used for study is dermoscopic skin images to detect malignant melanoma.

The non medical images chosen for the study are photographic images of coconut, to find the growth stages.

4.2 Prostate Disease Diagnosis

A non-invasive technique to identify prostate diseases from abdomen CT image slices using transform based texture analysis is presented in this section. Fig. 4.1 shows the scheme for prostate disease diagnosis.

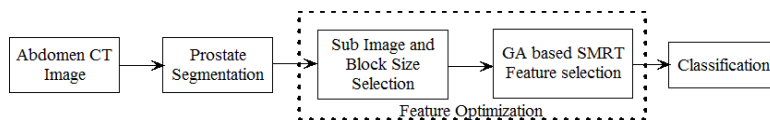


Fig. 4.1: Proposed Scheme

4.2.1 Abdomen CT Images

The abdomen CT scans and supporting clinical details of patients, validated by urologist, are collected in the initial phase of the research work. Data include prostate conditions such as normal, enlarged, local carcinoma and hard. Proper image slices with prostate gland are selected from the CT scan using DICOM viewer. The details of image slices obtained are shown in Table 4.1.

Table 4.1: Details of Image Slices

Sl. No.	Prostate Condition	No. of images
1	Normal	96
2	Enlarged	86
3	Local Carcinoma	48
4	Hard	56
5	Total Number	286

The 10 samples of abdomen CT image slices collected are given in Fig. 4.2.

Next step in the scheme of diagnosis (Fig. 4.1) is to segment the prostate gland from the abdomen CT image. The prostate is first located in the image using segmentation algorithm and cropped into an $N \times N$ matrix, where N is a power of 2.

4.2.2 Segmentation of Prostate Gland

Segmentation of prostate gland from CT image demands medical knowledge about the shape and its position. In the initial phase of work, support from urologist is utilized for its successful identification. Prostate located in the abdomen CT image is shown in Fig. 4.3 with a red boundary.



Fig. 4.2: Samples of Abdomen CT Image Slices



Fig. 4.3: Abdomen CT Image

It is difficult to distinguish prostate tissue from the surrounding tissues due to low contrast as seen from the image in Fig. 4.3. Hence, the segmentation of prostate gland is difficult. A simple edge based segmentation using SMRT is tried to extract the required region of interest.

Edge Based Segmentation

A new edge based segmentation algorithm using SMRT [24] is tried which uses the quad tree concept. This technique is normally applied to images having region to be segmented with some homogeneous property.

Fig. 4.4 shows the result of edge based segmentation. The result shows that edges are clearly identified by the algorithm, but it is not effective for prostate segmentation, since prostate does not have proper edge in abdomen CT images. Hence other methods are tried for prostate gland segmentation and finally ASM method is used as discussed below.



Fig. 4.4: Result of Edge Based Segmentation

ASM Segmentation

Prostate gland is walnut shaped whose size, location and shape varies slightly from person to person and slice to slice. Hence, shape model based segmentation algorithm, suggested by J.F. Cootes [90], is performed to capture the shape variability. The algorithm has two parts. In the first part, Point Distribution Model (PDM) and gray level appearance model are generated from a set of training images. In the latter part, PDM and appearance model are used for segmentation of ROI in the test images.

The shape variability is captured by the PDM developed as explained below:

Landmark points are marked manually in the image. The points should represent the boundary of prostate gland. The number and the way in which landmarks are identified should be same in all training images, Fig. 4.5.

The n land mark points of a single image (x_i, y_i) is represented

as a shape vector \mathbf{x} . For S training images, there will be S shape vectors and all the shape vectors are aligned to ensure that all shapes are in the same coordinate frame. The aligned shape vectors form an n dimensional distribution which is modeled using Principal Component Analysis (PCA), to develop the PDM.

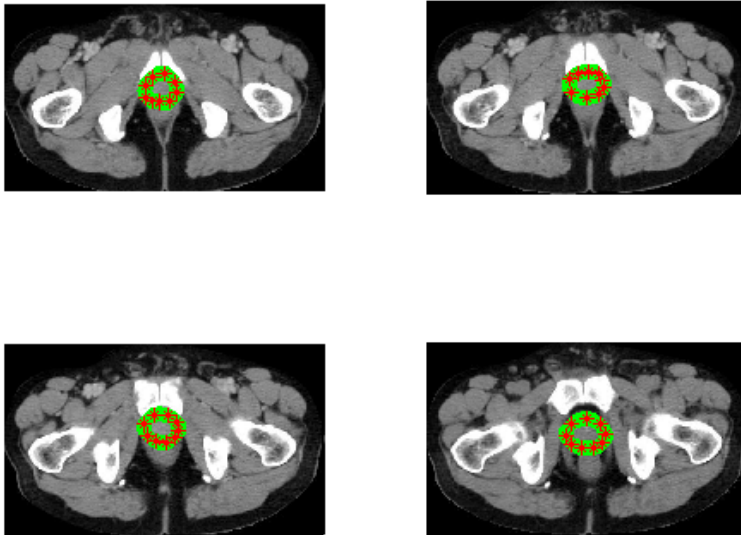


Fig. 4.5: Land mark points in training images

Next step is to develop gray level appearance model so that the gray level information about the training set can be used to identify the desired movements of landmark points in the test image. Normalized first derivative \mathbf{g} of the gray level profiles around each landmark, sampled perpendicular to the contour, as shown

in Fig. 4.6, is used to create the gray level appearance model. The covariance matrix \mathbf{S}_g is also computed.

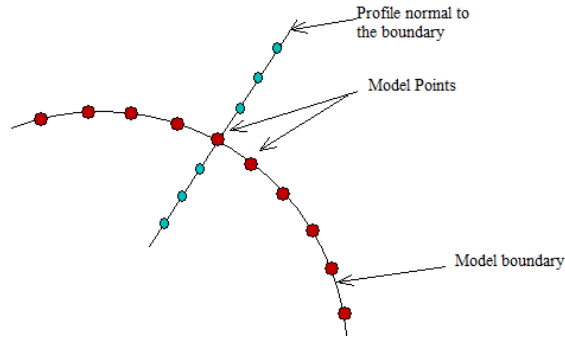


Fig. 4.6: Gray level Profile perpendicular to contour

The iterations in the test image based on PDM and gray level appearance model are explained below:

Shapes are fitted in iterative manner, in the test image, starting from shape model. The movement of landmark points is along normal, to n_s positions on either side (Fig. 4.7). New gray level profile \mathbf{g}_i is computed at each position of landmark points. Mahalanobis (MB) distance $f(g_i)$, between new profile \mathbf{g}_i and gray level model \mathbf{g} , is calculated as $f(g_i) = \mathbf{g}_i - \mathbf{g}\bar{\mathbf{S}}_g^{-1} \mathbf{g}_i - \bar{\mathbf{g}}^T$. All landmarks are put at lowest MB distance and shape model is fitted to displaced points.

A multi resolution approach is implemented to make the search faster by generating a pyramid of image with different resolutions. The base of the pyramid will have the original image and a lower resolution image will be placed at higher locations. So search will start from the top of the pyramid and gradually moves down.



Fig. 4.7: Iterations

The new shape model is obtained after the iterations and hence segment mask is generated as shown in Fig. (4.8). The length and breadth of segmented images are in the range 35 to 42. Hence, a 32×32 image is cropped out to extract the SMRT texture features from the isolated image.

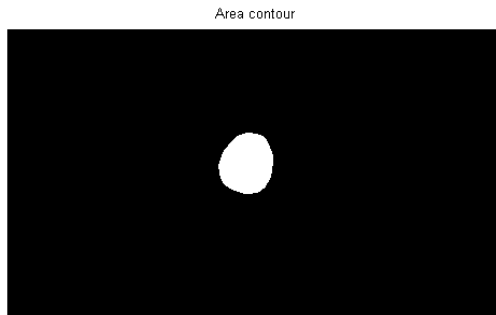


Fig. 4.8: Segmentation Mask obtained

The next step in the development of algorithm for prostate disease diagnosis is to optimize texture features.

4.2.3 Optimization of Texture Features

The first step in optimization is the selection of sub image and block size to compute the SMRT based texture features, as given by expressions 3.1 & 3.2 in section 3.2.

Sub image and Block Size Selection

Sub image size (I) and block size (N) in equations 3.1 & 3.2 are chosen using 32×32 images obtained from the segmented images. Different block sizes are considered to include all possible texture variations in the feature set corresponding to a sub image. The sub image size (I) chosen in this experiment are 8, 16 and 32. Each sub image is divided into blocks of size I, I/2 and I/4, with a minimum size of 8. Classification using K-NN classifier is done with 286 image data, divided into 206 training images and 80 test images, with 20 images from each class. The images are classified into four categories, namely, Normal Prostate, Enlarged prostate (BPH), Local Carcinoma prostate and Hard Prostate (Fully malignant) designated as class A, B, C and D respectively. Table 4.2 shows the results of the experimental study.

Table 4.2: Classification Accuracy of Feature Sets with different I and N

Sl. No.	Sub Image Size, I	Block size, N	Correctly classified (Out of 80)	Classification Accuracy(%)
1	8	8	32	40
2	16	8	28	35
3	32	8	24	30
4	16	16	42	52.5
5	32	16	38	47.5
6	32	32	58	72.5

Feature set obtained by considering 32×32 sub image taken as single block gives the highest classification accuracy and is chosen as the optimum feature set. Classification details of the test images using optimum feature set are given in Table 4.3.

Table 4.3: Confusion Table for Classes A, B, C and D

	A	B	C	D	Classification Accuracy(%)
A	16	4	0	0	80
B	6	12	2	0	60
C	0	2	12	6	60
D	0	0	2	18	90
					72.5

The confusion table, of classification into A, B, C and D classes, presented in Table 4.3, shows that 16 out of 20 samples are correctly classified as class A (normal) and 4 are misclassified as class B (Enlarged). Misclassified 4 samples may be in the boundary between normal and enlarged. The other classes also show similar results as illustrated in the table. The results presented shows potential in classifying the hard prostate with 90 % accuracy (class D). The malignant tissues, class C and D together, are classified with 75 % accuracy.

A 32×32 image block produces 94 elements in the SMRT texture feature set, of which many are not required for classification. Hence a GA based feature selection procedure is presented in the following section.

GA based optimization of Texture Features

GA based optimization of 8×8 SMRT texture features used in Brodatz images, discussed in section 3.3, is extended to opti-

mize 32×32 SMRT texture features for prostate images. The optimization process produced three sets of features (G1, G2 & G3), each with 21 elements and 95% classification accuracy are presented in Table 4.4.

Table 4.4: SMRT Texture Feature Sets G1, G2 and G3

	G1	G2	G3
Sl. No.	$f_{i_1}(c_1, c_2)$ or $f_{i_2}(c_1, c_2)$	$f_{i_1}(c_1, c_2)$ or $f_{i_2}(c_1, c_2)$	$f_{i_1}(c_1, c_2)$ or $f_{i_2}(c_1, c_2)$
1	$f_1(0, 0)$	$f_1(1, 1)$	$f_1(0, 0)$
2	$f_1(0, 2)$	$f_2(1, 1)$	$f_1(1, 0)$
3	$f_5(1, 1)$	$f_6(1, 1)$	$f_1(2, 0)$
4	$f_6(1, 1)$	$f_2(1, 2)$	$f_2(1, 1)$
5	$f_9(1, 1)$	$f_3(1, 2)$	$f_3(1, 1)$
6	$f_{10}(1, 1)$	$f_5(1, 2)$	$f_{10}(1, 1)$
7	$f_{11}(1, 1)$	$f_5(2, 1)$	$f_{11}(1, 1)$
8	$f_4(1, 2)$	$f_6(2, 1)$	$f_{14}(1, 1)$
9	$f_5(1, 2)$	$f_1(4, 1)$	$f_{15}(1, 1)$
10	$f_1(2, 1)$	$f_2(4, 1)$	$f_5(1, 2)$
11	$f_2(2, 1)$	$f_3(4, 1)$	$f_8(1, 2)$
12	$f_3(2, 1)$	$f_4(1, 4)$	$f_1(2, 1)$
13	$f_7(2, 1)$	$f_1(8, 1)$	$f_2(2, 1)$
14	$f_8(2, 1)$	$f_2(1, 8)$	$f_6(2, 1)$
15	$f_4(4, 1)$	$f_2(2, 2)$	$f_3(1, 4)$
16	$f_2(1, 8)$	$f_7(2, 2)$	$f_4(1, 4)$
17	$f_2(8, 1)$	$f_1(2, 4)$	$f_1(4, 1)$
18	$f_2(4, 2)$	$f_2(4, 2)$	$f_2(8, 1)$
19	$f_1(4, 4)$	$f_1(8, 2)$	$f_1(2, 2)$
20	$f_4(4, 4)$	$f_2(2, 8)$	$f_8(2, 2)$
21	$f_2(4, 8)$	$f_2(4, 8)$	$f_3(4, 4)$

The performance of the features are evaluated using the confusion matrix for each feature set as shown in Tables 4.5, 4.6 & 4.7.

Table 4.5: Confusion Table for G1

	A	B	C	D	Classification Accuracy(%)
A	19	1	0	0	95
B	1	19	0	0	95
C	0	1	18	1	90
D	0	0	0	20	100
					95

Table 4.6: Confusion Table for G2

	A	B	C	D	Classification Accuracy(%)
A	18	1	1	0	90
B	1	18	1	0	90
C	0	0	20	0	100
D	0	0	0	20	100
					95

Table 4.7: Confusion Table for G3

	A	B	C	D	Classification Accuracy(%)
A	20	0	0	0	100
B	1	19	0	0	95
C	0	2	18	0	90
D	1	0	0	19	95
					95

The tables show that G1, G2 and G3 feature sets classify malignant images (class C and D together) with classification accuracy

of 95%, 100% and 92.5 % respectively. Thus G2 is the best feature set for identification of prostate cancer.

Next section presents the performance comparison of the set G2 with that of GLCM features, introduced in section 1.4.3.

Comparison of GA optimized SMRT Texture Features with GLCM features

SMRT and GLCM features are compared based on the classification accuracy of prostate images. The images are classified as cancerous and non cancerous. Optimized SMRT feature set, G2, with 21 features are used for classification. In GLCM based classification, 14 features discussed in section 1.4.3 are used. The Table 4.8 gives the result of K-NN classifier using SMRT (G2) and GLCM features. From the table it is clear that SMRT features classify cancerous images with 100% accuracy.

The computations of SMRT features (section 3.2) are very simple and less complex compared to that of GLCM features(Appendix A).

Table 4.8: Comparison Results of SMRT and GLCM texture features

Image Class	Total No. of Images	GLCM		SMRT	
		Correctly Classified	Mis-classified	Correctly Classified	Mis-classified
Non cancer	40	22	6	38	0
Cancer	40	34	18	40	2

The results show that the performance of SMRT texture feature set G2 with 21 features is superior to GLCM texture features in both accuracy and simplicity.

Hence, G2 is considered as the optimum feature set for prostate images obtained from SMRT coefficients under GA optimization procedure.

The above optimization procedure is based on K-NN classifier. The process of prostate disease diagnosis can be made intelligent through the use of learning based classifier. The next step in the prostate disease diagnosis (Fig. 4.1) is the selection of suitable classifier.

4.2.4 Selection of Classifiers

Different types of classifiers are explained in section 1.4.6. The feasibility of PNN, SVM and BPN as classifier to classify the prostate images into four different classes are explored in this section. The selection of the classifier is made based on accuracy, sensitivity and specificity. The images are classified into four classes A, B, C and D as explained in section 4.2.3. The same set of images, as discussed in section 4.2.3, are used to select the classifier. The classification procedure is repeated to include all samples in the test set by changing the training and test sets. The confusion tables for the three different classifiers are as given in Tables 4.9, 4.10 and 4.11.

Table 4.9: Confusion Table for PNN classifier

	A (96)	B (86)	C (48)	D (56)	Classification Accuracy(%)
A	92	3	1	0	95.83
B	3	80	3	0	93.02
C	0	0	46	2	95.83
D	0	0	2	54	96.43
					95.28

Table 4.10: Confusion Table for SVM classifier

	A (96)	B (86)	C (48)	D (56)	Classification Accuracy(%)
A	84	6	5	1	87.5
B	13	71	2	0	82.56
C	0	2	39	7	81.25
D	0	1	4	51	87.93
					84.81

Table 4.11: Confusion Table for BPN classifier

	A (96)	B (86)	C (48)	D (56)	Classification Accuracy(%)
A	67	18	8	3	69.79
B	19	55	8	4	63.95
C	0	2	45	1	93.75
D	2	3	4	47	83.93
					77.86

The accuracy of the classifiers is given in the confusion tables. The correct rate, sensitivity and specificity of the three classifiers are given in the Table 4.12.

SL. No.	Property	PNN	SVM	BPNN
1.	Correct Rate	94.44	85.09	74.3
2.	Sensitivity	95.83	87.5	69.79
3.	Specificity	98.43	84.81	89.06

Table 4.12: Classification of Prostate Diseases using different classifiers

PNN classifier gives very good classification performance for prostate disease diagnosis using CT images. The various prostate diseases

are diagnosed with 95.28 % accuracy. GA optimised SMRT texture features using PNN classifier is found to be a very good tool for prostate disease diagnosis. Hence, the method can be extended to other category of images to address different issues. Another type of medical image chosen for analysis are dermoscopic images of skin, explained in the next section.

4.3 Skin Cancer Detection

Now a days skin diseases are very common across the world. Many of the skin diseases become dangerous if not treated at an earlier stage. Occurrence of skin cancers are also increasing.

It is necessary to have a basic idea about the structure of skin, largest human organ, for the study about skin diseases. The skin has mainly three layers: epidermis, dermis and hypodermis. In the epidermis, there are cells called melanocytes which are responsible for the skin's pigmentation as shown in Fig. 4.9.

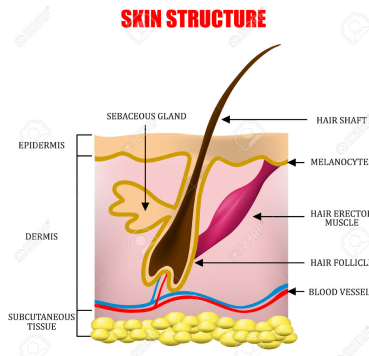


Fig. 4.9: Structure of Skin

Cancer occurring on skin can be broadly classified as melanocytic and non-melanocytic. Melanoma is a malignant tumor of melanocytes. The most common non-melanocytic skin cancers are basal cell carcinoma (BCC) and squamous cell carcinoma (SCC). Malignant Melanoma is the deadliest of all skin cancers and it must be diagnosed early for effective treatment.

Dermoscopic images of skin lesions are analyzed to diagnose the different skin diseases shown in Fig. 4.10.

In the present work, dermoscopic skin images are obtained from an online data base [www. dermnet. com]. 36 Malignant Melanoma images named as class A and 36 other benign melanoma images named class B, are downloaded from the website. Sub images of size 32×32 , 16×16 and 8×8 are cropped manually from the affected area seen in the images. Then texture features based on SMRT with different sub image size (I) and block size (N) are obtained from the two sets of 36 images each. The classification is performed with K-NN classifier to find the optimum I and N which gives high classification accuracy. The results are tabulated in Table 4.13.

The table shows that 32×32 sub image size gave high accuracy. But GA optimization of 32×32 sub image size texture features gave only 56.25% accuracy. The reduction in classification accuracy is due to the presence of image water mark. Hence, GA optimization is performed for 16×16 and 8×8 sub images cropped from area avoiding watermark.



Fig. 4.10: Image Samples of Skin Diseases

Table 4.13: Comparison of feature sets with different I

Sl. No.	Sub Image Size(I)	% η of K-NN for different k				
		K=1	K=2	K=3	K=4	K=5
1	64x64	25	18.75	43.75	43.75	43.75
2	32x32	37.5	37.5	43.75	50	56.25
3	16x16	31.25	31.25	31.25	37.5	37.57
4	8x8	25	25	31.25	31.25	31.25

GA optimization of SMRT texture features derived from 16×16 sub image resulted in one feature set with 13 features and gave 100% correct classification accuracy. The GA optimized 16x16 feature set is given in the Table 4.14.

Table 4.14: Result of GA optimized 16x16 Feature set

Sl. No.	$f_{i_1}(c_1, c_2)$ or $f_{i_2}(c_1, c_2)$
1	$f_1(0, 1)$
2	$f_1(0, 4)$
3	$f_1(4, 0)$
4	$f_3(1, 1)$
5	$f_4(1, 1)$
6	$f_7(1, 1)$
7	$f_3(1, 2)$
8	$f_4(1, 2)$
9	$f_2(2, 1)$
10	$f_3(2, 1)$
11	$f_2(1, 4)$
12	$f_1(4, 1)$
13	$f_1(4, 2)$

The result in this section reveals that if a good database of dermoscopic images is available, SMRT based texture analysis technique developed in section 4.2 for prostate disease can be easily extended for detection and classification of different types of skin

diseases also. But, even after repeated email requests, permission to use the images is not obtained and hence the work is discontinued at this level.

In this stage, it is beneficial to explore the possibility of extending SMRT based texture analysis technique developed for medical images in non medical applications also. Hence a problem that is hot in agriculture field is chosen for further study.

Coconut harvesting is a big social problem faced by people of Kerala and even in many countries, which can be solved by an automated coconut harvesting system. Coconut growth stage identification is the most important step to develop such an automated system. Next section discusses about the study on effectiveness of SMRT based texture analysis technique, developed in previous sections, for identification of the maturity level of coconut from photographic images.

4.4 Coconut growth stage identification

The procedure for growth stage identification technique explained in this section is very similar to that of prostate disease diagnosis technique, as given in the block diagram Fig. 4.11. The data used for prostate disease diagnosis are slices of CT images whereas photographic images are used in the case of coconut growth stage identification.

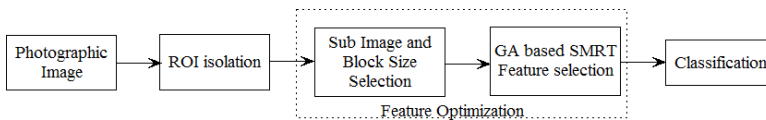


Fig. 4.11: Proposed Scheme

The first step to be performed in growth stage identification is the collection of coconut images, as explained in the next section.

Photographs of Coconut

Photographic images of coconut in different growth stages, taken in high resolution DSLR camera by experts in the field of coconut research, is collected for analysis. The different stages to which the coconuts are classified, shown in Fig. 4.12, are as explained below:

In the early stage, large number of flowers start to form near the trunk of the coconut, around the palm leaves. Nuts are formed in these flowers and later grow into bright green fruits, identified as young coconut, Fig. 4.12(a).

When the green fruit is seven months old, a jelly layer begins to develop just inside the inner surface. The meat will keep thickening and the water will sweeten as the coconut begins to ripen. This stage is known as tender coconut, Fig. 4.12(b).

In the palm tree, as the coconut continues maturing, its size increases and the husk becomes harder, Fig. 4.12(c). Also, the coconut meat inside the shell becomes hard and thick. This is the mature stage of coconut.

At the mature stage, if coconut is not plucked, it transforms to dry or brown coconut, 4.12(d) and it will not have any green color. This is the final stage of coconut on the palm tree. Images of young, tender, mature and dry coconuts are collected. The next step in coconut growth stage identification is to isolate the region of interest (ROI).



(a) Young Coconut



(b) Tender Coconut



(c) Mature Coconut



(d) Brown Coconut

Fig. 4.12: Different stages of coconut

ROI Isolation

The ROIs are cropped out manually from the images. $N \times N$ images, N varies from 8 to 256 as powers of 2, are cropped manually to optimize sub image and block size as shown in Fig. 4.13.

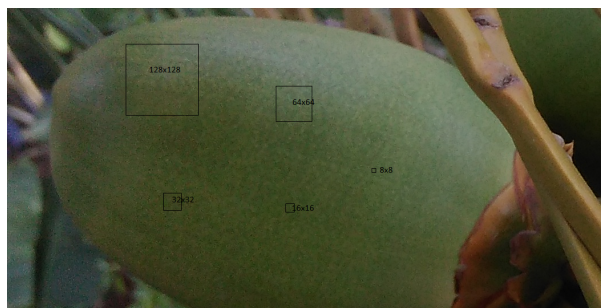


Fig. 4.13: Cropping Images of different size

The images are classified into five classes as : 1) Young Coconut 2) Tender Coconut 3) Mature Coconut 4) Dry Coconut 5) Other parts.

A total of 500 cropped images of a particular size comprising of 100 images from each class is used to analyze the classification capability of SMRT features. Next step in the coconut classification procedure is to optimize the texture features.

Optimization of SMRT texture features

In this section, subimage and block size selection and GA based optimization of SMRT texture feature set for coconut growth stage identification are discussed.

Selection of subimage size and block size The classification performance is compared for images of different subimage and block sizes, from 256×256 to 8×8 . In the study, 80% of images from each class is used for training the classifier and 20% is used for testing. Hence, training image database is of size 400, formed from 80 images of each class and test database of 100 formed from 20 images of each class. Classification accuracy is computed using K-NN classifier with different values of K. The results are given in Table 4.15 for K of K-NN classifier giving highest classification accuracy.

Table 4.15: Classification Accuracy with different Sub Image Size and Block Size

Sub Image Size	Block Size	Value of K in K-NN Classifier giving Highest Accuracy	Classification Accuracy (%)
256	256	4	38
	128	4	32
	64	5	32
128	128	4	44
	64	4	40
	32	4	38
64	64	5	60
	32	4	32
	16	4	56
32	32	5	79
	16	5	56
	8	4	43
16	16	4	40
	8	4	37
8	8	4	22

The results in the Table 4.15 shows that the classification accu-

racy is high when we choose the same sub image size and block size of 32×32 . After sub image and block size selection, GA based optimization is performed to obtain feature set with minimum number of elements giving high classification accuracy.

GA based optimization of texture features GA optimization, as discussed in section 3.3, is performed in the 32×32 feature sets to eliminate the unwanted and redundant features giving maximum classification efficiency with K-NN classifier. It improves the classification accuracy to 92 % with 38 features. After identifying the classification ability of GA optimized SMRT texture features, it is compared with GLCM texture features for the same set of images. Fourteen GLCM features proposed by Haralick is used for classification and yields a classification accuracy of 57%. After identification of optimum texture feature set the classification is performed using PNN classifier, which is found to yield good classification for prostate images.

Growth stage Identification

The result of classification with GA optimized feature set using PNN classifier is as shown below.

Table 4.16: Classification Accuracy of GA optimized feature set

	Young	Tender	Mature	Dry	Other	Class Acc%
Young	17	2	0	0	1	85
Tender	2	18	0	0	0	90
Mature	0	0	20	0	0	100
Dry	0	0	0	20	0	100
Other	2	0	0	0	18	90
						93

Color histogram feature in the RGB space, discussed in section 1.4.2, used to incorporate the color property of different stages of coconut is used along with GA optimized SMRT feature set for classification. But it has no effect on the classification accuracy.

The studies on coconut classification with SMRT texture features shows that GA optimization along with K-NN classifier as an efficient method to discriminate between different stages of coconut. This serves as a starting step to automate the process of harvesting coconut.

4.5 Conclusion

GA optimized SMRT texture features are found to be one of the best methods for feature selection when used along with K-NN classifier. The results obtained with the optimized feature set is better than that obtained with other feature sets for other classifiers also. It can be used for both medical and non medical image classification. After analyzing the experimental results for different real time images which is discussed in the chapter indicates that the SMRT texture feature gives high classification accuracy when sub image size and block size is the same. Hence in the expression for SMRT texture feature, N_b can be taken as 1.

Chapter 5

Parallel Distributed Architecture for 8×8 SMRT

Contents

5.1	Introduction	96
5.2	Modified Primitive Symbols based on 2×2 Data	96
5.3	Visual Representation of 8×8 SMRT based on 2×2 Data	98
5.4	Parallel Distributed Architecture for 8×8 SMRT based on 2×2 data	100
5.5	Parallel Distributed Architecture for 8×8 SMRT based on M-spacing Data	112
5.6	Hybrid Architecture for 8×8 SMRT	124
5.7	FPGA Implementation	134
5.8	Conclusion	135

5.1 Introduction

SMRT texture features found to be very effective in the classification of medical and non-medical images as discussed in chapter 3 and chapter 4. Different processing steps involved in SMRT based texture analysis of images are the acquisition of images, preprocessing, segmentation of ROI, texture feature extraction, feature selection and finally classification. They are implemented using different algorithms having a different scale of complexity and speed. The complexity of algorithms varies and depends on the way it is implemented.

There exist many situations in real time applications of texture analysis that requires faster computation. Transform is one of the major components that slows down the processing. Exhaustive researches are going on to improve the transformation speed.

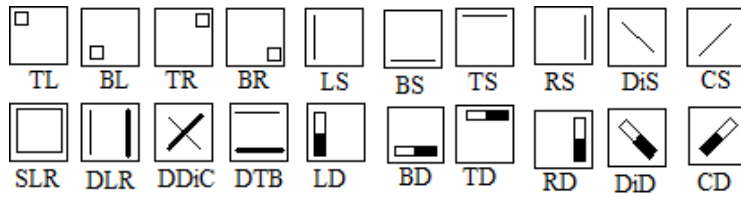
The computation time of SMRT coefficients increases with matrix size, since it is done based on basic MRT computation approach, explained in section 1.4.3, that involves too many data search.

Different parallel distributed architectures for UMRT were developed [17], as discussed in section 2.7.1, based on analysis of visual representation [15], [17]. The following sections explore the possibility of extending the hardware implementation methods developed for UMRT to implement SMRT.

5.2 Modified Primitive Symbols based on 2×2 Data

The set of primitive symbols used in the visual representation in [82] is shown in Appendix C. In the set of symbols, many are sign reversed versions of another primitive symbol. The redundant

symbols are removed from the primitive symbol set, to reduce the complexity in both hardware and software implementations. For example, LAN is sign reversed version of LAP and hence removed. Since LAP is the Top Left element of the 2×2 data matrix, it is renamed as TL. Similarly, many symbols are removed and renamed using similar mnemonics. Hence, the symbol set is reduced to 20 elements as shown in Fig. 5.1. Their names and corresponding mathematical relations are shown below.



Legend: S - Sum, D - Difference, T - Top, B - Bottom, L - Left
R - Right, Di - Diagonal, C - Cross diagonal

Fig. 5.1: Modified Primitive Symbols based on 2×2 Data

Let the 2×2 data be X ,

$$X = \begin{bmatrix} x_{00} & x_{01} \\ x_{10} & x_{11} \end{bmatrix} \quad (5.1)$$

then the primitive symbols are,

$$\begin{aligned} TL &= x_{00}, & BL &= x_{10}, & TR &= x_{01}, & BR &= x_{11} \\ LS &= x_{00} + x_{10}, & RS &= x_{01} + x_{11}, & TS &= x_{00} + x_{01}, & BS &= x_{10} + x_{11} \\ LD &= x_{00} - x_{10}, & RD &= x_{01} - x_{11}, & TD &= x_{00} - x_{01}, & BD &= x_{10} - x_{11} \\ DiS &= x_{00} + x_{11}, & CS &= x_{01} + x_{10} \\ DiD &= x_{00} - x_{11}, & CD &= x_{01} - x_{10} \\ SLR &= x_{00} + x_{10} + x_{01} + x_{11}, & DLR &= x_{00} + x_{10} - x_{01} + x_{11} \\ DDiC &= x_{00} + x_{11} - x_{10} + x_{01}, & DTB &= x_{00} + x_{01} - x_{10} + x_{11} \end{aligned}$$

5.3 Visual Representation of 8×8 SMRT based on 2×2 Data

Visual representation for 8×8 SMRT based on 2×2 data and new primitive symbols is developed, as shown in Fig. 5.2 using the primitive symbols discussed in section 5.2.

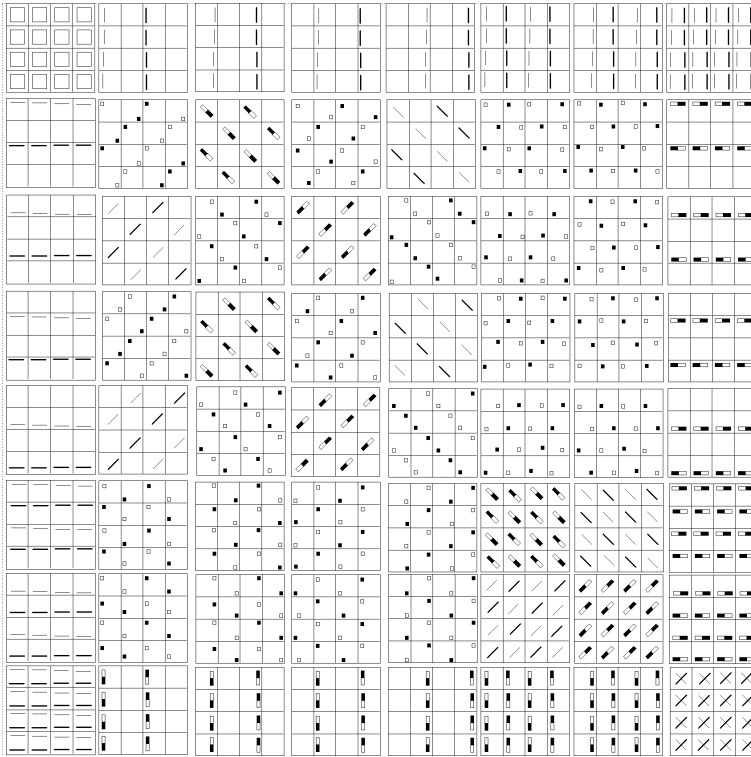


Fig. 5.2: Visual Representation of 8×8 SMRT Coefficients

The visual representation of SMRT coefficients is analyzed to develop a parallel distributed architecture. The analysis shows that

only one or two primitive symbols are involved in the computation of the coefficients coming under a particular sequency. Table 5.1 gives the details of primitive symbols based on 2×2 data involved in the computation of a particular SMRT coefficient.

Table 5.1: Primitive Symbols involved in 8x8 SMRT coefficients

Group	Sequency	SMRT Coefficient	Primitive Symbols
1	0,0	S00	SLR
	0,4	S07	DLR
	4,0	S70	DTB
	4,4	S77	DDiC
2	0,1	S01, S03	LS
		S02, S04	RS
	0,2	S05	LS
		S06	RS
	1,0	S10,S30	TS
		S20,S40	BS
	1,4	S17,S37	TD
		S27,S47	BD
	2,0	S50	TS
		S60	BS
	2,4	S57	TD
		S67	BD
	4,1	S71,S73	LD
		S72,S74	RD
4,2	S75	LD	
	S76	RD	
3	1,1	S11, S31, S13, S33	TL,BR
		S21, S41	CS
		S12,S32	DD
		S22,S42, S24, S44	BL,TR
		S23, S43	CD
		S14, S34	DiS
	1,2	S15,S35, S16,S36	TL,TR
		S25,S45,S26, S46	BL,BR
	2,1	S51,S53,S61,S63	TL,BR
		S52,S54,S62,S64	BL,TR
	2,2	S55	DiD
		S65	CS
		S56	DS
		S66	CD

5.4 Parallel Distributed Architecture for 8×8 SMRT based on 2×2 data

The group of primitive symbols involved in the computation of SMRT coefficients for a particular sequency is almost unique. Based on the similarity of computation and primitive symbols involved, the sequencies are grouped into three as in Table 5.1. Sequencies coming under group 1 are (0,0), (0,4), (4,0) and (4,4) whose computation involves primitive symbols SLR, DLR, DTB and DDiC respectively. The group 2 involves sequencies (0,1), (0,2), (1,0), (2,0), (1,4), (2,4), (4,1) and (4,2) with primitive symbols LS, RS, TS, BS, LD, RD, TD and BD. The primitive symbols involved in the computation of the remaining coefficients (group 3) are TL, BL, TR, BR, CS, DiS, CD and DiD.

The parallel distributed architecture developed for 8×8 SMRT using 2×2 based on the observations in Table 5.1 is shown in Fig. 5.3.

In layer L0, 8×8 input matrix is partitioned into 2×2 matrices. Various primitive symbols in layer L1 are computed from the partitioned 2×2 input matrices in layer L0. The primitive symbols in layer L1 are grouped into three based on the arithmetic operations involved. LS, RS, TS, BS, CS and DiS are grouped as G1, with addition operation. G2 includes LD, RD, TD, BD, CD, DiD, which are obtained using subtraction of elements. TL, BL, TR and BR extracted from the input matrix forms L1G3. The primitive symbol matrices computed in layer L1 are of size 4×4 .

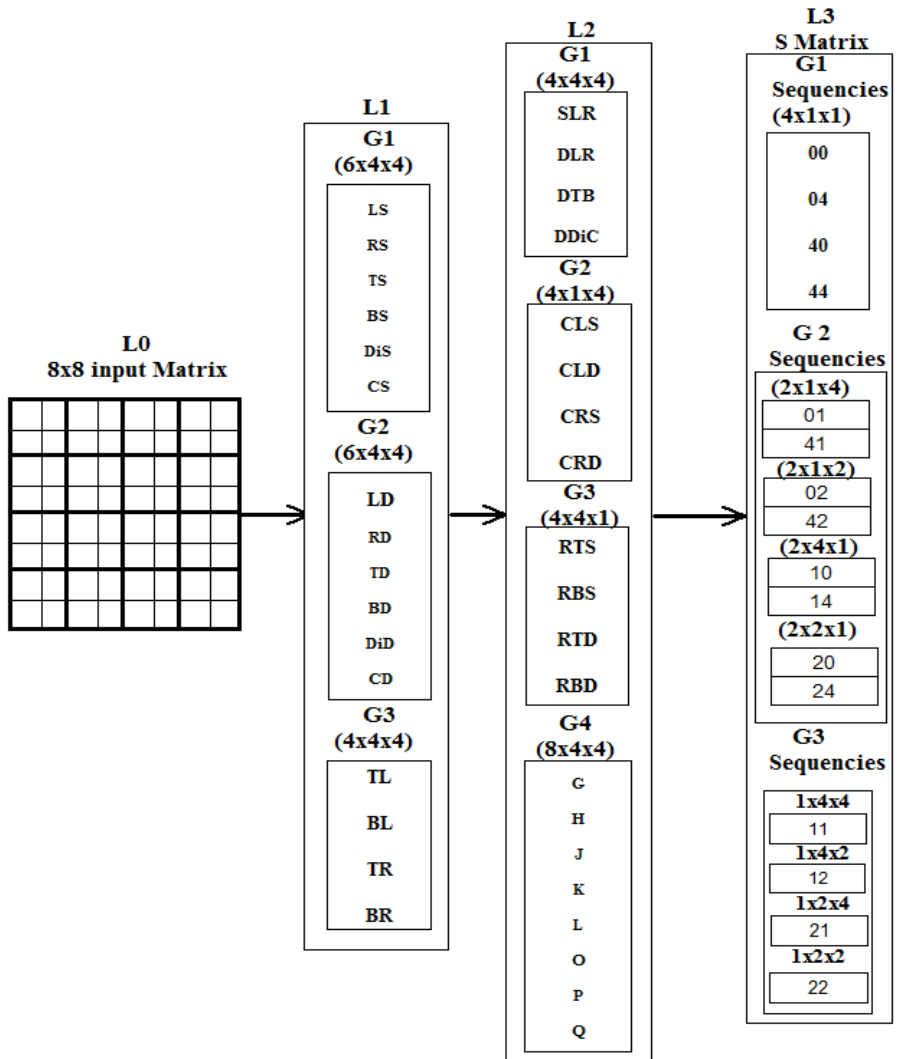


Fig. 5.3: Parallel Distributed Architecture for 8×8 SMRT based on 2×2 Data

Operations in layer L2 are performed in four groups on outputs

from layer L1. In L2G1, primitive symbols SLR, DTB, DLR, DDiC matrices of size 4×4 are obtained from LS, RS, LD and RD. Similarly in L2G2 column additions of LS, LD, RS & RD are performed and stored as row vectors CLS, CLD, CRS & CRD of size 4. In group 3 row additions of TS, TD, BS & BD are carried out to obtain 4-point column vectors RTS, RTD, RBS and RBD. Also, CS, DiS, CD, DiD, TL, BL, TR and BR is copied to 4×4 matrices G, H, J, K, L, O, P, Q respectively in L2G4.

The computations in layer L3 are done as three groups taking the output from L2. L3G1 includes sequences 00, 04, 40 & 44 with one coefficient each are computed from SLR, DLR, DTB and DDiC respectively of L2G1. CLS, CLD, CRS, CRD, RTS, RTD, RBS and RBD vectors from L2G2 and L2G3 are used to compute coefficient in sequences 01, 02, 10, 20, 14, 24, 41 and 42 respectively of L3G2. A particular coefficient in L3G1 and L3G2 depends on one primitive symbol only. The coefficients coming under sequences in L3G3 are derived from one or two primitive symbols as seen in Table 5.1 and computed from the output of L2G4.

The algorithm used for the implementation of the three layer architecture is discussed below:

Algorithm

In the architecture shown in Fig. 5.3, 8×8 input data matrix, X is partitioned into 2×2 matrices and S is the output 8×8 SMRT matrix.

Computations in L1

L1G1

For $0 \leq i, j < M = 4$

$$LS(i, j) = x(2i, 2j) + x(2i + 1, 2j)$$

$$RS(i, j) = x(2i, 2j + 1) + x(2i + 1, 2j + 1)$$

$$TS(i, j) = x(2i, 2j) + x(2i, 2j + 1)$$

$$BS(i, j) = x(2i + 1, 2j) + x(2i + 1, 2j + 1)$$

$$DiS(i, j) = x(2i, 2j) + x(2i + 1, 2j + 1)$$

$$CS(i, j) = x(2i + 1, 2j) + x(2i, 2j + 1)$$

L1G2

For $0 \leq i, j < M = 4$

$$LD(i, j) = x(2i, 2j) - x(2i + 1, 2j)$$

$$RD(i, j) = x(2i, 2j + 1) - x(2i + 1, 2j + 1)$$

$$TD(i, j) = x(2i, 2j) - x(2i, 2j + 1)$$

$$BD(i, j) = x(2i + 1, 2j) - x(2i + 1, 2j + 1)$$

$$DiD(i, j) = x(2i, 2j) - x(2i + 1, 2j + 1)$$

$$CD(i, j) = -x(2i + 1, 2j) + x(2i, 2j + 1)$$

L1G3

For $0 \leq i, j < M = 4$ $TL(i, j) = x(2i, 2j)$, $TR(i, j) = x(2i, 2j + 1)$

$$BL(i, j) = x(2i + 1, 2j), \quad BR(i, j) = x(2i + 1, 2j + 1)$$

Computations in L2

L2G1

For $0 \leq i, j < M = 4$

$$SLR(i, j) = LS(i, j) + RS(i, j) \quad DTB(i, j) = LD(i, j) + RD(i, j)$$

$$DLR(i, j) = LS(i, j) - RS(i, j) \quad DDiC(i, j) = LD(i, j) - RD(i, j)$$

L2G2

$$\text{For } 0 \leq i < M = 4 \quad CLS(i) = \sum_{j=0}^3 LS(j, i), \quad CRS(i) = \sum_{j=0}^3 RS(j, i) \\ CLD(i) = \sum_{j=0}^3 LD(j, i), \quad CRD(i) = \sum_{j=0}^3 RD(j, i)$$

L2G3

$$\text{For } 0 \leq i < M = 4 \quad RTS(i) = \sum_{j=0}^3 TS(i, j), \quad RTD(i) = \sum_{j=0}^3 TD(i, j) \\ RBS(i) = \sum_{j=0}^3 BS(i, j), \quad RBD(i) = \sum_{j=0}^3 BD(i, j)$$

L2G4

$$\text{For } 0 \leq i, j < M = 4 \\ G(i, j) = TL(i, j) \quad L(i, j) = DiS(i, j) \\ H(i, j) = BL(i, j) \quad O(i, j) = DiD(i, j) \\ J(i, j) = TR(i, j) \quad P(i, j) = CS(i, j) \\ K(i, j) = BR(i, j) \quad Q(i, j) = CD(i, j)$$

Computations in L3

L3G1

Sequency 0,0

$$S(0, 0) = \sum_{i=0}^3 \sum_{j=0}^3 SLR(i, j)$$

Sequency 0,4

$$S(0, 7) = \sum_{i=0}^3 \sum_{j=0}^3 DRL(i, j)$$

Sequency 4,0

$$S(7, 0) = \sum_{i=0}^3 \sum_{j=0}^3 DTB(i, j)$$

Sequency 4,4

$$S(7, 7) = \sum_{i=0}^3 \sum_{j=0}^3 DDiC(i, j)$$

L3G2

Sequency 0,1

$$\begin{aligned} S(0, 1) &= CLS(0) - CLS(2) & S(0, 3) &= CLS(1) - CLS(3) \\ S(0, 2) &= CRS(0) - CRS(2) & S(0, 4) &= CRS(1) - CRS(3) \end{aligned}$$

Sequency 0,2

$$\begin{aligned} S(0, 5) &= CLS(0) + CLS(2) - CLS(1) - CLS(3) \\ S(0, 6) &= CRS(0) + CRS(2) - CRS(1) - CRS(3) \end{aligned}$$

Sequency 1,0

$$\begin{aligned} S(1, 0) &= RTS(0) - RTS(2) & S(3, 0) &= RTS(1) - RTS(3) \\ S(2, 0) &= RBS(0) - RBS(2) & S(4, 0) &= RBS(1) - RBS(3) \end{aligned}$$

Sequency 2,0

$$\begin{aligned} S(5, 0) &= RTS(0) + RTS(2) - RTS(1) - RTS(3) \\ S(6, 0) &= RBS(0) + RBS(2) - RBS(1) - RBS(3) \end{aligned}$$

Sequency 1,4

$$\begin{aligned} S(1, 7) &= RTD(0) - RTD(2) & S(3, 7) &= RTD(1) - RTD(3) \\ S(2, 7) &= RBD(0) - RBD(2) & S(4, 7) &= RBD(1) - RBD(3) \end{aligned}$$

Sequency 2,4

$$\begin{aligned} S(5, 7) &= RTD(0) - RTD(1) + RTD(2) - RTD(3) \\ S(6, 7) &= RBD(0) - RBD(1) + RBD(2) - RBD(3) \end{aligned}$$

Sequency 4,1

$$\begin{aligned} S(7, 1) &= CLD(0) - CLD(2) & S(7, 3) &= CLD(1) - CLD(3) \\ S(7, 2) &= CRD(0) - CRD(2) & S(7, 4) &= CRD(1) - CRD(3) \end{aligned}$$

Sequency 4,2

$$\begin{aligned} S(7, 5) &= CLD(0) - CLD(1) + CLD(2) - CLD(3) \\ S(7, 6) &= CRD(0) - CRD(1) + CRD(2) - CRD(3) \end{aligned}$$

L3G3

Sequency 1,1

$$\begin{aligned} S(1, 1) &= G(0, 0) - G(0, 2) - K(0, 1) + K(0, 3) - K(1, 0) + K(1, 2) - \\ &G(1, 1) + G(1, 3) - G(2, 0) + G(2, 2) + K(2, 1) - K(2, 3) + K(3, 0) - \\ &K(3, 2) + G(3, 1) - G(3, 3) \\ S(2, 1) &= P(0, 0) - P(0, 2) - P(1, 1) + P(1, 3) - P(2, 0) + P(2, 2) + \\ &PG(3, 1) - P(3, 3) \\ S(3, 1) &= K(0, 0) - K(0, 2) + G(0, 1) - G(0, 3) + G(1, 0) - G(1, 2) - \\ &K(1, 1) + K(1, 3) - K(2, 0) + K(2, 2) - G(2, 1) + G(2, 3) - G(3, 0) + \\ &G(3, 2) + K(3, 1) - K(3, 3) \\ S(4, 1) &= P(0, 1) - P(0, 3) + P(1, 0) - P(1, 2) - P(2, 1) + P(2, 3) - \\ &P(3, 0) + P(3, 2) \end{aligned}$$

$$S(1, 2) = O(0, 0) - O(0, 2) + O(1, 1) - O(1, 3) - O(2, 0) + O(2, 2) - O(3, 1) + O(3, 3)$$

$$S(2, 2) = J(0, 0) - H(0, 1) - J(0, 2) + H(0, 3) + H(1, 0) + J(1, 1) - H(1, 2) - J(1, 3) - J(2, 0) + H(2, 1) + J(2, 2) - H(2, 3) - H(3, 0) - J(3, 1) + H(3, 2) + J(3, 3)$$

$$S(3, 2) = O(0, 1) - O(0, 3) - O(1, 0) + O(1, 2) - O(2, 1) + O(2, 3) + O(3, 0) - O(3, 2)$$

$$S(4, 2) = H(0, 0) - H(0, 2) + J(0, 1) - J(0, 3) - J(1, 0) + H(1, 1) + J(1, 2) - H(1, 3) - H(2, 0) - J(2, 1) + H(2, 2) + J(2, 3) + J(3, 0) - H(3, 1) - J(3, 2) + H(3, 3)$$

$$S(1, 3) = G(0, 0) + K(0, 1) - G(0, 2) - K(0, 3) + K(1, 0) - G(1, 1) - K(1, 2) + G(1, 3) - G(2, 0) - K(2, 1) + G(2, 2) + K(2, 3) - K(3, 0) + G(3, 1) + K(3, 2) - G(3, 3)$$

$$S(2, 3) = Q(0, 0) - Q(0, 2) - Q(1, 1) + Q(1, 3) + Q(2, 2) - Q(2, 0) + Q(3, 1) - Q(3, 3)$$

$$S(3, 3) = -K(0, 0) + G(0, 1) + K(0, 2) - G(0, 3) + G(1, 0) + K(1, 1) - G(1, 2) - K(1, 3) + K(2, 0) - G(2, 1) - K(2, 2) + G(2, 3) - G(3, 0) - K(3, 1) + G(3, 2) + K(3, 2)$$

$$S(4, 3) = Q(0, 1) - Q(0, 3) + Q(1, 0) - Q(1, 2) - Q(2, 1) + Q(2, 3) - Q(3, 0) + Q(3, 2)$$

$$S(1, 4) = L(0, 0) - L(0, 2) + L(1, 1) - L(1, 3) - L(2, 0) + L(2, 2) + L(3, 3) - L(3, 1)$$

$$S(2, 4) = J(0, 0) + H(0, 1) - J(0, 2) - H(0, 3) - H(1, 0) + J(1, 1) + H(1, 2) - J(1, 3) - J(2, 0) - H(2, 1) + J(2, 2) + H(2, 3) + H(3, 0) - J(3, 1) - H(3, 2) + J(3, 3)$$

$$S(3, 4) = L(0, 1) - L(0, 3) - L(1, 0) + L(1, 2) - L(2, 1) + L(2, 3) + L(3, 0) - L(3, 2)$$

$$S(4, 4) = -H(0, 0) + J(0, 1) + H(0, 2) - J(0, 3) - J(1, 0) - H(1, 1) + J(1, 2) + H(1, 3) + H(2, 0) - J(2, 1) - H(2, 2) + J(2, 3) + J(3, 0) + H(3, 1) - J(3, 2) - H(3, 3)$$

Sequency 1,2

$$S(1, 5) = G(0, 0) - G(0, 1) + G(0, 2) - G(0, 3) - J(1, 0) + J(1, 1) -$$

$$J(1, 2) + J(1, 3) - G(2, 0) + G(2, 1) - G(2, 2) + G(2, 3) + J(3, 0) - J(3, 1) + J(3, 2) - J(3, 3)$$

$$S(2, 5) = H(0, 0) - H(0, 1) + H(0, 2) - H(0, 3) - K(1, 0) + K(1, 1) - K(1, 2) + K(1, 3) - H(2, 0) + H(2, 1) - H(2, 2) + H(2, 3) + K(3, 0) - K(3, 1) + K(3, 2) - K(3, 3)$$

$$S(3, 5) = J(0, 0) - J(0, 1) + J(0, 2) - J(0, 3) + A(1, 0) - G(1, 1) + G(1, 2) - G(1, 3) - J(2, 0) + J(2, 1) - J(2, 2) + J(2, 3) - G(3, 0) + G(3, 1) - G(3, 2) + G(3, 3)$$

$$S(4, 5) = K(0, 0) - K(0, 1) + K(1, 3) - K(1, 4) + H(2, 1) - H(2, 2) + H(2, 3) - H(2, 4) - K(3, 1) + K(3, 2) - K(3, 3) + K(3, 4) - H(4, 1) + H(4, 2) - H(4, 3) + H(4, 4)$$

$$S(1, 6) = G(0, 0) - G(0, 1) + G(0, 2) - G(0, 3) + J(1, 0) - J(1, 1) + J(1, 2) - J(1, 3) - G(2, 0) + G(2, 1) - G(2, 2) + G(2, 3) - J(3, 0) + J(3, 1) - J(3, 2) + J(3, 3)$$

$$S(2, 6) = -K(0, 0) + K(0, 1) - K(0, 2) + K(0, 3) + H(1, 0) - H(1, 1) + H(1, 2) - H(1, 3) + K(2, 0) - K(2, 1) + K(2, 2) - K(2, 3) - H(3, 0) + H(3, 1) - H(3, 2) + H(3, 3)$$

$$S(3, 6) = J(0, 0) - J(0, 1) + J(0, 2) - J(0, 3) - G(1, 0) + G(1, 1) - G(1, 2) + G(1, 3) - J(2, 0) + J(2, 1) - J(2, 2) + J(2, 3) + G(3, 0) - G(3, 1) + G(3, 2) - G(3, 3)$$

$$S(4, 6) = H(0, 0) - H(0, 1) + H(0, 2) - H(0, 3) + K(1, 0) - K(1, 1) + K(1, 2) - K(1, 3) - H(2, 0) + H(2, 1) - H(2, 2) + H(2, 3) - K(3, 0) + K(3, 1) - K(3, 2) + K(3, 3)$$

Sequency 2,1

$$S(5, 1) = G(0, 0) - H(0, 1) - G(0, 2) + H(0, 3) - G(1, 0) + H(1, 1) + G(1, 2) - H(1, 3) + G(2, 0) - H(2, 1) - G(2, 2) + H(2, 3) - G(3, 0) + H(3, 1) + G(3, 2) - H(3, 3)$$

$$S(5, 2) = J(0, 0) - K(0, 1) - J(0, 2) + K(0, 3) - J(1, 0) + K(1, 1) + J(1, 2) - K(1, 3) + J(2, 0) - K(2, 1) - J(2, 2) + K(2, 3) - J(3, 0) + K(3, 1) + J(3, 2) - K(3, 3)$$

$$S(5, 3) = H(0, 0) + G(0, 1) - H(0, 2) - G(0, 3) - H(1, 0) - G(1, 1) + H(1, 2) + G(1, 3) + H(2, 0) + G(2, 1) - H(2, 2) - G(2, 3) - H(3, 0) -$$

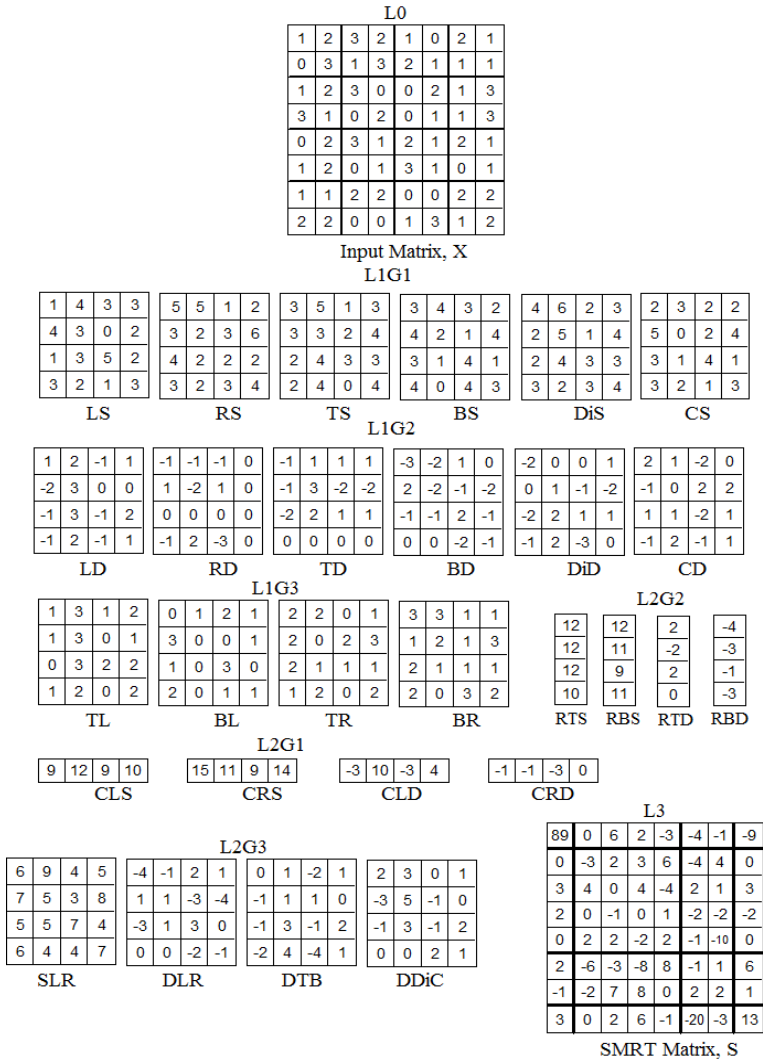
$$\begin{aligned}
&G(3, 1) + H(3, 2) + G(3, 3) \\
S(5, 4) &= K(0, 0) + J(0, 1) - K(0, 2) - J(0, 3) - K(1, 0) - J(1, 1) + \\
&K(1, 2) + J(1, 3) + K(2, 0) + J(2, 1) - K(2, 2) - J(2, 3) - K(3, 0) - \\
&J(3, 1) + K(3, 2) + J(3, 3) \\
S(6, 1) &= G(0, 0) + H(0, 1) - G(0, 2) - H(0, 3) - G(1, 0) - H(1, 1) + \\
&G(1, 2) + H(1, 3) + G(2, 0) + H(2, 1) - G(2, 2) - H(2, 3) - G(3, 0) - \\
&H(3, 1) + G(3, 2) + H(3, 3) \\
S(6, 2) &= J(0, 0) + K(0, 1) - J(0, 2) - K(0, 3) - J(1, 0) - K(1, 1) + \\
&J(1, 2) + K(1, 3) + J(2, 0) + K(2, 1) - J(2, 2) - K(2, 3) - J(3, 0) - \\
&K(3, 1) + J(3, 2) + K(3, 3) \\
S(6, 3) &= G(0, 1) - H(0, 0) + H(0, 2) - G(0, 3) + H(1, 0) - G(1, 1) - \\
&H(1, 2) + G(1, 3) - H(2, 0) + G(2, 1) + H(2, 2) - G(2, 3) + H(3, 0) - \\
&G(3, 1) - H(3, 2) + G(3, 3) \\
S(6, 4) &= -K(0, 0) + J(0, 1) + K(0, 2) - J(0, 3) + K(1, 0) - J(1, 1) - \\
&K(1, 2) + J(1, 3) - K(2, 0) + J(2, 1) + K(2, 2) - J(2, 3) + K(3, 0) - \\
&J(3, 1) - K(3, 2) + J(3, 3)
\end{aligned}$$

Sequency 2,2

$$\begin{aligned}
S(5, 5) &= O(0, 0) - O(0, 1) + O(0, 2) - O(0, 3) - O(1, 0) + O(1, 1) - \\
&O(1, 2) + O(1, 3) + O(2, 0) - O(2, 1) + O(2, 2) - O(2, 3) - O(3, 0) + \\
&O(3, 1) - O(3, 2) + O(3, 3) \\
S(6, 5) &= P(0, 0) - P(0, 1) + P(0, 2) - P(0, 3) - P(1, 0) + P(1, 1) - \\
&P(1, 2) + P(1, 3) + P(2, 0) - P(2, 1) + P(2, 2) - P(2, 3) - P(3, 0) + \\
&P(3, 1) - P(3, 2) + P(3, 3) \\
S(5, 6) &= L(0, 0) - L(0, 1) + L(0, 2) - L(0, 3) - L(1, 0) + L(1, 1) - \\
&L(1, 2) + L(1, 3) + L(2, 0) - L(2, 1) + L(2, 2) - L(2, 3) - L(3, 0) + \\
&L(3, 1) - L(3, 2) + L(3, 3) \\
S(6, 6) &= Q(0, 0) - Q(0, 1) + Q(0, 2) - Q(0, 3) - Q(1, 0) + Q(1, 1) - \\
&Q(1, 2) + Q(1, 3) + Q(2, 0) - Q(2, 1) + Q(2, 2) - Q(2, 3) - Q(3, 0) + \\
&Q(3, 1) - Q(3, 2) + Q(3, 3)
\end{aligned}$$

Example

The computation of 8×8 SMRT based on 2×2 data shown in Fig. 5.3 is illustrated with an example as given below:



The number of additions in the parallel architecture algorithm for SMRT based on 2×2 data can be summarized as in Table 5.2.

Table 5.2: 8×8 SMRT computations based on 2×2 data

Layer1		Layer2		Layer3		
Matrix and Size	No. of additions	Matrix and Size	No. of additions	Sequencies	No. of Coefficients	No. of additions
LS (4x4)	16	CLS (1x4)	4	0,0	1	15
RS (4x4)	16	CRS (1x4)	4	0,1	4	1
LD (4x4)	16	CLD (1x4)	4	0,2	2	3
RD (4x4)	16	CRD (1x4)	4	0,4	1	15
TS (4x4)	16	RTS (4x1)	4	1,0	4	1
TD (4x4)	16	RTD (4x1)	4	1,1	8	15
BS (4x4)	16	RBS (4x1)	4	1,1	8	7
BD (4x4)	16	RBD (4x1)	4	1,2	8	15
CS (4x4)	16	SLR (4x4)	16	1,4	4	1
DiS (4x4)	16	DTB (4x4)	16	2,0	2	3
CD (4x4)	16	DDiC (4x4)	16	2,1	8	15
DiD (4x4)	16	DLR (4x4)	16	2,2	4	15
				2,4	2	3
				4,0	1	15
				4,1	4	1
				4,2	2	3
				4,4	1	15
Total Additions	192	Total Additions	96		Total Additions	576
Total No. of Computations: 864						

The number of additions involved in the computation of 8×8 SMRT directly from data using the visual representation shown in Fig. 5.2 is 1344. The Table 5.2 shows that the total number of additions involved in the computation of 8×8 SMRT using the parallel distributed architecture presented in section 5.4 is 864. The inference is that parallel distributed computation reduces the number of additions from 1344 to 864, i.e. it reduces computational complexity. As seen from the Table 5.2, the maximum memory requirement is 192.

The computational complexity of coefficients in L3G3 is high when compared to L3G1 and L3G2. There are simple computational patterns for coefficients in L3G1 and L3G2, whereas it is complex for that of L3G3. Hence, new approaches are explored in the following sections.

5.5 Parallel Distributed Architecture for 8×8 SMRT based on M-spacing Data

Visual representation of coefficients shows clear patterns in terms of M-spaced data, where $M = N/2$, termed as M-spacing based data availability [17].

M-spacing based Data

Theorem 4.1 developed in [17] for the M-spacing data is restated as:

Theorem 5.1

If a data (n_1, n_2) is present in the visual representation of Y_{k_1, k_2}^P then the data at $(n_1, n_2 + M)$, $(n_1 + M, n_2)$ and $(n_1 + M, n_2 + M)$ will also be present in the computation of the coefficient.

Different sign patterns of M-spacing data is given as C, D, E and F shown in Fig. 5.4.

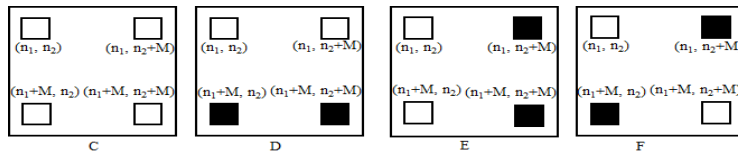


Fig. 5.4: M-Spacing Patterns

The type of M-spacing pattern present in the computation of UMRT coefficient depends on the frequency index (k_1, k_2) and based on that a theorem (Theorem 4.2) was developed in [17] which is restated as,

Theorem 5.2

One type of pattern C, D, E, or F and/or its sign reversed form will be present in the visual representation of UMRT coefficients and the type of the pattern depends on whether the frequency index k_1 and/or k_2 is even or odd.

There is a direct relationship, given in Table 5.3 [22], between sequencies c_1 & c_2 of SMRT coefficients and frequency index k_1 & k_2 of UMRT coefficients.

Table 5.3: Relationship between sequency and frequency parameters

	$c_1 \leq c_2$ $c_2 = 0$	$c_1 > c_2$ $c_1 = 0$
k_1	$c_1 \cdot (1 + 2 \cdot i_2)$	$c_1 \cdot (1 + 2 \cdot i_1)$
k_2	c_2	c_2
p	$c_1 \cdot i_1$	$c_2 \cdot i_2$

The table shows that the sequencies c_1 and c_2 of SMRT coefficients are directly related to frequency indices k_1 and k_2 respectively of UMRT coefficients. Hence the theorem (Theorem 4.2 in [17]) is modified for SMRT as,

Theorem 5.3

One type of pattern C, D, E, or F and/or its sign reversed form will be present in the visual representation of SMRT coefficients and the type of the pattern depends on whether the sequency c_1 and/or c_2 is even or odd.

Proof of theorem 5.3 is in Appendix D.

The sequency values and M-spacing patterns for 8×8 SMRT is grouped into four based on Theorem 5.3 as listed in Table 5.4.

Table 5.4: Sequencies and M-spacing Patterns

Sequency Groups		Sequencies in 8×8 SMRT	M-spacing Pattern
c_1	c_2		
even	even	00, 02, 04, 20, 22, 24, 40, 42, 44	C
odd	even	10, 12, 14	D
even	odd	01, 21, 41	E
odd	odd	11	F

A three layered parallel distributed architecture for 8×8 SMRT based on M-spacing data is developed from the above observations as shown in Fig. 5.5. Matrices A and B, of size 8×4 each in layer L1, are computed as the sum and difference respectively of input data elements 4 columns apart in L0. Matrices C, D, E & F of size 4×4 in layer L2 are computed from A & B matrices of layer L1 using patterns C, D, E and F (given in Fig. 5.4). The SMRT coefficients in layer L3 are computed, based on Table 5.4, from the matrices C, D, E, F and are grouped as G1, G2, G3, G4 respectively.

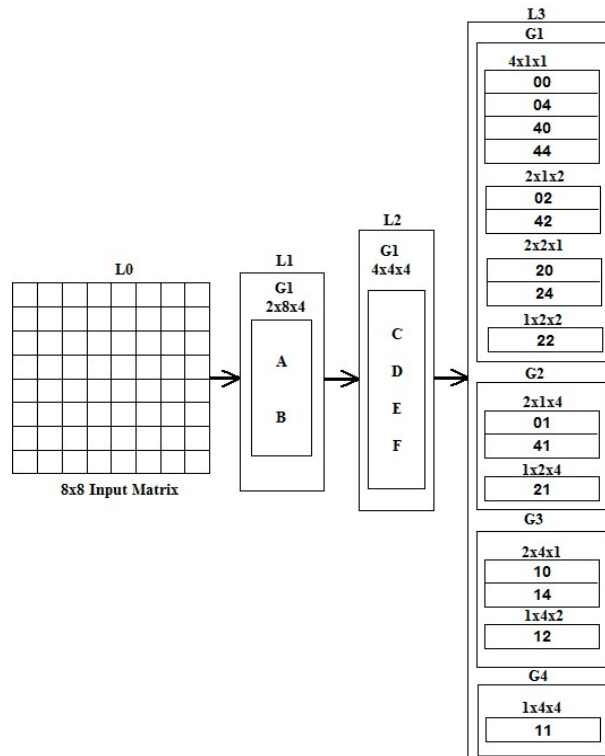


Fig. 5.5: Parallel Distributed Architecture based on M-spacing Data

Algorithm used in the computation of 8×8 SMRT matrix S from an 8×8 input data matrix, X for the parallel distributed architecture based on M-spacing data is discussed below:

Computations in L1

For $0 \leq i < 7, 0 \leq j < 3$, $A(i, j) = x(i, j) + x(i, j + 4)$, $B(i, j) = x(i, j) - x(i, j + 4)$

Computations in L2

For $0 \leq i, j \leq 3$ $C(i, j) = A(i, j) + A(i + 4, j)$, $E(i, j) = B(i, j) + B(i + 4, j)$
 $D(i, j) = A(i, j) - A(i + 4, j)$, $F(i, j) = B(i, j) - B(i + 4, j)$

Computations in L3

L3G1

Sequency 00

$$S(0, 0) = C(0, 0) + C(1, 0) + C(2, 0) + C(3, 0) + C(0, 1) + C(1, 1) + C(2, 1) + C(3, 1) + C(0, 2) + C(1, 2) + C(2, 2) + C(3, 2) + C(0, 3) + C(1, 3) + C(2, 3) + C(3, 3)$$

Sequency 02

$$S(0, 5) = C(0, 0) + C(1, 0) + C(2, 0) + C(3, 0) - C(0, 2) - C(1, 2) - C(2, 2) - C(3, 2)$$

$$S(0, 6) = C(0, 1) + C(1, 1) + C(1, 1) + C(3, 1) - C(0, 3) - C(1, 3) - C(2, 3) - C(3, 3)$$

Sequency 04

$$S(0, 7) = C(0, 0) + C(1, 0) + C(2, 0) + C(3, 0) - C(0, 1) - C(1, 1) - C(2, 1) - C(3, 1) + C(0, 2) + C(1, 2) + C(2, 2) + C(3, 2) - C(0, 3) - C(1, 3) - C(2, 3) - C(3, 3)$$

Sequency 20

$$S(5, 0) = C(0, 0) + C(0, 1) + C(0, 2) + C(0, 3) - C(2, 0) - C(2, 1) - C(2, 2) - C(2, 3)$$

$$S(6, 0) = C(1, 0) + C(1, 1) + C(1, 2) + C(1, 3) - C(3, 0) - C(3, 1) - C(3, 2) - C(3, 3)$$

Sequency 22

$$S(5, 5) = C(0, 0) - C(0, 2) - C(1, 1) + C(1, 3) - C(2, 0) + C(2, 2) + C(3, 1) - C(3, 3)$$

$$S(6, 5) = C(0, 1) - C(0, 3) + C(1, 0) - C(1, 2) - C(2, 1) + C(2, 3) - C(3, 0) + C(3, 2)$$

$$S(5, 6) = C(0, 0) - C(0, 2) + C(1, 1) - C(1, 3) - C(2, 0) + C(2, 2) - C(3, 1) + C(3, 3)$$

$$S(6, 6) = C(0, 1) - C(0, 3) - C(1, 0) + C(1, 2) - C(2, 1) + C(2, 3) + C(3, 0) - C(3, 2)$$

Sequency 24

$$S(5, 7) = C(0, 0) + C(0, 2) + C(2, 1) + C(2, 3) - C(0, 1) - C(0, 3) - C(2, 0) - C(2, 2)$$

$$S(6, 7) = C(1, 0) + C(1, 2) + C(3, 1) + C(3, 3) - C(1, 1) - C(1, 3) - C(3, 0) - C(3, 2)$$

Sequency 40

$$S(7, 0) = C(0, 0) - C(1, 0) + C(2, 0) - C(3, 0) + C(0, 1) - C(1, 1) + C(2, 1) - C(3, 1) + C(0, 2) - C(1, 2) + C(2, 2) - C(3, 2) + C(0, 3) - C(1, 3) + C(2, 3) - C(3, 3)$$

Sequency 42

$$S(7, 5) = C(0, 0) + C(2, 0) + C(1, 2) + C(3, 2) - C(0, 2) - C(2, 2) - C(1, 0) - C(3, 0)$$

$$S(7, 6) = C(0, 1) + C(2, 1) + C(1, 3) + C(3, 3) - C(0, 3) - C(2, 3) - C(1, 1) - C(3, 1)$$

Sequency 44

$$S(7, 7) = C(0, 0) - C(1, 0) + C(2, 0) - C(3, 0) - C(0, 1) + C(1, 1) - C(2, 1) + C(3, 1) + C(0, 2) - C(1, 2) + C(2, 2) - C(3, 2) - C(0, 3) + C(1, 3) - C(2, 3) + C(3, 3)$$

L3G2

Sequency 01

$$S(0, 1) = E(0, 0) + E(1, 0) + E(2, 0) + E(3, 0)$$

$$S(0, 2) = E(0, 1) + E(1, 1) + E(2, 1) + E(3, 1)$$

$$S(0, 3) = E(0, 2) + E(1, 2) + E(2, 2) + E(3, 2)$$

$$S(0, 4) = E(0, 3) + E(1, 3) + E(2, 3) + E(3, 3)$$

Sequency 21

$$S(5, 1) = E(0, 0) - E(1, 2) - E(2, 0) + E(3, 2)$$

$$S(5, 2) = E(0, 1) - E(1, 3) - E(2, 1) + E(3, 3)$$

$$S(5, 3) = E(0, 2) + E(1, 0) - E(2, 2) - E(3, 0)$$

$$S(5, 4) = E(0, 3) + E(1, 1) - E(2, 3) - E(3, 1)$$

$$S(6, 1) = E(0, 0) + E(1, 2) - E(2, 0) - E(3, 2)$$

$$S(6, 2) = E(0, 1) + E(1, 3) - E(2, 1) - E(3, 3)$$

$$S(6, 3) = E(0, 2) - E(1, 0) - E(2, 2) + E(3, 0)$$

$$S(6, 4) = E(0, 3) - E(1, 1) - E(2, 3) + E(3, 1)$$

Sequency 41

$$S(7, 1) = E(0, 0) - E(1, 0) + E(2, 0) - E(3, 0)$$

$$S(7, 2) = E(0, 1) - E(1, 1) + E(2, 1) - E(3, 1)$$

$$S(7, 3) = E(0, 2) - E(1, 2) + E(2, 2) - E(3, 2)$$

$$S(7, 4) = E(0, 3) - E(1, 3) + E(2, 3) - E(3, 3)$$

L3G3

Sequency 10

$$S(1, 0) = D(0, 0) + D(0, 1) + D(0, 2) + D(0, 3)$$

$$S(2, 0) = D(1, 0) + D(1, 1) + D(1, 2) + D(1, 3)$$

$$S(3, 0) = D(2, 0) + D(2, 1) + D(2, 2) + D(2, 3)$$

$$S(4, 0) = D(3, 0) + D(3, 1) + D(3, 2) + D(3, 3)$$

Sequency 12

$$S(1, 5) = D(0, 0) - D(0, 2) - D(2, 1) + D(2, 3)$$

$$S(2, 5) = D(1, 0) - D(1, 2) - D(3, 1) + D(3, 3)$$

$$S(3, 5) = D(0, 1) - D(0, 3) - D(2, 2) + D(2, 0)$$

$$S(4, 5) = D(1, 1) - D(1, 3) + D(3, 0) - D(3, 2)$$

$$S(1, 6) = D(0, 0) - D(0, 2) + D(2, 1) - D(2, 3)$$

$$S(2, 6) = D(1, 3) - D(1, 1) + D(3, 0) - D(3, 2)$$

$$S(3, 6) = D(0, 1) - D(0, 3) - D(2, 0) + D(2, 2)$$

$$S(4, 6) = D(1, 0) - D(1, 2) + D(3, 1) - D(3, 3)$$

Sequency 14

$$S(1, 7) = D(0, 0) - D(0, 1) + D(0, 2) - D(0, 3)$$

$$S(2, 7) = D(1, 0) - D(1, 1) + D(1, 2) - D(1, 3)$$

$$S(3, 7) = D(2, 0) - D(2, 1) + D(2, 2) - D(2, 3)$$

$$S(4, 7) = D(3, 0) - D(3, 1) + D(3, 2) - D(3, 3)$$

L3G4

Sequency 11

$$S(1, 1) = F(0, 0) - F(1, 3) - F(2, 2) - F(3, 1)$$

$$S(2, 1) = F(0, 1) + F(1, 0) - F(2, 3) - F(3, 2)$$

$$S(3, 1) = F(0, 2) + F(1, 1) + F(2, 0) - F(3, 3)$$

$$S(4, 1) = F(0, 3) + F(1, 2) + F(2, 1) + F(3, 0)$$

$$S(1, 2) = F(0, 0) - F(1, 1) + F(2, 2) - F(3, 3)$$

$$S(2, 2) = F(0, 1) - F(1, 2) + F(2, 3) + F(3, 0)$$

$$S(3, 2) = F(0, 2) - F(1, 3) - F(2, 0) + F(3, 1)$$

$$S(4, 2) = F(0, 3) + F(1, 0) - F(2, 1) + F(3, 2)$$

$$S(1, 3) = F(0, 0) + F(1, 3) - F(2, 2) + F(3, 1)$$

$$S(2, 3) = F(0, 1) - F(1, 0) - F(2, 3) + F(3, 2)$$

$$S(3, 3) = F(0, 2) - F(1, 1) + F(2, 0) + F(3, 3)$$

$$S(4, 3) = F(0, 3) - F(1, 2) + F(2, 1) - F(3, 0)$$

$$S(1, 4) = F(0, 0) + F(1, 1) + F(2, 2) + F(3, 3)$$

$$S(2, 4) = F(0, 1) + F(1, 2) + F(2, 3) - F(3, 0)$$

$$S(3, 4) = F(0, 2) + F(1, 3) - F(2, 0) - F(3, 1)$$

$$S(4, 4) = F(0, 3) - F(1, 0) - F(2, 1) - F(3, 2)$$

The calculations in each layer can be summarized as in Table 5.5 which clearly shows that the number of additions is further reduced as compared to that of 2×2 data based architecture shown in Table 5.2.

Table 5.5: 8x8 SMRT Computations based on M-spacing

Layer1		Layer2		Layer3		
Matrix and Size	No. of additions	Matrix and Size	No. of additions	Sequences	Coefficients	No. of additions
A (8X4)	32	C (4x4)	16	0,0	1	15
B (8x4)	32	D (4x4)	16	0,1	4	3
		E (4x4)	16	0,2	2	7
		F (4x4)	16	0,4	1	15
				1,0	4	3
				1,1	8	3
				1,1	8	3
				1,2	8	3
				1,4	4	3
				2,0	2	7
				2,1	8	3
				2,2	4	7
				2,4	2	7
				4,0	1	15
				4,1	4	3
				4,2	2	7
				4,4	1	15
Total Additions	64	Total Additions	64	Total Additions		288
Total No. of Computations: 416						

Tables 5.2 and 5.5 show that the computations in both architectures are unevenly distributed among layers. The computational

complexity in both the architectures is more in layer L3 compared to the other two layers.

The algorithms developed for the 8×8 SMRT using 2×2 data and M-spacing data based architectures are analyzed to reduce the computational complexity of L3. The number of additions in L3 required for the computation of coefficients in row 0, row 7, column 0 and column 7 of SMRT matrix in both architectures is as in Table 5.6

The number of additions in L3, for the sequency packets in the outer rows and columns of SMRT matrix, is less for 2×2 data based architecture compared to that of M-spacing based architecture. Also, 2×2 data based architecture has a predictable pattern of primitive symbols in the computation for the sequency packets given in Table 5.6.

A similar analysis is performed for the sequency packets in the inner rows and columns of SMRT matrix. The derived conclusions are tabulated as shown in Table 5.7.

Table 5.6: No. of additions in rows 0, 7 & columns 0, 7 of SMRT matrix for both the architectures

Sequency Packets	No. of Coefficients per sequency packet	No. of additions per coefficient	
		2x2 Data	M-spacing Data
00, 04, 40, 44	1	15	15
01,10,14,41	4	1	3
02,20,24,42	2	3	7

Table 5.7: No. of additions in inner rows/ columns of SMRT matrix for both the architectures

Sequency Packets	No. of Co-efficients per sequency packet	No. of additions per coefficient	
		2x2 Data	M-spacing Data
11	16	7/15	3
12	8	15	3
21	8	15	3
22	4	15	7

The observations in Table 5.7 show that the number of additions for sequency packets in the inner rows and columns of 8×8 SMRT matrix is less for M-spacing based architecture compared to that of 2×2 data based architecture. The algorithm for M-spacing based architecture given in section 5.5 shows that the computation pattern is also simple for M-spacing data.

The analysis suggests that a hybrid architecture which uses 2×2 data based algorithm for rows 0, 7 and columns 0, 7 and M-spacing based algorithm for all the inner rows and columns.

5.6 Hybrid Architecture for 8×8 SMRT

A hybrid three layered architecture, shown in Fig. 5.6, is developed based on the observations made in section 5.5. In the hybrid architecture, 2×2 data based algorithm is used when any of the sequency is 0 or 4 and M-spacing data based algorithm is used for all other sequency combinations.

L1G1 and L1G2, in layer L1, of Fig. 5.6 is derived by removing

CS, DiS, CD and DiD from L1G1 and L1G2 of 2×2 data based architecture shown in Fig. 5.3. L1G3 of layer L1 is same as L1 of M-spacing based architecture in Fig. 5.5. L2G1, L2G2 and L2G3 in layer L2 of Fig. 5.6 is same as L2G1, L2G2 and L2G3 of 2×2 data based architecture shown in Fig. 5.3. L2G4 of layer L2 in Fig. 5.6 is same as L2 of M-spacing based architecture in Fig. 5.5.

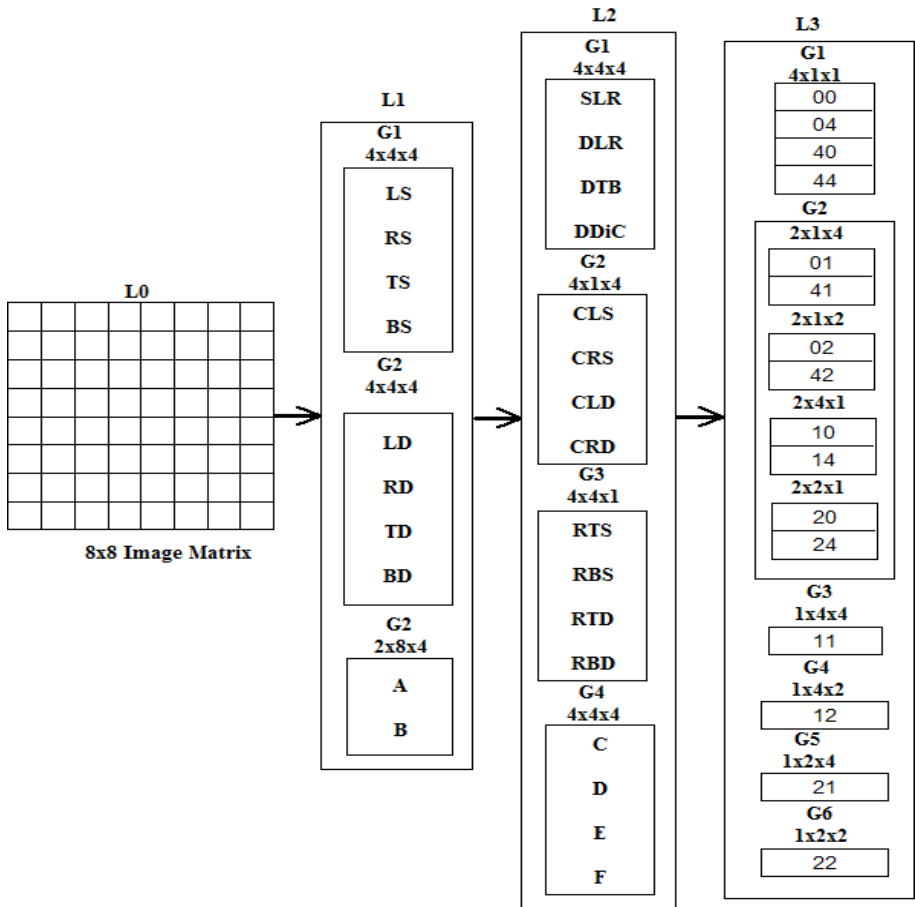


Fig. 5.6: Hybrid Architecture based on combination of M-spacing and 2×2 Data

The algorithm for the hybrid architecture for an 8×8 data matrix is explained below :

Computations in L1

L1G1

For $0 \leq i, j < 4$

$$LS(i, j) = x(2i, 2j) + x(2i + 1, 2j), \quad RS(i, j) = x(2i, 2j + 1) + x(2i + 1, 2j + 1)$$

$$TS(i, j) = x(2i, 2j) + x(2i, 2j + 1), \quad BS(i, j) = x(2i + 1, 2j) + x(2i + 1, 2j + 1)$$

L1G2

For $0 \leq i, j < 4$

$$LD(i, j) = x(2i, 2j) - x(2i + 1, 2j)$$

$$TD(i, j) = x(2i, 2j) - x(2i, 2j + 1)$$

$$RD(i, j) = x(2i, 2j + 1) - x(2i + 1, 2j + 1)$$

$$BD(i, j) = x(2i + 1, 2j) - x(2i + 1, 2j + 1)$$

L1G3

For $0 \leq i < 7, 0 \leq j < 3,$

$$A(i, j) = x(i, j) + x(i, j + 4), \quad B(i, j) = x(i, j) - x(i, j + 4)$$

Computations in L2

L2G1

$$\text{For } 0 \leq i < M = 4 \quad \begin{aligned} CLS(i) &= \sum_{j=0}^3 LS(j, i), & CRS(i) &= \sum_{j=0}^3 RS(j, i) \\ CLD(i) &= \sum_{j=0}^3 LD(j, i), & CRD(i) &= \sum_{j=0}^3 RD(j, i) \end{aligned}$$

L2G2

$$\text{For } 0 \leq i < M = 4 \quad \begin{aligned} RTS(i) &= \sum_{j=0}^3 TS(i, j), & RTD(i) &= \sum_{j=0}^3 TD(i, j) \\ RBS(i) &= \sum_{j=0}^3 BS(i, j), & RBD(i) &= \sum_{j=0}^3 BD(i, j) \end{aligned}$$

L2G3

$$\text{For } 0 \leq i, j < M = N/2$$

$$\begin{aligned} SLR(i, j) &= LS(i, j) + RS(i, j) \\ DLR(i, j) &= LS(i, j) - RS(i, j) \\ DTB(i, j) &= LD(i, j) + RD(i, j) \\ DDiC(i, j) &= LD(i, j) - RD(i, j) \end{aligned}$$

L2G4

$$\text{For } 0 \leq j < 3$$

$$\begin{aligned} C(i, j) &= A(i, j) + A(i + 4, j) \\ D(i, j) &= A(i, j) - A(i + 4, j) \end{aligned}$$

$$E(i, j) = B(i, j) + B(i + 4, j)$$

$$F(i, j) = B(i, j) - B(i + 4, j)$$

Computations in L3

L3G1

Sequency 00

$$S(0, 0) = \sum_{i=0}^3 \sum_{j=0}^3 SLR(i, j)$$

Sequency 40

$$S(7, 0) = \sum_{i=0}^3 \sum_{j=0}^3 DTB(i, j)$$

Sequency 01

$$S(0, 1) = CLS(0) - CLS(2)$$

$$S(0, 2) = CRS(0) - CRS(2)$$

$$S(0, 3) = CLS(1) - CLS(3)$$

$$S(0, 4) = CRS(1) - CRS(3)$$

Sequency 02

$$S(0, 5) = CLS(0) + CLS(2) - CLS(1) - CLS(3)$$

$$S(0, 6) = CRS(0) + CRS(2) - CRS(1) - CRS(3)$$

Sequency 10

$$S(1, 0) = RTS(0) - RTS(2)$$

$$S(2, 0) = RBS(0) - RBS(2)$$

$$S(3, 0) = RTS(1) - RTS(3)$$

$$S(4, 0) = RBS(1) - RBS(3)$$

Sequency 20

$$S(5, 0) = RTS(0) + RTS(2) - RTS(1) - RTS(3)$$

$$S(6, 0) = RBS(0) + RBS(2) - RBS(1) - RBS(3)$$

Sequency 41

$$S(7, 1) = CLD(0) - CLD(2)$$

$$S(7, 2) = CRD(0) - CRD(2)$$

$$S(7, 3) = CLD(1) - CLD(3)$$

$$S(7, 4) = CRD(1) - CRD(3)$$

Sequency 42

$$S(7, 5) = CLD(0) - CLD(1) + CLD(2) - CLD(3)$$

$$S(7, 6) = CRD(0) - CRD(1) + CRD(2) - CRD(3)$$

Sequency 14

$$S(1, 7) = RTD(0) - RTD(2)$$

$$S(2, 7) = RBD(0) - RBD(2)$$

$$S(3, 7) = RTD(1) - RTD(3)$$

$$S(4, 7) = RBD(1) - RBD(3)$$

Sequency 24

$$S(5, 7) = RTD(0) - RTD(1) + RTD(2) - RTD(3)$$

$$S(6, 7) = RBD(0) - RBD(1) + RBD(2) - RBD(3)$$

Sequency 44

$$S(7, 7) = \sum_{i=0}^3 \sum_{j=0}^3 DDiC(i, j)$$

Sequency 11

$$S(1, 1) = F(0, 0) - F(1, 3) - F(2, 2) - F(3, 1)$$

$$S(2, 1) = F(0, 1) + F(1, 0) - F(2, 3) - F(3, 2)$$

$$S(3, 1) = F(0, 2) + F(1, 1) + F(2, 0) - F(3, 3)$$

$$S(4, 1) = F(0, 3) + F(1, 2) + F(2, 1) + F(3, 0)$$

$$S(1, 2) = F(0, 0) - F(1, 1) + F(2, 2) - F(3, 3)$$

$$S(2, 2) = F(0, 1) - F(1, 2) + F(2, 3) + F(3, 0)$$

$$S(3, 2) = F(0, 2) - F(1, 3) - F(2, 0) + F(3, 1)$$

$$S(4, 2) = F(0, 3) + F(1, 0) - F(2, 1) + F(3, 2)$$

$$S(1, 3) = F(0, 0) + F(1, 3) - F(2, 2) + F(3, 1)$$

$$S(2, 3) = F(0, 1) - F(1, 0) - F(2, 3) + F(3, 2)$$

$$S(3, 3) = F(0, 2) - F(1, 1) + F(2, 0) + F(3, 3)$$

$$S(4, 3) = F(0, 3) - F(1, 2) + F(2, 1) - F(3, 0)$$

$$S(1, 4) = F(0, 0) + F(1, 1) + F(2, 2) + F(3, 3)$$

$$S(2, 4) = F(0, 1) + F(1, 2) + F(2, 3) - F(3, 0)$$

$$S(3, 4) = F(0, 2) + F(1, 3) - F(2, 0) - F(3, 1)$$

$$S(4, 4) = F(0, 3) - F(1, 0) - F(2, 1) - F(3, 2)$$

Sequency 12

$$S(1, 5) = D(0, 0) - D(0, 2) - D(2, 1) + D(2, 3)$$

$$S(2, 5) = D(1, 0) - D(1, 2) - D(3, 1) + D(3, 3)$$

$$S(3, 5) = D(0, 1) - D(0, 3) - D(2, 2) + D(2, 0)$$

$$S(4, 5) = D(1, 1) - D(1, 3) + D(3, 0) - D(3, 2)$$

$$S(1, 6) = D(0, 0) - D(0, 2) + D(2, 1) - D(2, 3)$$

$$S(2, 6) = D(1, 3) - D(1, 1) + D(3, 0) - D(3, 2)$$

$$S(3, 6) = D(0, 1) - D(0, 3) - D(2, 0) + D(2, 2)$$

$$S(4, 6) = D(1, 0) - D(1, 2) + D(3, 1) - D(3, 3)$$

Sequency 21

$$S(5, 1) = E(0, 0) - E(1, 2) - E(2, 0) + E(3, 2)$$

$$S(5, 2) = E(0, 1) - E(1, 3) - E(2, 1) + E(3, 3)$$

$$S(5, 3) = E(0, 2) + E(1, 0) - E(2, 2) - E(3, 0)$$

$$S(5, 4) = E(0, 3) + E(1, 1) - E(2, 3) - E(3, 1)$$

$$S(6, 1) = E(0, 0) + E(1, 2) - E(2, 0) - E(3, 2)$$

$$S(6, 2) = E(0, 1) + E(1, 3) - E(2, 1) - E(3, 3)$$

$$S(6, 3) = E(0, 2) - E(1, 0) - E(2, 2) + E(3, 0)$$

$$S(6, 4) = E(0, 3) - E(1, 1) - E(2, 3) + E(3, 1)$$

Sequency 22

$$S(5, 5) = C(0, 0) - C(0, 2) - C(1, 1) + C(1, 3) - C(2, 0) + C(2, 2) + C(3, 1) - C(3, 3)$$

$$S(6, 5) = C(0, 1) - C(0, 3) + C(1, 0) - C(1, 2) - C(2, 1) + C(2, 3) - C(3, 0) + C(3, 2)$$

$$S(5, 6) = C(0, 0) - C(0, 2) + C(1, 1) - C(1, 3) - C(2, 0) + C(2, 2) - C(3, 1) + C(3, 3)$$

$$S(6, 6) = C(0, 1) - C(0, 3) - C(1, 0) + C(1, 2) - C(2, 1) + C(2, 3) + C(3, 0) - C(3, 2)$$

The total number of computations in each layer of hybrid algorithm can be summarized as in Table 5.8.

Comparison of the algorithms of three architectures, it is clear that in the first two algorithms number of additions is more in layer L3. But in the hybrid algorithm, the number of additions is almost equally distributed between layers as shown in Table 5.9.

Table 5.8: 8x8 SMRT Computations Based on Hybrid Algorithm

Layer1		Layer2		Layer3		
Matrix and Size	No. of additions	Matrix and Size	No. of additions	Sequencies	Coefficients	No. of additions
A (8X4)	32	C (4x4)	16	0,0	1	15
B (8x4)	32	D (4x4)	16	0,1	4	1
LS (4x4)	16	E (4x4)	16	0,2	2	3
RS (4x4)	16	F (4x4)	16	0,4	1	15
LD (4x4)	16	CLS (4x1)	4	1,0	4	1
RD (4x4)	16	CRS (4x1)	4	1,1	16	3
TS (4x4)	16	CLD (4x1)	4	1,2	8	3
TD (4x4)	16	CRD (4x1)	4	1,4	4	1
BS (4x4)	16	RTS (4x1)	4	2,0	2	3
BD (4x4)	16	RTD (4x1)	4	2,1	8	3
		RBS (4x1)	4	2,2	4	7
		RBD (4x1)	4	2,4	2	3
		SLR (4x4)	16	4,0	1	15
		DTB (4x4)	16	4,1	4	1
		DDiC (4x4)	16	4,2	2	3
		DLR (4x4)	16	4,4	1	15
Total Additions	192	Total Additions	160	Total Additions		224
Total No. of Computations: 640						

Table 5.9: Comparison of the three algorithms based on number of computations

Layer wise number of computations				
	Layer1	Layer2	Layer3	Total
VR Based	192	96	576	864
M-spacing based	64	64	288	416
Hybrid	192	160	224	576

The three algorithms are simulated in MATLAB software with different 8x8 input matrices. Simulation is performed using MATLAB 7.12 (R2011a) software package on Intel core i5 machine, 2.4 GHz, 4 GB RAM. The results tabulated in Table 5.10 shows that M-spacing based algorithm is faster than the other two.

Table 5.10: Comparison of the three algorithms based on computation time

	Time(msec)
VR Based	69.5
M-spacing based	16.7
Hybrid	36.4

5.7 FPGA Implementation

The three parallel distributed architectures described in sections 5.4 to 5.6 for 8x8 SMRT are implemented in Xilinx ISE 14.7 using VHDL on Virtex 5 FPGA- XC5VLX30 as the target device. The algorithms are simulated using file input-output operations in VHDL. The input matrix is written as a text script. The SMRT output is also obtained as text script. All the algorithms are implemented as fully parallel by exploiting the parallelism of FPGA and Table 5.11 shows the results.

Table 5.11: Results of Fully Parallel Implementation of 8x8 UMRT

Performance Factors	Visual Representation	M-spacing	Hybrid
No. of Adders/ Subtractors	904	401	651
Cell Usage- BELS	19978	8991	14237
No. of slice LUTs	7091	3169	5085
Max. Combina- tional Path de- lay (ns)	19.212	18.594	19.193
Logic Delay (ns)	10.005	10.005	10.005
Routing Delay (ns)	9.207	8.589	9.188

5.8 Conclusion

The three different parallel distributed architectures developed for implementation of 8×8 SMRT are compared based on software simulation and FPGA based hardware implementation. Computational complexity and time are minimum for M-spacing data based approach and maximum for 2×2 data based approach.

But, the computational complexity of layer L3 is much high compared to layer L1 and L2 in parallel distributed architecture based on M-spacing data. The computational complexity is almost equally distributed among three layers in parallel distributed architecture based on hybrid approach. Also, it has predictable computation patterns that can be extended for the computation of higher order SMRT matrices.

Chapter 6

Parallel Distributed Architecture for SMRT

Contents

6.1	Introduction	138
6.2	Hybrid Architecture for $N \times N$ SMRT, N a power of 2	139
6.2.1	Algorithm	140
6.2.2	Matlab Simulation	151
6.2.3	FPGA Implementation	152
6.3	Software Hardware Co-development .	154
6.3.1	Prostate Disease Diagnosis using Soft- ware Hardware Co-development	155
6.3.2	Hardware Implementation of SMRT Texture Feature Extractor for Prostate Disease Diagnosis	156
6.4	Forward and inverse N-point SMRT, N a power of 2	161

6.4.1 M-spacing Data for N-point SMRT . . 163
6.4.2 Forward N-point SMRT Algorithm . . 164
6.4.3 Inverse N-point SMRT Algorithm . . . 166
6.5 Conclusion 169

6.1 Introduction

The parallel distributed architecture developed and implemented for 8×8 SMRT using different approaches are discussed in sections 5.4, 5.5 and 5.6.

The visual representation of 4×4 SMRT and the corresponding algorithms, derived to develop 2×2 & M-spacing data based architectures, are given in Appendix F.

Following observations are made for different $N \times N$ data, after analysis of 2×2 data and M-spacing data based on the informations in chapter 5 and Appendix F:

1. Computational complexity, measured as the number of additions given in Tables 5.2 & 5.5, is more on the third layer compared to the first two layers in 2×2 & M-spacing data based approaches and increases as size of the input data increases.
2. When any of the sequency index is 0 or M, less complex computation and simple generalization are possible for 2×2 data based approach. But for all other sequency combinations, computations are highly complex and the complexity increases with the increase in size of the input data.
3. Computational complexity is higher for outer rows/columns (sequency is 0 or M) and increases with size of the input data in M-spacing data based approach. Whereas the M-spacing data based approach is simple compared to that of 2×2 data based approach for inner rows and columns of SMRT matrix.

Thus the hybrid architecture developed for 8×8 SMRT discussed in section 5.6 can be generalized for $N \times N$ SMRT, N a power of 2.

6.2 Hybrid Architecture for $N \times N$ SMRT, N a power of 2

A generalized hybrid architecture is developed for $N \times N$ SMRT, as shown in Fig. 6.1, based on 2×2 data based approach for sequency index 0 or M and M -spacing approach for other sequency index combinations.

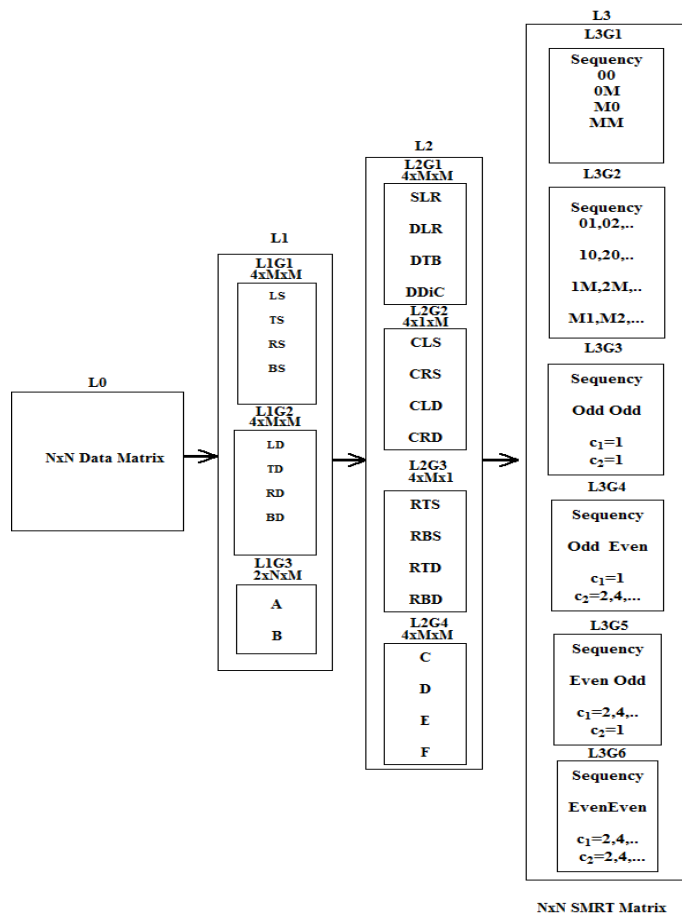


Fig. 6.1: Hybrid Architecture for $N \times N$ SMRT

Layers L1 and L2 are derived as in section 5.6 and the size of matrices are as given in Fig. 6.1. In layer L3, SMRT coefficients are computed from the output of layer L2 and grouped into six.

Algorithm is explained in the following subsection.

6.2.1 Algorithm

Computations in L1

Four primitive symbols LS, RS, TS and BS in L1G1 are computed from the addition of data elements, whereas LD, RD, TD and BD in L1G2 are obtained from subtraction of the corresponding elements based on 2×2 data and results in $M \times M$ matrices. A and B matrices in L1G3, of size $N \times M$, are obtained as the sum and difference respectively from the corresponding elements M columns apart in the input data matrix.

L1G1

For $0 \leq i, j < M = \frac{N}{2}$

$$LS(i, j) = x(2i, 2j) + x(2i + 1, 2j)$$

$$TS(i, j) = x(2i, 2j) + x(2i, 2j + 1)$$

$$RS(i, j) = x(2i, 2j + 1) + x(2i + 1, 2j + 1)$$

$$BS(i, j) = x(2i + 1, 2j) + x(2i + 1, 2j + 1)$$

L1G2

For $0 \leq i, j < M = \frac{N}{2}$

$$LD(i, j) = x(2i, 2j) - x(2i + 1, 2j)$$

$$TD(i, j) = x(2i, 2j) - x(2i, 2j + 1)$$

$$RD(i, j) = x(2i, 2j + 1) - x(2i + 1, 2j + 1)$$

$$BD(i, j) = x(2i + 1, 2j) - x(2i + 1, 2j + 1)$$

L1G3

For $0 \leq i < N - 1, 0 \leq j < M - 1$

$$A(i, j) = x(i, j) + x(i, j + M)$$

$$B(i, j) = x(i, j) - x(i, j + M)$$

Computations in L2

The computations in layer L2 are performed as four groups based on the results from layer L1.

L2G1

For $0 \leq i, j < M$

$$SLR(i, j) = LS(i, j) + RS(i, j)$$

$$DLR(i, j) = LS(i, j) + RS(i, j)$$

$$DTB(i, j) = LD(i, j) + RD(i, j)$$

$$DDiC(i, j) = LD(i, j) - RD(i, j)$$

L2G2

For $0 \leq i < M$

$$\begin{aligned}CLS(i) &= \sum_{j=0}^{M-1} LS(j, i), & CRS(i) &= \sum_{j=0}^{M-1} RS(j, i) \\CLD(i) &= \sum_{j=0}^{M-1} LD(j, i), & CRD(i) &= \sum_{j=0}^{M-1} RD(j, i)\end{aligned}$$

L2G3

For $0 \leq i < M$

$$\begin{aligned}RTS(i) &= \sum_{j=0}^{M-1} TS(i, j), & RTD(i) &= \sum_{j=0}^{M-1} TD(i, j) \\RBS(i) &= \sum_{j=0}^{M-1} BS(i, j), & RBD(i) &= \sum_{j=0}^{M-1} BD(i, j)\end{aligned}$$

L2G4

For $0 \leq j < M - 1$

$$\begin{aligned}C(i, j) &= A(i, j) + A(i + M, j) & D(i, j) &= A(i, j) - A(i + M, j) \\E(i, j) &= B(i, j) + B(i + M, j) & F(i, j) &= B(i, j) - B(i + M, j)\end{aligned}$$

Computations in L3

All SMRT coefficients are computed using the output of layer L2 and placed in different sequency packets, grouped into six, in layer L3.

The algorithm for computation and placement of elements in layer L3 involves the following steps in general:

1. The range of sequencies in the group is to be identified.
2. The size of each sequency packet is to be determined.
3. Address of the first element in the first sequency packet of the group has to be initialized.

4. The elements in the sequency packets are to be computed and placed.

The implementation steps are as explained below:

L3G1 and L3G2 represent groups of elements in row 0, row N-1, column 0 and column N-1, where any of the sequency is 0 or M. The remaining coefficients are grouped as different sequency packets in L3G3 to L3G6.

L3G1

Coefficients with sequencies (0,0), (0,M), (M,0) and (M,M) are computed and placed in L3G1. The elements of SLR, DLR, DTB and DDiC in L2G1 are summed up to obtain the coefficients S(0,0), S(0,N-1), S(N-1,0) and S(N-1,N-1) of sequencies 00, 0M, M0 and MM respectively.

$$S(0,0) = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} SLR(i,j)$$

$$S(0,N-1) = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} DLR(i,j)$$

$$S(N-1,0) = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} DTB(i,j)$$

$$S(N-1,N-1) = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} DDiC(i,j)$$

L3G2

Sequency index (c_1, c_2) of the packets in L3G2 has 0 or M as one of the index and the other one varies as 2^l , where $l = 0$ to $(v-1)$ and $v = \log_2 M$. Thus, coefficients belonging to sequency packet groups $\{(0,1), (0,2), \dots, (0,2^{v-1})\}$, $\{(1,0), (2,0), \dots, (2^{v-1},0)\}$,

$\{(M,1), (M,2), \dots, (M,2^{v-1})\}$ and $\{(1,M), (2,M), \dots, (2^{v-1},M)\}$ are computed and placed in L3G2. The general algorithm for L3G2 is explained with the help of a flow chart shown in Fig. 6.2.

The algorithm depicted in the flowchart is further explained for sequency packet groups with index $c_1 = 0$, i.e.

$\{(0,1), (0,2), \dots, (0,2^{v-1})\}$.

The elements are placed in row 0. Initial row index of the element in SMRT matrix, n_1 is set as 0 and column index n_2 initialized as 1. Since two elements are computed in an iteration, n_2 increases by 2 after each iteration. The number of elements in a sequency packet is $\frac{M}{c_2}$. The number of additions for computation of each element depends on the sequency, c_2 . The matrices involved in the computation are CLS and CRS of L2G2. The computation steps in each iteration are as given in expressions 6.1 and 6.2 with k varying from 0 to $\frac{M}{2c_2}$.

$$S(n_1, n_2) = S(n_1, n_2) + CLS(k) - CLS(k + \frac{M}{2c_2}) \quad (6.1)$$

$$S(n_1, (n_2+1)) = S(n_1, (n_2+1)) + CRS(k) - CRS(k + \frac{M}{2c_2}) \quad (6.2)$$

The algorithm for sequency packet groups with $c_2 = M$ is identical, except the change in CLS & CRS to CLD & CRD and $n_1 = 0$ to $n_1 = N - 1$. Similarly, the computation for sequency packet groups with $c_1 = 0$ and $c_1 = M$ involves (RTS, RBS) & (RTD, RBD) combinations respectively and interchange of row & column indices in the expressions 6.1 and 6.2 for SMRT coefficient, S.

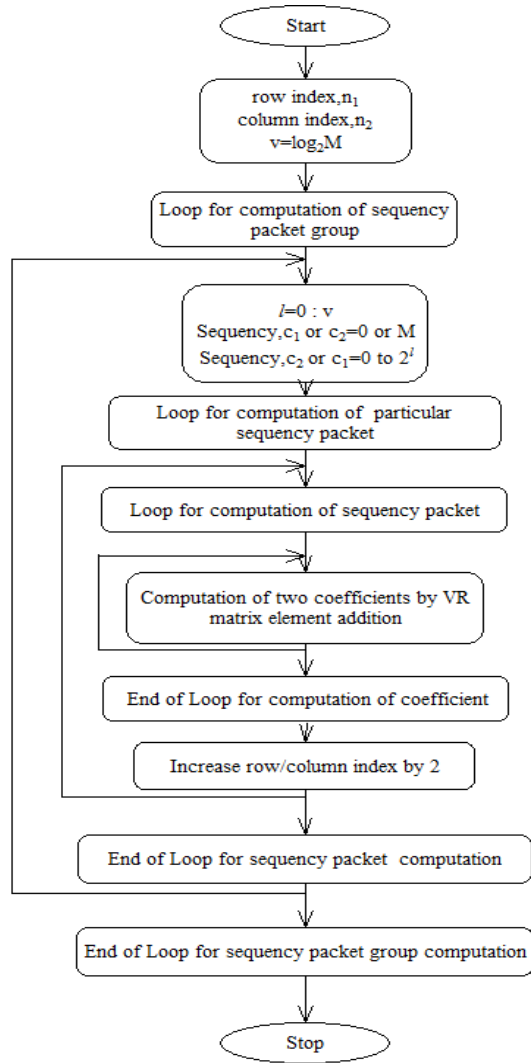


Fig. 6.2: Sequencies $\{(0,1), (0,2), \dots, (0,2^{v-1})\}$, $\{(1,0), (2,0), \dots, (2^{v-1},0)\}$, $\{(M,1), (M,2), \dots, (M,2^{v-1})\}$, $\{(1,M), (2,M), \dots, (2^{v-1},M)\}$

The four groups L3G3, L3G4, L3G5 and L3G6 in layer L3 are classified based on sequency combinations as (odd, odd), (odd,

even), (even, odd) and (even, even). Size of each sequency packet in all the groups is obtained as $\frac{M}{c_1} \times \frac{M}{c_2}$ [22]. The SMRT kernel expression 1.4.4 is used to compute the sign of the element of M-spacing matrix (C, D, E or F) involved in the computation of SMRT coefficient.

$$((r \times c_1(1 + 2 \times i_2) + c \times c_2))_N - c_1 \times i_1 = 0 \text{ or } M \quad (6.3)$$

In the expression 6.3, (i_1, i_2) represents the row and column indices of the coefficients within the packet with sequency (c_1, c_2) , having values, $i_1 = 0, 1, 2, \dots, \frac{M}{c_1} - 1$ and $i_2 = 0, 1, 2, \dots, \frac{M}{c_2} - 1$. (r, c) is the row and column indices of the data matrix, here the M-spacing matrix.

When $c_1 \leq c_2$, and equation 6.3 gives zero, the element of M-spacing matrix(C, D, E or F) involved in the computation of a particular SMRT coefficient is positive and is negative if the equation 6.3 gives M. If $c_1 > c_2$, then to find the sign, just interchange i_1 and i_2 in the equation 6.3.

The algorithm for computation and placement of elements in L3G3, L3G4, L3G5 and L3G6 can be generalized using the following steps.

- a. The indices of first row/column of SMRT coefficients corresponding to a sequency packet are to be initialized.
- b. The range of sequencies in the group are to be identified based on odd sequency refers to $2^0 = 1$ and even sequency is calculated as:

$$v = \log_2 M,$$

$$l = 1 : (v - 1),$$
 Even sequency varies as 2^l .
- c. The row and column size of a sequency packet are to be computed as $\frac{M}{c_1}$ and $\frac{M}{c_2}$ respectively.

-
- d. For $c_1 \leq c_2$, the sequency packets are computed column wise according to the following steps:
1. Sequency ratio, $m = \frac{c_2}{c_1}$ is calculated and the computation begins at the starting row of the sequency packet.
 2. A column identifier pointing to each column of the sequency packet is initialized as $k = 1$ to $\frac{N}{c_2}$ in steps of 2.
 3. The row and column indices of elements in M-spacing matrix are calculated as $r = 1$ and $c = M - \frac{M}{c_2} + 1$ respectively.
 4. If the column index c is less than 0, $c = c + M$ or if greater than $M - 1$, $c = c - M$.
 5. The sign of element of M-spacing matrix is found from the expression 6.3.
 6. The SMRT coefficient is computed by the addition of the element of M-spacing matrix with proper sign.
 7. Column index c is changed as $c = c - \frac{M}{c_2}$ after addition of each M-spacing element.
 8. Steps 4 to 7 are repeated until the selection of elements in each row of M-spacing matrix is completed.
 9. Row index r is changed as $r + m$ and column index c is changed as $c = c + \frac{M}{c_2} - k$. If the column index 2 is less than 0, $r = r + M$ or if greater than $M - 1$, $r = r - M$.
 10. Steps 4 to 9 are repeated until all elements of M-spacing matrix involved in the computation of a particular coefficient is completed.
 11. Change row index $r = r + k_{im}$, where k_{im} is the inverse mod of k with $\frac{M}{c_2} \times m$. Also check whether r is within the range of size of M-spacing matrix as in step 4.

12. Steps 4 to 11 are computed until all elements in a column of a sequency packet are computed.
 13. Change $n_2 = n_2 + 1$. Column identifier, $k = k + 2$. Steps 2 to 11 are repeated until all elements of a sequency packet are considered.
 14. Column index of new sequency packet is updated by adding the column size of present sequency packet. Steps from d is repeated until all the coefficients for sequency $c_1 \leq c_2$ are completed.
- e. For $c_1 > c_2$, the sequency packets are computed row wise according to the following steps.
1. Sequency ratio, $m = \frac{c_1}{c_2}$ and the computation starts at the first column of the sequency packet.
 2. Computation steps are similar to $c_1 \leq c_2$, except interchange of row and column.

The values of different variables in the algorithm for different groups is as shown in Table 6.1

Table 6.1: Variables used in the algorithm for different groups in L3G3 - L3G6

Group	Sequency	Sequencies c_1, c_2	Sequency ratio, m	Sequency Packet Size	First indices values n_1, n_2	M-spacing matrix
Group3	Odd,Odd	$c_1 = 1,$ $c_2 = 1$	1	M*M	1,1	F
Group4	Odd,Even	$c_1 = 1,$ $c_2 = 2^l$	c_2	$M*M/c_2$	1,1+M	D
Group5	Even,Odd	$c_1 = 2^l,$ $c_2 = 1$	c_1	M/c_1*M	1+M,1	E
Group6	Even,Even	$c_1 = 2^l,$ $c_2 = 2^l$	c_2/c_1 or c_1/c_2	$M/c_1*M/c_2$	1+M,1+M	C

The computational procedure described above for L3G3 to L3G6 are performed for individual groups with different parameters as explained below:

L3G3 - Odd Odd Sequency

SMRT coefficients belonging to Odd Odd sequency index are computed in L3G3. The size of sequency packet is $M \times M$, as the sequencies are $2^0 = 1$. Starting row index n_1 and column index n_2 are (1,1). Each coefficient is computed in M iterations using F matrix. The starting index of the element in the F matrix involved in the computation is (1,1).

L3G4 -Odd Even Sequency

Table 6.1 shows the values of variables to be initialized in L3G4. The starting row index, r of D matrix involved in the computation of SMRT coefficient, is initialized as 1 at the beginning of each sequency packet. The column identifier, k varies as $1, 3, \dots, \frac{N}{c_2}$ for the computation of $\frac{M}{c_2}$ columns in each sequency packet. Each column for L3G4 starts from the second row of SMRT matrix. c_2 elements in a row and $\frac{M}{c_2}$ elements in a column of D matrix is involved in the computation of an SMRT coefficient. At the start of the iteration, column index (c) of the element of D matrix is initialized as $M - \frac{M}{c_2} + 1$. The sign of the element is computed using equation 6.3. In each row iteration, the column index c is reduced by $\frac{M}{c_2}$ and at the end column index c is changed as $c + \frac{M}{c_2} - k$. The row index, r is increased by c_2 after each iteration. After completion of the computation of an SMRT coefficient, the row index is increased by the inverse mod of k, k_{im} , with respect to $\frac{M}{c_2}$.

L3G5-Even Odd Sequency

The algorithm for Even Odd Sequency using E matrix is very much similar to L3G4 except the interchange of c_1 & c_2 , n_1 & n_2 and r & c . Also, row index r is changed as $m + \frac{M}{c_1} - k_{im}$, where k_{im} is the inverse mod of k with respect to M after the end of row iteration.

L3G6 - Even Even Sequency

Table 6.1 shows the values of variables to be initialized in L3G6. When the sequency index $c_1 \leq c_2$, the algorithm is similar to that of L3G4 and when $c_1 > c_2$, it is similar to L3G5.

When $c_1 \leq c_2$, $\frac{M}{c_2}$ columns in each sequency packet are identified by the column identifier (k) varied as $1, 3, \dots, \frac{N}{c_2}$. Also, $\frac{M}{c_1}$ SMRT coefficients are to be computed in each column. For computing a coefficient, c_2 elements in a row and $\frac{M}{c_2} \times c_1$ elements in a column of M -spacing matrix C , are to be added. At the start of iteration, column index (c) of element of matrix C is initialized as $M - \frac{M}{c_2} + 1$. In each row iteration, the column index c is reduced by $\frac{M}{c_2}$. After the end of row iteration, column index c is changed as $c + \frac{M}{c_2} - k$. The row index, r is increased by $m = \frac{c_2}{c_1}$ after each iteration. After the completion of computation of an SMRT coefficient, the row index is increased by k_{im} with respect to $\frac{M}{c_2}$.

When $c_1 > c_2$, $\frac{M}{c_1}$ rows in each sequency packet is identified by the row identifier (k) varied as $1, 3, \dots, \frac{N}{c_2}$. c_1 elements in a column and $\frac{M}{c_1}$ elements in a row of M -spacing C matrix are to be added for the computation of an SMRT coefficient. Row index (r) of the element of matrix C is initialized as $M - \frac{M}{c_1} + 1$, at the beginning of the iteration. In each column iteration the row index r is reduced by $\frac{M}{c_1}$. After the end of row iteration, row index r is changed as $r + \frac{M}{c_1} - k_{im}$, where k_{im} is the inverse mod of k with respect to $\frac{M}{c_1}$. The column index (c) is increased by m after each iteration.

Inverse Modulo Operation

The inverse modulo operation is to be used in the algorithm when the sequencies c_1 and c_2 are not equal. When $c_1 \leq c_2$, after the completion of a column iteration of sequency packet and value of k is expected as the initialization value of r in M -spacing matrix for the next iteration. But its actual value is $k_{im} =$ inverse modulo of k with respect to $\frac{M}{c_2}$. When $c_1 > c_2$, column index will vary with inverse modulo of k with respect to $\frac{M}{c_1}$. The values of k and k_{im} is shown in figure 6.3. The definition and computation of inverse modulo is explained in Appendix E.

k	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31
$k_{im}(M=4)$	1	3														
$k_{im}(M=8)$	1	3	5	7												
$k_{im}(M=16)$	1	11	13	7	9	3	5	15								
$k_{im}(M=32)$	1	11	13	23	25	3	5	15	17	27	29	7	9	19	21	31

Fig. 6.3: k and k_{im} values

6.2.2 Matlab Simulation

Simulation of hybrid computation approach of $N \times N$ SMRT, for N a power of 2 developed in section 6.2 is performed using MATLAB 8.5 (R2015a) software package on Intel core i5, 2.4 GHz, 4 GB RAM. The comparison of execution time of the hybrid algorithm based computation and direct SMRT computation [22] is given in Table 6.2. The results show that as the matrix size increases, direct SMRT algorithm becomes time consuming.

The computation time varies between the hybrid and direct approaches varies in the order of powers of 10.

Table 6.2: Computation time for direct SMRT and SMRT Hardware Algorithms

Size	Time in seconds	
	Hybrid Approach	Direct SMRT
2x2	1.59×10^{-4}	2.54×10^{-4}
4x4	1.88×10^{-4}	1.86×10^{-4}
8x8	2.2×10^{-3}	2.23×10^{-4}
16x16	1.62×10^{-2}	4.1×10^{-2}
32x32	4.99×10^{-2}	5.64×10^{-2}
64x64	2.27×10^{-1}	8.13×10^{-1}
128x128	1.09×10^0	1.29×10^1
256x256	5.45×10^0	2.09×10^2
512x512	1.94×10^1	3.34×10^3
1024x1024	2.54×10^2	7.98×10^3

6.2.3 FPGA Implementation

The parallel architecture is implemented using Xilinx package, presented in section 5.7, for the different size of matrices. The results of hardware implementation is as shown in Table 6.3.

From the Table 6.3, it is clear that the time of computation is very much reduced as the data size increases. But the resource requirements increase with the size of the data matrix.

Table 6.3: Comparison of FPGA Implementation for Matrices of different size

Performance Factors	4x4	8x8	16x16	32x32	64x64
No. of Adders/ Subtractors	66	190	543	728	1382
No. of Counters	2	2	2	2	2
No. of Latches	15	45	125	370	725
No. of slice LUTs	546	1548	4856	9898	14486
Max. Combinational Path delay(ns)	16.715	16.72	16.724	16.83	17.21

The comparison of implementation of the hybrid architecture for 8×8 SMRT discussed in section 5.6 using equations with the 8×8 SMRT using general hybrid architecture implementation here is tabulated as shown in Table 6.4

Table 6.4: Comparison of FPGA Hybrid Architecture Implementation of 8×8 SMRT

	Hybrid Algorithm	
	Based on equations	General Algorithm
Performance Factors		
No. of Adders/ Subtractors	651	190
Cell Usage- BELS	14237	4145
No. of slice LUTs	5085	1548
Max. Combinational Path delay (ns)	19.193	16.72
Logic Delay (ns)	10.005	9.447
Routing Delay (ns)	9.188	7.273

The hybrid $N \times N$ SMRT algorithm developed for N a power of 2 can be used for a variety of computations for image processing

applications. In the direct SMRT algorithm [22], if a particular sequency packet has to be computed, the whole SMRT matrix is to be calculated and the coefficients are to be isolated from the packet based on the row/column indices. But the hybrid algorithm can be easily modified to compute particular sequency packets or a particular set of coefficients, without computing the whole SMRT matrix.

6.3 Software Hardware Co-development

The ease of implementation of the software hardware co-developed system is suggested for the real time implementation of texture analysis problems. The general block diagram of such a system is as shown in Fig. 6.4

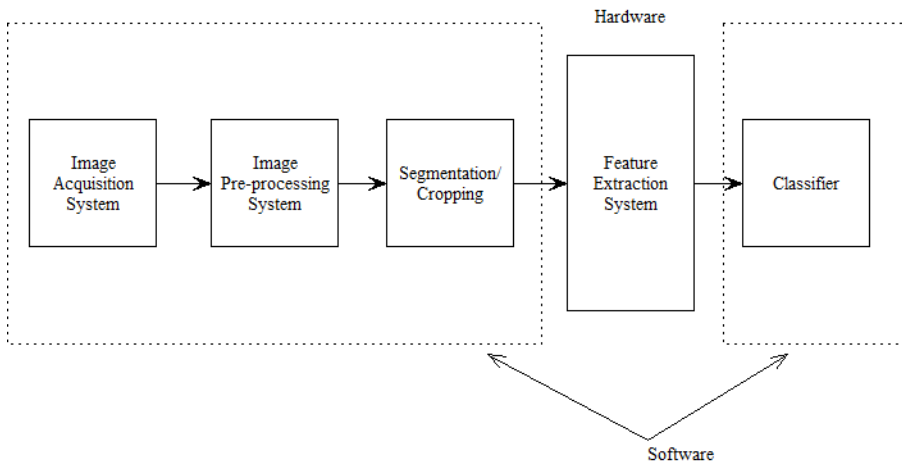


Fig. 6.4: General Block Diagram of Software Hardware Co-developed System

The tasks performed by software part in co-development are, capture the image using some imaging techniques, pre-process the

image if necessary, segment or locate ROI, crop an $N \times N$ image, N a power of 2. The cropped image is given as input to the SMRT texture feature extraction hardware system and the output of the hardware is given to a classifier implemented in software. This system can be custom made for all SMRT texture feature based classification.

6.3.1 Prostate Disease Diagnosis using Software Hardware Co-development

Block diagram of the software hardware co-developed system for prostate disease diagnosis based on texture analysis using SMRT features is shown in Fig. 6.5.

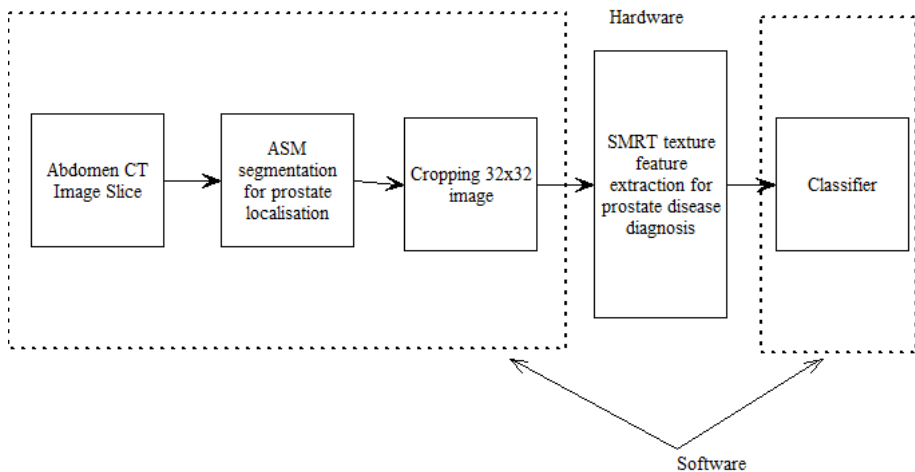


Fig. 6.5: Co-developed System for Prostate Disease Diagnosis

Slice of abdomen CT image is given as input to ASM segmentation software, to extract the prostate region. 32×32 image is cropped from the prostate region and given as input to the SMRT texture feature extraction hardware. The module output

is given to classifier implemented in software, which identifies the prostate diseases.

A similar hardware can be developed for skin cancer detection and coconut growth stage identification. In each case the general algorithm developed for $N \times N$ SMRT has to be modified accordingly. The implementation of modified SMRT algorithm for SMRT Texture Feature Extractor for Prostate Disease Diagnosis is explained in the next subsection.

6.3.2 Hardware Implementation of SMRT Texture Feature Extractor for Prostate Disease Diagnosis

The details of optimized SMRT texture features used for prostate disease diagnosis, explained in section 4.2.3, is given in Table 6.5. The number of features belongs to a particular sequency and the number of coefficients involved in the computation of that feature is depicted in the table.

The hybrid algorithm discussed in section 6.2.1 is modified to compute the required rows and columns in the sequency packets for the computation of features.

Computations based on 2×2 data are not required in the texture feature computation, as seen from the table. The sequency groups involved in the computation of features are (odd, odd), (odd, even), (even, odd) and (even, even) as in the Table 6.5. Hence, the hybrid algorithm can be modified as discussed below:

Table 6.5: GA optimized SMRT Texture feature set

Sequency Group	Sequency	No.of fea- tures	No. of co- efficients	row/column identifier,k
Odd Odd	1,1	3	48	1,3,11
Odd Even	1,2	3	48	3,5,9
Odd Even	1,4	1	16	7
Odd Even	1,8	1	16	1
Even Odd	2,1	2	32	9,11
Even Odd	4,1	3	48	1,3,5
Even Odd	8,1	1	16	3
Even Even	2,2	2	16	3,13
Even Even	2,4	1	8	1
Even Even	2,8	1	8	3
Even Even	4,2	1	8	3
Even Even	8,2	1	8	1
Even Even	4,8	1	4	3
	Total	21	276	

In layer L1, only L1G3 operations required to find A and B matrices (32×16), are to be performed. Similarly, only L3G4 computations to find M-spacing matrices: C, D, E and F, are to be done in layer L2. In layer L3, only few row/columns of sequency packets in each groups L3G3, L3G4, L3G5 and L3G6 are to be computed to get the SMRT texture features for prostate disease diagnosis. The layer L3 algorithm is implemented in the following steps.

- a. SMRT texture feature vector with 21 features is to be initialized. The sequencies in each group are to be identified, as given in Table 6.5.
- b. If $c_1 \leq c_2$, the sequency packets are computed as column wise and absolute value of each coefficient in a column is to be added to obtain texture features, according to the following

steps:

1. Sequence ratio, m is calculated as $\frac{c_2}{c_1}$ and the computation starts at starting row of the sequence packet.
2. A column identifier which points to the required columns of the sequence packet, given in Table 6.5 is initialized as row vector k .
3. The row and column indices of M-spacing matrix elements are to be calculated, as $r = 1$ and $c = M - \frac{M}{c_2} + 1$ respectively.
4. If the column index c is less than 0 or greater than $M-1$, $r+M$ or $r-M$.
5. The sign of M-spacing element is obtained from the expression 6.3.
6. The SMRT coefficient is computed by the addition of the element of M-spacing matrix with proper sign.
7. After addition of each M-spacing element, column index c is changed as $c = c - \frac{M}{c_2}$.
8. Steps 4 to 7 is repeated until all elements in row of M-spacing matrix involved in a particular coefficient computation is added.
9. Row index r is changed as $r + m$ and column index c is changed as $c = c - \frac{M}{c_2} - k$. Also check whether r is within the range of M-spacing matrix size, else adjust with M ($r+M$ or $r-M$).
10. Steps 4 to 9 is repeated until all elements of M-spacing matrix involved in a particular coefficient computation is added.
11. Absolute value of the coefficient is added to obtain the texture feature.

-
12. Row index r is changed as $r + k_{im}$, where k_{im} is the inverse mod of k with $\frac{M}{c_2} \times m$. Also check whether r is within the range of M-spacing matrix size, else adjust with M. Also the row index of SMRT coefficient increased by one after each iteration.
 12. Steps 4 to 11 is computed until all elements in a column of a sequency packet is computed and added to obtain a texture feature.
 13. Feature vector index is increased by 1 after the completion of one iteration. The next column identifier is taken from the column identifier vector. Steps 2 to 11 is repeated until all columns identifiers in a sequency packet is used.
 14. Column index of new sequency packet is computed by adding the column size of present sequency packet. Steps from d is repeated until all the features in sequency packets with $c_1 \leq c_2$ are computed.
- e. If the $c_1 > c_2$, features in the sequency packets are computed as row wise and hence the texture features are obtained by addition of the absolute value of SMRT coefficients in a row according to the following steps.
1. The computation starts at the first column of the sequency packet.
 2. Computation steps are almost similar to $c_1 \leq c_2$. The difference is that instead of column identifier vector, row identifier vector is to be initialized. All the row steps in $c_1 \leq c_2$ computations are column steps in $c_1 > c_2$ computations. In all calculations, c_1 is to be replaced by c_2 and vice versa. All r,c values are to be replaced by c,r values.

In the algorithm, row/ column identifier vector has different size

depending on the number of features in each sequency packet. For example, in odd odd sequency group, the column identifier vector is initialized as $\{1, 3, 11\}$ corresponding to the three features in the sequency packet given in Table 6.5.

The optimized set of coefficients are to be identified and extracted from the 94 features computed using direct SMRT computation algorithm based on MRT [22] for 32×32 image. Here, the hybrid algorithm is modified to compute only the coefficients to obtain the required feature set and hence reducing the computational time. The computation time of texture feature sets for three different cropped prostate images using the hybrid algorithm and direct SMRT algorithm is shown in Table 6.6.

Table 6.6: Computation time comparison of SMRT Texture features using Hybrid Algorithm and Direct SMRT algorithm

Image	Hybrid algorithm	Direct SMRT algorithm
	Computation Time in sec	
1	3×10^{-3}	1.8×10^{-2}
2	1.2×10^{-3}	2×10^{-2}
3	2.3×10^{-3}	3.2×10^{-2}

The computation time of features, as seen from the Table 6.6, is reduced to approximately $1/30^{th}$.

The above algorithm is implemented in Xilinx ISE 14.7, target device is Virtex 5 FPGA- XC5VLX30 and the hardware language used is VHDL.

Table 6.7: FPGA Implementation of SMRT texture feature extraction for Prostate Disease Diagnosis

Performance Factors	
No. of Adders/Subtractors	465
No. of Counters	2
No. of Latches	311
No. of slice LUTs	8798
Max. Combinational Path delay(ns)	15.62

The results tabulated in Table 6.7 show that the hardware implementation reduces the computation time from milliseconds to nanoseconds. It clearly shows that the time for disease diagnosis gets drastically reduced through the proposed software hardware co-developed system.

The algorithms developed for computation of 2-D SMRT is faster compared to that of direct SMRT. 1-D SMRT is also used in many signal processing applications similar to 2-D SMRT. Hence, an attempt is made to improve the computational speed of 1-D SMRT with a similar approach as that for 2-D SMRT.

6.4 Forward and inverse N-point SMRT, N a power of 2

The concept of 1-D MRT is explained in [18]. An input vector of size N has N^2 MRT coefficients. Many of the coefficients appear more than once either in the same form or with a change in sign. Hence, there is much redundancy and can be eliminated. 1-D UMRT [18] is formed by eliminating redundancy in 1-D MRT and placing the unique coefficients. The elements of 1-D UMRT can be rearranged based on sequency to obtain the 1-D SMRT. Here visual representation of 1-D SMRT coefficients is to be analyzed

in a similar way as 2-D SMRT and an algorithm can be developed for the computation of 1-D SMRT based on the patterns present in the visual representation.

The sequency distributions in 1-D SMRT for $N= 4, 8$ and 16 is as shown in figure 6.6.

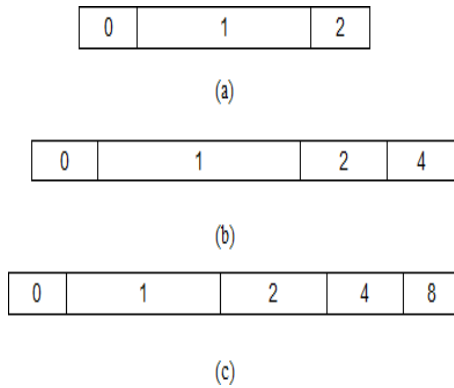


Fig. 6.6: Sequency distribution in 1-D SMRT for $N=4,8,16$

The sequency c in a 1-D SMRT vector varies as $0, 2^0, 2^1, \dots, 2^v$, where $v = \log_2 M$, $M = \frac{N}{2}$. There will be only one element for zero sequency. For all other sequencies the packet size is computed as $\frac{M}{c}$. The visual patterns of SMRT for $N=4$ and 8 are as shown below, Fig. 6.7 and 6.8.

+	+	+	+
Sequency 0			
+		-	
Sequency 1			
	+		-
Sequency 1			
+	-	+	-
Sequency 2			

Fig. 6.7: Sequency Pattern in 1-D SMRT for N=4

+	+	+	+	+	+	+	+
sequency 0							
+				-			
	+				-		
		+				-	
			+				-
sequency 1							
+		-		+		-	
	+		-		+		-
sequency 2							
+	-	+	-	+	-	+	-
sequency 4							

Fig. 6.8: Sequency Pattern in 1-D SMRT for N=8

6.4.1 M-spacing Data for N-point SMRT

The theorem discussed in section 5.5, on M-spacing data in the 2-D SMRT coefficient can be derived to 1-D SMRT. The theorem

is modified for 1-D SMRT as:

If data at n is involved in the computation of a particular SMRT coefficient, data at $n+M$ location will also contribute to that coefficient. Based on this two vectors A and B are defined, which are the sum and difference of data elements M distance apart, as in Fig. 6.9.



Fig. 6.9: A and B Matrices

An algorithm is developed for the computation of 1-D SMRT coefficients based on M -spacing data availability,.

6.4.2 Forward N-point SMRT Algorithm

Let X be the data vector and S be the SMRT vector, of size N . The coefficients are computed in the order of sequency.

1. M -spacing vectors A and B are computed.

$$M = \frac{N}{2}$$

for $i= 0$ to $M-1$

$$A(i) = x(i) + x(i + M)$$

$$B(i) = x(i) - x(i + M)$$

end

2. Computation of DC SMRT coefficient.

$r=0$.

$$S(r) = \sum_{i=0}^{M-1} A(i).$$

$r=r+1$.

3. *Computation of SMRT coefficients in packet with sequency,*
 $c=1$
 $v = \log_2 M$
 for $i = 1$ to $v-1$
 $S(r) = B(i)$
 $r = r + 1$
 end

4. *Computation of SMRT coefficients in other sequency packets*
 for $l= 1$ to v
 sequency, $c= 2^l$
 Sequency packet size, $q = \frac{M}{c}$
 for $m= 1$ to q
 for $k= 1$ to $\frac{c}{2}$
 $S(r) = A(m) - A(m + q)$
 end
 $r = r + 1$
 end

Table 6.8: Comparison of Execution time of 1-D SMRT Algorithms

Sl. No.	Vector size	M-spacing based algorithm	MRT based algorithm
		microsecs	
1	4	66×10^{-0}	21.1×10^1
2	8	98×10^0	37.3×10^1
3	16	10.5×10^1	10.65×10^2
4	32	13.6×10^1	14.20×10^2
5	64	16.5×10^1	25.76×10^2
6	128	22.1×10^1	55.05×10^2
7	256	34.6×10^1	98.84×10^2
8	512	59.8×10^1	24.900×10^3
9	1024	11.36×10^2	71×10^3
10	2048	24.8×10^2	16.6×10^4

The algorithm is implemented in MATLAB and is compared with the algorithm based on MRT [190] in terms of execution time.

From the Table 6.8, it is clear that M-spacing based algorithm is on an average 25 times faster than the earlier MRT based algorithm. An inverse algorithm is also developed based on the visual representation of 1-D SMRT coefficients.

6.4.3 Inverse N-point SMRT Algorithm

Fig. 6.7 is analyzed to derive the inverse SMRT vector for $N=4$. Let X be the data vector and S be the SMRT vector, of size N . For $N=4$, SMRT vector is $S=[s(1), s(2), s(3), s(4)]$ and data vector is $X=[x(1), x(2), x(3), x(4)]$. Analyzing the visual representation, the elements of data vector is derived as:

$$x(1) = \frac{s(1) + 2s(2) + s(4)}{4} \quad (6.4)$$

$$x(2) = \frac{s(1) + 2s(3) - s(4)}{4} \quad (6.5)$$

$$x(3) = \frac{s(1) - 2s(2) + s(4)}{4} \quad (6.6)$$

$$x(4) = \frac{s(1) - 2s(3) - s(4)}{4} \quad (6.7)$$

Similarly, analysis of visual pattern for $N=8$, Fig. 6.8, gives the data elements as:

$$x(1) = \frac{s(1) + 4s(2) + 2s(6) + s(8)}{8} \quad (6.8)$$

$$x(2) = \frac{s(1) + 4s(3) + 2s(7) - s(8)}{8} \quad (6.9)$$

$$x(3) = \frac{s(1) + 4s(4) - 2s(6) + s(8)}{8} \quad (6.10)$$

$$x(4) = \frac{s(1) + 4s(5) - 2s(7) - s(8)}{8} \quad (6.11)$$

$$x(5) = \frac{s(1) - 4s(2) + 2s(6) + s(8)}{8} \quad (6.12)$$

$$x(6) = \frac{s(1) - 4s(3) + 2s(7) - s(8)}{8} \quad (6.13)$$

$$x(7) = \frac{s(1) - 4s(4) - 2s(6) + s(8)}{8} \quad (6.14)$$

$$x(8) = \frac{s(1) - 4s(5) - 2s(7) - s(8)}{8} \quad (6.15)$$

After studying the expressions for data elements for $n=4$ and 8 , the following conclusions are made: $v = \log_2 M$

No. of iterations, $l = 0$ to M ;

Table 6.9: Sign changes and multiplication terms based on sequency in 1-D SMRT inverse computation

Sequency	0	2^v
sign change	No sign changes	Sign changes after $2^{(v-l)}$ alterations
multiplication factor	1	$2^{(v-l)}$

Based on the above observations an algorithm is developed for the computation of 1-D inverse SMRT, as explained in the preceding section.

Inverse N-point SMRT Algorithm

Let S be the SMRT vector and X be the data vector, of size N.

1. $s(1)$ is added to obtain all data elements.

```

for i=1:N
   $x(i) = x(i) + s(1)$ 
end

```

2. $s(n)$ is added to obtain all data elements with proper sign.

```

for i=1:N
  if mod(i,2)=0
    p=-1
  else
    p=1
  end
   $x(i) = x(i) + p \times s(n)$ 
end

```

3. Other coefficients are added to X_i with proper sign and multiplication factor in $v \times N$ iterations.

```

 $q1 = 0, n = N, p = 1, v = \log_2 M$ 
for  $l = 1 : v$ 
  Sequency,  $c=2^{(v-l)}$ 
  Multiplication Factor,  $r= 2^l$ 
  Sequency packet size,  $q=\frac{M}{c}$ 
   $q1 = q1 + q$ 
   $n = n - q1$ 
  for  $i = 1 : N$  if mod(i,r)=0,
     $p = -p$ 
     $n = n + 1 - q$ 
  else
     $n = n + 1$ 

```



```

end
 $x(i) = x(i) + p \times r \times s(n)$ 
end
end

```

The execution time of the inverse N-point SMRT algorithm for different vector size is computed by MATLAB implementation and is as given in Table 6.10

Table 6.10: Execution time of Inverse 1-D SMRT

Sl. No.	Vector size	Time taken
1		secs
2	4	6.3×10^{-5}
3	8	11.4×10^{-5}
4	16	12.9×10^{-5}
5	32	16.8×10^{-5}
6	64	24.2×10^{-5}
7	128	40.4×10^{-5}
8	256	64.9×10^{-5}
9	512	11×10^{-4}
10	1024	35.9×10^{-4}
11	2048	79×10^{-4}

Faster computation in 1-D forward and inverse SMRT is achieved using the visual representation based algorithm. Hence the algorithms can be used for various signal processing applications.

6.5 Conclusion

The architecture developed for 2-D SMRT implemented in the hardware, can be considered as a fast algorithm in software also. In the original 2-D SMRT algorithm as the size of the input vector increases, the time for computation increases many fold. The

reason for this increase of time is that in the algorithm MRT coefficients are to be computed, which involves data search. But in the new algorithm 2-D SMRT coefficients are computed directly by adding or subtracting data elements. So as the matrix size increases, time of computation is very much less compared to the early algorithm. So to analyze large 2-D data, the new algorithm will be more time efficient. For example, to analyze a 512x512 image, the early algorithm will take 3354 seconds whereas the same can be achieved in 19 seconds using the new algorithm. There is also considerable reduction in time of computation for 1-D forward and inverse SMRT algorithms when compared to the early algorithm. The beauty of the algorithm lies in the ability of isolating the computation of a particular set of coefficients or a particular sequency packet.

Chapter **7**

Results, Conclusion and Future Scope

Contents

7.1	Summary	172
7.1.1	GA Optimization of SMRT based Texture Features	172
7.1.2	SMRT based Texture analysis of Medical and Non Medical Images	173
7.1.3	Development of Parallel Distributed Architecture for 8×8 SMRT	173
7.1.4	Development of Parallel Distributed Architecture for SMRT	174
7.2	Research Contributions	174
7.3	Scope for Future Work	175

Summary of the investigations carried out and the conclusions drawn from the investigations are presented in this chapter. Major contributions and future scopes of work are also discussed.

7.1 Summary

The work is introduced in chapter 1 and the status of the related works in the literature are discussed in chapter 2. The essence of the research work is presented in chapters 3 to 6. SMRT based texture features and their optimization are introduced in chapter 3. Chapter 4 presents the application of SMRT based texture analysis in medical and non medical field. Development of parallel distributed architectures for 8×8 SMRT is presented in chapter 5. Chapter 6 explains the development of a parallel distributed architecture for $N \times N$ SMRT, N a power of 2, and other related works for software hardware co-development of SMRT based texture analysis of images.

7.1.1 GA Optimization of SMRT based Texture Features

SMRT texture features, simple and faster in computation, are developed in the same line as that of UMRT texture features. The SMRT texture feature extraction is six times faster than UMRT texture feature extraction. The SMRT texture feature set is optimized using genetic algorithm and K-NN classifier to obtain good classification accuracy with minimum number of features. The performance of optimized subset developed for Brodatz image classification is evaluated. Classification accuracy of GA optimized SMRT texture features is more when compared with tree structured wavelet transform based texture features.

7.1.2 SMRT based Texture analysis of Medical and Non Medical Images

SMRT based texture analysis of medical and non medical images is performed. In this work two deadly diseases, prostate cancer and skin cancer, are studied. Texture analysis of abdomen CT images is done in order to diagnose prostate diseases. Prostate gland is difficult to identify in images, as the prostate tissue does not vary much from surrounding tissues and hence, there is no border to distinguish the prostate gland in images. Active shape model segmentation is used which automatically segment the prostate gland in the image. Optimized SMRT texture feature set is computed from the segmented image. Prostate diseases are classified with very good classification accuracy based on optimized 32×32 SMRT texture feature subset with 21 features. The SMRT based texture analysis method is extended for malignant melanoma diagnosis using dermoscopic images. Images of malignant melanoma and other skin diseases are classified with 100% accuracy using eight 16×16 SMRT texture descriptors. The method developed for classification of medical images is extended for non medical images also. Here texture analysis is performed to identify the maturity level of coconut to decide on harvesting from photographs of coconut bunches.

7.1.3 Development of Parallel Distributed Architecture for 8×8 SMRT

Parallel distributed architectures are developed for 8×8 SMRT, as a first step in developing a software hardware co-development for texture analysis of images using SMRT features. The architectures are developed by analyzing the visual representation of SMRT coefficients based on 2×2 data. Three different architectures are developed: 1) based on 2×2 data, 2) based on M-spacing

data and 3) a hybrid architecture as combination of 2×2 data and M-spacing data. Performance of the three architectures are compared by implementing them in software and hardware.

7.1.4 Development of Parallel Distributed Architecture for SMRT

Parallel distributed model is developed for $N \times N$ SMRT, N a power of 2. The model is implemented as a three layer distributed architecture and the algorithm is implemented in both software and hardware. In software, the algorithm works faster than the direct SMRT [22] which involves modulus operation based data search. The algorithm is modified to compute the texture features for prostate disease diagnosis and implemented in FPGA hardware.

An algorithm for the computation of forward and inverse N -point SMRT, N a power of 2, is developed as a by-product of the generalized hardware algorithm for $N \times N$ SMRT. The algorithm for N -point SMRT is faster compared to the existing algorithm [190].

7.2 Research Contributions

Major contributions of the research work presented in the thesis are:

- Developed SMRT based texture features and used Genetic Algorithm optimization along with K-NN classifier as tool for Feature selection in SMRT based texture analysis
- Developed SMRT based texture analysis from CT images for Prostate Disease diagnosis.

- Verified the scope of SMRT based texture analysis method in skin cancer detection.
- Verified the scope of SMRT based texture analysis method in Non medical image processing through growth stage identification of Coconut from photographic images.
- Developed Parallel Distributed Architecture for 8x8 SMRT and is generalized for NxN SMRT, N a power of 2. They are implemented in both hardware and software.
- Developed Parallel distributed architecture for texture features for prostate disease diagnosis for software hardware co-design.
- Developed the algorithm for computation of forward and inverse N-point SMRT, N a power of 2, as a by-product of parallel distributed computation approach.

7.3 Scope for Future Work

The present work established the usefulness of 2D- SMRT for texture analysis of few medical and non-medical images. The possibility of 2-D SMRT for image analysis can be extended to many other forms of images like SAR images, hyper spectral images etc. Scope of 2-D SMRT in other signal processing applications can also be explored.

GA based local optimization technique is used in the work for optimizing SMRT texture features. Research can be continued to develop a technique for global optimization of texture features. The feasibility of adaptive optimization technique for global optimization of texture features can also be explored.

The present work mainly centers around 2D-SMRT with a size

as power of 2. The studies shall be extended to any size N . The parallel distributed architecture developed for 2D-SMRT computation as well as SMRT texture computation hardware can be implemented in ASIC based nano-technology.

The software hardware co-developed system can be designed and analyzed for any given application.

Also the transform can be extended for any multidimensional signals.

An efficient algorithm for 1-D SMRT computation is developed in the present work. This can be used for a number of signal processing applications.

List of Publications

International / National Conferences

- Manju B, K. Meenakshy, R. Gopikakumari, Optimum Selection of 8x8 UMRT based Texture Descriptors using Genetic Algorithm, *In proceedings of National Conference on Recent Trends in Electrical and Electronics Engineering* , NCRTEEE 2013, Jerusalem College of Engineering, Chennai, pp. 111-114, 13 May 2013.
- Manju B, K. Meenakshy, R. Gopikakumari, Prostate Disease Diagnosis from CT images using GA optimised SMRT based Texture Features, *In Procedia Computer Science* , ICICT 2014.

Refereed Journals

- Manju B, Jaya V.L., K. Meenakshy, R. Gopikakumari, 8x8 SMRT based Texture Descriptors, "Lecture Notes on Software Engg," Vol. 3, No. 4, pp. 295-298, November 2015.
- Manju B, Jaya V.L., K. Meenakshy, R. Gopikakumari, Comparison of SMRT and Wavelet Texture Features, *International Journal of Engineering and Future Technology*, Volume 14, Issue No. 3, pp. 14-22, Year 2017.
- B. Manju, R. Gopikakumari, Development of Parallel Distributed Architecture for 8 x 8 Sequency based Mapped Real Transform, *Journal of The Institution of Engineers(India): Series B* , communicated.

Bibliography

- [1] A.K.Jain, *Fundamentals of digital image processing*. Prentice-Hall, New Delhi, 2003.
- [2] S. Jayaraman, S. Esakkirajan, and T. Veerakumar, *Digital Image Processing*. McGraw Hill Education (India) pvt Ltd., 2009.
- [3] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Addison Wesley, 1992.
- [4] M. Tabb and N. Ahuja, “Multiscale image segmentation by integrated edge and region detection,” *IEEE Trans. Image Process.*, vol. 6, no. 5, pp. 642–655, 1997.
- [5] M. Leventon, W. Grimson, and O. Faugeras, “Statistical shape influence in geodesic active contours,” *5th IEEE EMBS Int. Summer Sch. Biomed. Imaging, 2002.*, pp. V_3_1–V_3_8, 2000.
- [6] T. F. Cootes, “Statistical Models of Appearance for Computer Vision,” *Report*, no. 1, pp. 1–124, 2004.

- [7] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," *Proc. fourth ACM Int. Conf. Multimed. (MULTIMEDIA '96)*, pp. 1–14, 1998.
- [8] R. Venkata Ramana Chary, "Feature Extraction Methods For Color Image Similarity," *Adv. Comput. An Int. J.*, vol. 3, no. 2, pp. 147–157, 2012.
- [9] A. J. Afifi and W. M. Ashour, "Image Retrieval Based on Content Using Color Feature," *ISRN Comput. Graph.*, vol. 2012, pp. 1–11, 2012.
- [10] A. Materka and M. Strzelecki, "Texture Analysis Methods – A Review," *Methods*, vol. 11, pp. 1–33, 1998.
- [11] R.M.Haralick, K.Shanmugham, and I.Dinstein, "Textural features for image classification," *IEEE Trans. Syst. Cybern.*, vol. SMC 3, pp. 610–621, 1973.
- [12] R. M. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE*, vol. vol. 67, pp. 786–804, 1979.
- [13] M.M.Galloway, "Textural Analysis using gray level run lengths," *Comput. Graph. image Process.*, vol. 4, pp. 172–179, 1975.
- [14] J.S. Weszka, C. R. Dyer, and A. Rosenfeld, "A comparative study of texture measures for terrain classification," *IEEE Trans. Syst. Man Cybern.*, vol. vol. SMC-6, pp. 269–285, 1976.
- [15] R. Gopikakumari, *Investigations on the Development of an ANN Model and Visual Manipulation Approach for 2-D DFT Computation in Image Processing*. PhD thesis, Cochin University of Science and Technology, Kochi, India, 1998.
- [16] R. C. Roy and R. Gopikakumari, "A new transform for 2D signal representation (MRT) and some of its properties,"

Proceedings of International Conference on Signal Processing and Communications, no. 3, pp. 363–367, 2004.

- [17] V. Bhadran, *Development and implementation of visual approach and parallel distributed architecture for 2-D DFT and UMRT computation*. Ph.D dissertation, Cochin University of Science and Technology, Kochi, 2009.
- [18] R. C. Roy and R. Gopikakumari, *Development of a new Transform: MRT*. PhD thesis, 2009.
- [19] K. Meenakshy, *Development and Implementation of a CAD System to predict the Fragmentation of Renal Stones Based on Texture Analysis of CT Images*. Ph.D dissertation, Cochin University of Science and Technology, Kochi, Kochi, 2010.
- [20] V. L. Jaya, P. Basu, and R. Gopikakumari, “SMRT: A new placement approach of 2-D unique MRT coefficients for N a power of 2,” *Annu. IEEE India Conf. INDICON*, pp. 233–237.
- [21] V. L. Jaya and R. Gopikakumari, “Sequency Based Mapped Real Transform: Properties and Applicatons,” *SIViP (2017)*, pp. 1551–1558, 2017.
- [22] V. L. Jaya, *Development of $N \times N$ SMRT for N a power of 2 and its Applications in Image Enhancement*. PhD thesis, Cochin University of Science and Technology, Kochi, India, 2015.
- [23] R. Cherian Roy, M. S. Anish Kumar, and R. Gopikakumari, “An invertible transform for image representation and its application to image compression,” *Proc. Int. Symp. signal Process. its Appl. 2007*), vol. 24, pp. 1–4, 2007.
- [24] M. S. AnishKumar, B. Preetha, and R. Gopikakumari, “UMRT based Adaptive Block Size Transform Coder for

- Images Using Quad-tree partitioning,” *Int. J. Sci. Eng. Res.*, vol. 4, no. 12, pp. 1170–1176, 2013.
- [25] K. De Jong, “Learning with genetic algorithms : An overview,” tech. rep., Machine Learning, Vol. 3, Kluwer Academic Publishers, 1988.
- [26] D. Goldberg, *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.
- [27] F. C. H. Rhee and Y. J. Lee, “Unsupervised feature selection using a fuzzy-genetic algorithm,” *IEEE International Fuzzy Systems Conference Proceedings*, pp. 1266–1269, 1999.
- [28] H. Frohlich, “Feature selection for support vector machines by means of genetic algorithm,” *Proceedings. 15th IEEE Int. Conf. Tools with Artif. Intell.*, no. December, pp. 142–148, 2002.
- [29] F. Tan, X. Fu, Y. Zhang, and A. G. Bourgeois, “A genetic algorithm-based method for feature subset selection,” *Soft Comput.*, vol. 12, no. 2, pp. 111–120, 2008.
- [30] K. Fukunaga, “Statistical Pattern Recognition,” *Pattern Recognit.*, vol. 22, no. 7, pp. 833–834, 1990.
- [31] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. 2nd Edition, John Wiley Publishers, 2004.
- [32] K. P. Murphy, “Naive Bayes classifiers Generative classifiers,” *Bernoulli*, vol. 4701, no. October, pp. 1–8, 2006.
- [33] S. Gunn, “Support Vector Machines for Classification and Regression,” *ISIS Tech. Rep.*, no. November, pp. 1–42, 1997.
- [34] F. Eddaoudi, F. Regragui, A. Mahmoudi, and N. Lamouri, “Masses Detection Using SVM Classifier Based on Textures

- Analysis,” *Appl. Math. Sci.*, vol. 5, no. 8, pp. 367–379, 2011.
- [35] S. Haykin, *Neural networks: A comprehensive foundation*. Prentice-Hall, 1999.
- [36] P. Wasserman, *Neural Computing Theory and Practice*. Van Nostrand Reinhold Publications, ANZA Research Inc., Van Nostrand Reinhold, New York, 1989.
- [37] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*. Peter Smith Publisher, Incorporated, 1981.
- [38] N. Mendoza, F. Sun D-W, and A. P, “Colour calibration of a laboratory computer vision system for quality evaluation of pre-sliced hams,” *Meat Sci(2009)*, p. 353362, 2009.
- [39] C. Du and D. Sun, “Quality measurement of cooked meats,” *Computer vision technology for food quality evaluation. Elsevier/Academic, London*, pp. 139–156, 2007.
- [40] D. M. W.H.Nailon, A.T. Redpath, “Texture analysis of 3D bladder cancer CT images for improving radiotherapy planning,” *Proceedings of the 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 652–655, May 2008.
- [41] Y. Huan, C. Caldwell, K. Mah, and D. Mozeg, “Coregistered FDG PET/CT-Based Textural Characterization of Head and Neck Cancer for Radiation Treatment Planning,” *IEEE Transactions on Medical Imaging*, vol. 28, no. 3, pp. 374–383, March 2009.
- [42] R. W. Connors and C. A. Harlow, “A theoretical comparison of texture algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. vol. PAMI-2, 1980.
- [43] Larry S. Davis, S. A. Johns, and J. K. Aggarwal, “Texture analysis using generalised co- occurrence matrices,”

IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. VOL. PAMI-1, pp. 251–259, 1979.

- [44] A. Gelzinis, A. Verikas, and M. Bacauskiene, “Increasing the discrimination power of the co-occurrence matrix-based features,” *Pattern Recognit.*, vol. 40, no. 9, pp. 2367–2372, 2007.
- [45] J. Kim, S. Kim, and T. Kang, “Fractal Dimension Co-occurrence Matrix Method for Texture Classification,” *TENCON 2006 - 2006 IEEE Reg. 10 Conf.*, vol. 00, no. 2, pp. 2–5, 2006.
- [46] F. Tsai, C.-K. Chang, J.-Y. Rau, T.-H. Lin, and G.-R. Liu, “3D Computation of Gray Level Co-occurrence in Hyperspectral Image Cubes,” *EMMCVPR 2007*, vol. LNCS 4679, pp. 429–440, 2007.
- [47] F. Argenti, L. Alparone, and G. Benelli, “Fast algorithms for texture analysis using co occurrence matrices,” *IEE Proceedings*, vol. 137, pp. 443–448, 1990.
- [48] M.E.Jernigan and F. Dastous, “Entropy based texture analysis in the spatial frequency domain,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. vol. PAMI-6, pp. 237–243, 1984.
- [49] F. Albrechtsen, B. Nielsen, and H. E. Danielsen, “Adaptive gray level run length features from class distance matrices,” *International Conference on Pattern Recognition 3(2000)*, pp. 3746–3749, 2000.
- [50] H. M. Hamdan and L. M. Larson, “Texture Classification through Level Lines,” *IEEE ICIP 2002*, pp. 937–940, 2002.
- [51] I. Epifanio and G. Ayala, “A random set view for texture classification,” *IEEE Transactions on Image Processing*, vol. 11, pp. 859–867, 1984.

- [52] P. Maillard, “Comparing texture analysis methods through classification,” *Photogrammetric Engineering and Remote Sensing*, vol. 69, pp. 357–367, 2003.
- [53] T. Tanaka, S. Torii, I. Kabuta, K. Shimizu, M. Tanaka, and H. Oka, “Pattern Classification of Nevus with Texture Analysis,” *Proc. 26th Annu. Int. Conf. IEEE EMBS San Fr.*, pp. 1459–1462, 2004.
- [54] M. Varma and A. Zisserman, *A statistical approach to texture classification from single images*. Kluwer Academic Publishers, 2004.
- [55] S. Liao and A. C. S. Chung, “Texture classification by using Advanced Local Binary Patterns and spatial distribution of dominant patterns,” *ICASSP 2007*, pp. 1221–1224, 2007.
- [56] M. Varma and A. Zisserman, “A statistical approach to texture classification from single images,” *Int. J. Comput. Vis.*, vol. 62, no. 1-2, pp. 61–81, 2005.
- [57] A. Alnadi, D. A. and Manzoor, “Independent component analysis for texture classification,” in *Proceedings of 5th International multiconference on Systems Signals and Devices*, (IEEE SSD), pp. 1–5, 2008.
- [58] T.-I. Hsu and R. Calway, A.D. and Wilson, “Texture analysis using multiresolution fourier transform,” *Proceedings of 8th Scandinavian conference on Image Analysis*, pp. 823–830, 1993.
- [59] T. Lonnestad, “A new set of texture features on the haar transform,” *Proceedings of the 11th International Conference on Pattern Recognition*, pp. 676–679, 1992.
- [60] M. Mokji and S. A. Bakar, “Gray level cooccurrence matrix computation based on haar transform,” in *Proceedings of the International Conference on Computer Graph-*

ics, Imaging and Visualisation, (Bangkok), pp. 273–279, August 2007.

- [61] T. Chang and C. Kuo, “Texture classification with tree structured wavelet transform,” *Proceedings of the 11th International Conference on Pattern Recognition*, vol. 2, pp. 256–259, 1992.
- [62] T. Chang and C. Kuo, “Wavelet transform approach to texture classification,” *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4, pp. 661–664, 1992.
- [63] M. Unser, “Texture classification and segmentation using wavelet frames,” *IEEE Transactions on Image Processing*, vol. 4, pp. 1549–1560, 1995.
- [64] P. Vautrot, N. Bonnet, and M. Herbin, “Comparative study of different spatial/spatial-frequency methods (gabor filters, wavelets, wavelets packets) for texture segmentation/classification,” *Proceedings of the 1996 IEEE Inter. Conf. Image Processing, ICIP’96*, vol. 3, pp. 145–148, 1996.
- [65] G. Van de Wouwer, P. Schenders, and D. Van Dyck, “Statistical texture characterization from discrete wavelet representations,” *IEEE Transactions on Image Processing*, vol. 8, pp. 592–598, 1999.
- [66] G. Naghdy, J. Wang, and P. Ogunbona, “Texture analysis using gabor wavelets,” *Proceedings of SPIE 2657*, pp. 74–85, 1996.
- [67] G. Naghdy, J. Wang, and P. Ogunbona, “A new wavelet based art network for texture classification,” *Proc. 1996 Australian New Zealand Conf. on Intelligent Information Systems*, pp. 250–253, 1996.

- [68] R. Porter and N. Canagarajah, “Robust rotation-invariant texture classification: Wavelet, gabor filter and gmrf based schemes,” *Proceedings of IEE Conference on Vision, Image, and Signal Processing*, vol. 144, pp. 180–188.
- [69] D. Kramer and F. Aghdasi, “Classifications of Microcalcifications in Digitized Mammograms using Multi-scale Statistical Texture Analysis,” *Proc. South African Symp. Commun. Signal Process.*, pp. 121–126, 1998.
- [70] S. Arivazhagan and L. Ganesan, “Texture classification using wavelet transform,” *Pattern Recogn. Lett.*, vol. 24, pp. 1513–1521, 2003.
- [71] K. Mala and V. Sadasivam, “Automatic segmentation and classification of diffused liver diseases using wavelet based texture analysis and neural network,” *Proc. IEEE Indicon 2005 Conf.*, pp. 216–219, 2005.
- [72] K. Mala and V. Sadasivam, “Wavelet based texture analysis of liver tumor from Computer Tomography images for characterization using linear vector quantisation neural network,” *Proc. Int. Conf. Adv. Comput. Commun. AD-COM*, pp. 267–270, 2006.
- [73] K. Mala, V. Sadasivam, and S. Alagappan, “Neural Network based Texture analysis of Liver Tumor from CT Images,” *World Acad. Sci. Eng. Technol.* 13, pp. 181–188, 2006.
- [74] K. Mala and V. Sadasivam, “Classification of fatty and cirrhosis liver using wavelet-based statistical texture features and neural network classifier,” *International Journal of Software Informatics*, vol. 4, pp. 151–163, 2010.
- [75] P. Hiremath and S. Shivashankar, “Texture classification using wavelet packet decomposition,” *ICGSTs GVIP J.*, vol. 6, no. 2, 2006.

- [76] P. S. Hiremath and S. Shivashankar, "Wavelet based features for Texture Classification," *ICGSTs GVIP J.*, vol. 6, pp. 55–58, 2006.
- [77] Z. Z. Wang and J. H. Yong, "Texture analysis and classification with linear regression model based on wavelet transform," *IEEE Transactions on Image Processing*, vol. 17, pp. 1421–1430, August 2008.
- [78] B. Ramana Reddy, M. Radhika Mani, and K. V. Subbaiah, "Texture classification method using wavelet transforms based on gaussian markov random field," *International Journal of Signal and Image Processing*, vol. 1, pp. 35–39, 2010.
- [79] R. C. Roy and R. Gopikakumari, "1d mrt: A transform for 1d signals," *Proceedings of All India Seminar on Electronic Innovations and Technology*, pp. 85–88, 2009.
- [80] M. S. Anish Kumar, R. C. Roy, and R. Gopikakumari, "A new image compression and decompression technique based on 8x8 MRT," *GVIP J.*, vol. 6, pp. 51–53, 2006.
- [81] M. Anish Kumar, R. C. Roy, and R. Gopikakumari, "A new transform coder for gray scale images using 4x4 mrt," *International Journal of Electronics and Communications*, pp. 627–630, 2008.
- [82] V. Bhadran, R. C. Roy, and R. Gopikakumari, "Visual representation of 2-D DFT in terms of 2X2 data: A pattern analysis," *Proc. 2008 Int. Conf. Comput. Commun. Networking, ICCCN 2008*, no. 4, 2008.
- [83] S. Loncaric and D. Kovacevic, "A method for segmentation of CT head images," *Proceedings of the 9th International Conference on Image Analysis and Processing, ICIAP 1997*, vol. II, pp. 388–395, 1997.

- [84] S. Nawaz and A. Dar, "Hepatic lesions classification by ensemble of svms using statistical features based on co-occurrence matrix," *Proceedings of 4th International Conference in emerging technologies*, pp. 21–26, 2008.
- [85] J. Y. Xiaojun Zang, Yongtian Wang and Y. Liu, "A novel method for ct brain image segmentation," *International Conference of Medical Image Analysis and Clinical Application*, pp. 109–112, 2010.
- [86] L. R. Varshney, "Abdominal Organ Segmentation in CT Scan Images: A Survey," tech. rep., Cornell University, 2002.
- [87] H. Wei and M. Bartels, "Unsupervised Segmentation Using Gabor Wavelets and Statistical Features in LIDAR Data Analysis," *IEEE*, 2006.
- [88] S. Luo, J. S. Jin, S. K. Chalup, and G. Qian, "A liver segmentation algorithm based on wavelets and machine learning," *International Conference on Computational Intelligence and Natural Computing*, pp. 122–125, 2009.
- [89] L. D. Cohen, "Contour Models," *CVGIP Graph. Model. Image Process.*, vol. 53, no. 2, pp. 211–218, 1991.
- [90] T. Cootes, A. Hill, C. J. Taylor, and J. Haslam, "The Use of active shape models for locating structures in medical images," *Image Vision Computing*, vol. 12, no. 6, pp. 355–365, 1994.
- [91] T. Cootes, C. Taylor, Cooper, and J. Graham, "Active shape models - Their training and applications," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.
- [92] N. Duta and M. Sonka, "Segmentation and interpretation of MR brain images: An improved active shape model,"

- IEEE Trans. Med. Imaging*, vol. 17, no. 6, pp. 1049–1062, 1998.
- [93] G. Behiels, D. Vandermeulen, F. Maes, P. Suetens, and P. Dewaele, “Active Shape Model-Based Segmentation of Digital X-ray Images Point Distribution Model : Capturing Shape Statistics,” *Proc. MICCAI*, vol. LNCS 1679, pp. 128–138, 1999.
- [94] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models,” *Proceedings of European conference on computer vision*, vol. 2, pp. 484–498, 1998.
- [95] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.
- [96] T. Cootes and C. Taylor, “Statistical models of appearance for medical image analysis and computer vision,” *Med. Imaging 2001*, pp. 236–248, 2001.
- [97] M. M. Dickens, S. Gleason, and H. Sari-Sarraf, “Volumetric segmentation via 3D active shape models,” *Image Anal. Interpret.*, vol. 02, pp. 248 – 252, 2002.
- [98] B. Van Ginneken, A. F. Frangi, J. J. Staal, Ter Haar Romeny, M. Bart, and M. A. Viergever, “Active shape model segmentation with optimal features,” *IEEE Trans. Med. Imaging*, vol. 21, no. 8, pp. 924–933, 2002.
- [99] H. C. V. Assen, M. G. Danilouchkine, F. Behloul, H. J. Lamb, and R. J. V. D. Geest, “Cardiac LV Segmentation Using a 3D Active Shape,” *LNCS*, vol. MICCAI 2003, no. 2878, pp. 533–540, 2003.
- [100] M. de Bruijne, B. van Ginneken, M. a. Viergever, and W. J. Niessen, “Adapting Active Shape Models for 3D Segmenta-

- tion of Tubular Structures in Medical Images,” *Inf. Process. Med. Imaging*, vol. 18, pp. 136–147, 2003.
- [101] J. W. Ma and Y. H. Fa, “Face segmentation algorithm based on ASM,” *Proc. - 2009 IEEE Int. Conf. Intell. Comput. Intell. Syst. ICIS 2009*, vol. 4, pp. 495–499, 2009.
- [102] Tian Shen, Hongsheng Li, Zhen Qian, and Xiaolei Huang, “Active volume models for 3D medical image segmentation,” *2009 IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. M, pp. 707–714, 2009.
- [103] P. Ghosh, M. Mitchell, J. A. Tanyi, and A. Y. Hung, “Automatic Segmentation of the Prostate Using a Genetic Algorithm for Prostate Cancer Treatment Planning,” *2010 Ninth Int. Conf. Mach. Learn. Appl.*, vol. 0, pp. 752–757, 2010.
- [104] S. Ghose, A. Oliver, R. Martí, X. Lladó, J. Freixenet, J. C. Vilanova, and F. Meriaudeau, “Texture guided active appearance model propagation for prostate segmentation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6367 LNCS, pp. 111–120, 2010.
- [105] S. Ghose, A. Oliver, R. Martí, X. Lladó, J. Freixenet, J. C. Vilanova, and F. Meriaudeau, “Prostate segmentation with texture enhanced active appearance model,” *Proc. 6th Int. Conf. Signal Image Technol. Internet Based Syst. SITIS 2010*, pp. 18–22, 2010.
- [106] S. Ghose, A. Oliver, R. Martí, X. Lladó, J. Freixenet, J. Mitra, J. C. Vilanova, J. Comet-Batlle, and F. Meriaudeau, “Statistical shape and texture model of quadrature phase information for prostate segmentation,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 7, no. 1, pp. 43–55, 2012.

- [107] H. Akbari, X. Yang, and L. Halig, “3D segmentation of prostate ultrasound images using wavelet transform,” *Proc. SPIE Vol*, vol. m, no. 404, pp. 1–10, 2011.
- [108] S. Sun, C. Bauer, and R. R. Beichel, “Robust active shape model based lung segmentation in CT scans,” *Fourth Int. Work. Pulm. Image Anal.*, pp. 213–223, 2011.
- [109] E. Ebhotemhen, “Medical Image Segmentation using an Extended Active Shape Model,” *International Journal of Computer Applications*, vol. 69, no. 19, pp. 24–29, 2013.
- [110] S.-I. Joo, S.-W. Jang, S.-W. Han, and G.-Y. Kim, “ASM-Based Objectionable Image Detection in Social Network Services,” *Int. J. Distrib. Sens. Networks*, vol. 2014, pp. 1–10, 2014.
- [111] H. Vafaie and I. F. Imam, “Feature selection methods: genetic algorithms vs. greedy-like search,” *Proceedings of the International Conference on Fuzzy and Intelligent Control Systems*, 1994.
- [112] A. Stolpmann and L. S. , “Genetic algorithm for automatic feature selection in a texture classification system,” *Proceedings of the Fourth International Conference on Signal Processing*, vol. 2, pp. 1229–1232, 1998.
- [113] S. Mougiakakou, I. Valavanis, K. Nikita, a. Nikita, and D. Kelekis, “Characterization of CT liver lesions based on texture features and a multiple neural network classification scheme,” *Proc. 25th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pp. 1287–1290, 2003.
- [114] Y. Liu and Y. F. Zheng, “FS SFS : A Novel Feature Selection Method for Support Vector Machines,” no. 1, pp. 797–800, 2004.

- [115] F. M. Van Coillie, L. P. Verbeke, and R. R. De Wulf, "Feature selection by genetic algorithms in object-based classification of IKONOS imagery for forest mapping in Flanders, Belgium," *Remote Sens. Environ.*, vol. 110, no. 4, pp. 476–487, 2007.
- [116] B. M. M. A. K. Kharrat A., Gasmi K. and B. N., "Automated classification of magnetic resonance brain images using wavelet genetic algorithm and support vector machine," *Proceedings of the 9th IEEE International Conference on Cognitive Informatics*, pp. 369–374, 2010.
- [117] P. Gnanasekar and A. Nagappan, "Investigation on Feature Extraction and Classification of Medical Images," *World Acad. Sci. Eng. Technol.*, vol. 60, pp. 327–332, 2011.
- [118] A. Padma and D. R. Sukanesh, "Automatic Diagnosis of Abnormal Tumor Region from Brain Computed Tomography Images Using Wavelet Based Statistical Texture Features," vol. 1, no. 3, p. 17, 2011.
- [119] A. Padma and R. Sukanesh, "Automatic Classification and Segmentation of Brain Tumor in CT Images using Optimal Dominant Gray level Run length Texture Features," *Int. J. Adv. Comput. Sci. Appl.*, vol. 2, no. 10, pp. 53–59, 2011.
- [120] J. Adams, D. Woodard, G. Dozier, P. Miller, G. Glenn, and K. Bryant, "GEFE : Genetic & Evolutionary Feature Extraction for Categories and Subject Descriptors," *ACMSE 10 April 15-17, 2010, Oxford, MS, USA*, 2010.
- [121] F. Alam and R. Faruqui, "Optimised calculation of haralick texture features," *European Journal of Scientific Research*, vol. 50, pp. 543–553, 2011.
- [122] M. Peterson, T. Doom, and M. Raymer, "GA-Facilitated KNN Classifier Optimization with Varying Similarity Mea-

- tures,” *2005 IEEE Congr. Evol. Comput.*, vol. 3, pp. 2514–2521, 2005.
- [123] K. Meenakshy and R. Gopikakumari, “Study on renal stones using grey level statistical texture features,” *Proceedings of the Third National Conference on Intelligent and Efficient Electrical*, pp. 627–630, 2008.
- [124] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Trans. Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
- [125] M. Aly, “Survey on multiclass classification methods,” *Neural Netw*, no. November, pp. 1–9, 2005.
- [126] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998.
- [127] R. Burbidge and B. Buxton, “An introduction to support vector machines for data mining,” *Keynote Pap. young OR12*, pp. 2–14, 2001.
- [128] P. Erästö, “Support Vector Machines - Backgrounds and Practice,” *Tech. Rep.*, pp. 1–79, 2001.
- [129] R. Berwick, “An Idiot’s guide to Support vector machines (SVMs),” *Retrieved Oct.*, pp. 1–25, 2003.
- [130] A. Asano, “Support vector machine and kernel method,” *Pattern Inf. Process. (2004 Autumn Semester)*, vol. 12, no. 3, pp. 4–6, 2004.
- [131] M. Diem, F. Kleber, and R. Sablatnig, “Text classification and document layout analysis of paper fragments,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, pp. 854–858, 2011.

- [132] J. D. Rennie, "Improving Multi-class Text Classification with Naive Bayes," p. 43, 2001.
- [133] Y. Avrithis and S. Kollias, "Fuzzy image classification using multiresolution neural networks with applications to remote sensing," *Proc. 13th Int. Conf. Digit. Signal Process.*, vol. 1, pp. 261–264.
- [134] W. Zhou, "Verification of the Nonparametric Characteristics of Backpropagation Neural Networks for Image Classification," vol. 37, no. 2, pp. 771–779, 1999.
- [135] S. Shah and V. Gandhi, "Image Classification Based on Textural Features using Artificial Neural Network (ANN)," pp. 72–77, 2004.
- [136] S. B. Park, J. W. Lee, and S. K. Kim, "Content-based image classification using a neural network," *Pattern Recognit. Lett.*, vol. 25, no. 3, pp. 287–300, 2004.
- [137] W. W. Y. Ng, A. Dorado, D. S. Yeung, W. Pedrycz, and E. Izquierdo, "Image classification with the use of radial basis function neural networks and the minimization of the localized generalization error," *Pattern Recognit.*, vol. 40, no. 1, pp. 19–32, 2007.
- [138] Z. Liu and Z. Luo, "Hand Motion Pattern Classifier Based on EMG Using Wavelet Packet Transform and LVQ Neural Networks," *Proc. 2008 IEEE Int. Symp. IT Med. Educ.*, pp. 28–32, 2008.
- [139] H. M. F. Ashour, Mohammed and K. M. Mahar, "Supervised Texture Classification Using Several Features Extraction Techniques Based on ANN and SVM," *AICCSA-08 Pattern Recognit. Image Process. Track*, pp. 567–574, 2008.
- [140] A. M. Elsayad, "Classification of ECG arrhythmia using learning vector quantization neural networks," *Proc. - 2009*

- Int. Conf. Comput. Eng. Syst. ICCES'09*, no. October, pp. 139–144, 2009.
- [141] C.-C. Lee and C.-Y. Shih, “Classification of Liver Disease from CT Images Using Sigmoid Radial Basis Function Neural Network,” *2009 WRI World Congr. Comput. Sci. Inf. Eng.*, pp. 656–660, 2009.
- [142] M. Lotfi, A. Solimani, A. Dargazany, H. Afzal, and M. Bandarabadi, “Combining wavelet transforms and neural networks for image classification,” *2009 IEEE Int. Symp. Sustain. Syst. Technol. ISSST 2009*, pp. 44–48, 2009.
- [143] M. Soranamageswari and C. Meena, “Statistical Feature Extraction for Classification of Image Spam Using Artificial Neural Networks,” *2010 Second Int. Conf. Mach. Learn. Comput.*, pp. 101–105, 2010.
- [144] J. Jiang, J. Zhang, G. Yang, D. Zhang, and L. Zhang, “Application of Back Propagation Neural Network in the Classification of High Resolution Remote Sensing Image,” *18th Int. Conf. Geoinformatics*, pp. 1–6, 2010.
- [145] T. Kavzoglu, “Increasing the accuracy of neural network classification using refined training data,” *Environ. Model. Softw.*, vol. 24, no. 7, pp. 850–858, 2009.
- [146] S. Kang and S. Park, “A fusion neural network classifier for image classification,” *Pattern Recognit. Lett.*, vol. 30, no. 9, pp. 789–793, 2009.
- [147] D. F. Specht, “Probabilistic neural networks and the polynomial Adaline as complementary techniques for classification,” *IEEE Trans. Neural Networks*, vol. 1, no. 1, pp. 111–121, 1990.

- [148] D. Specht, “Enhancements to probabilistic neural networks,” [*Proceedings 1992*] *IJCNN Int. Jt. Conf. Neural Networks*, vol. 1, pp. 761–768.
- [149] P. Spyridonos, D. Glostos, D. Cavouras, P. Ravazoula, and G. Nikiforidis, “A prognostic-classification system based on Probabilistic NN for predicting urine bladder recurrence ,” *14th International Conference on Digital Signal Processing, DSP2002*, pp. 1161–1164, July, 2002.
- [150] F. Goranescu, M. Gorunescu, E. El-Darzi, M. Ene, and S. Gorunescu, “Statistical comparison of a probabilistic neural network approach in hepatic cancer diagnosis,” *EU-ROCON 2005 - Int. Conf. Comput. as a Tool*, pp. 237–240, 2005.
- [151] Sheela Rani and B. S. Renganathan, “Texture Classification using wavelet packets with genetic algorithm as a tool for threshold selection,” *International Conference on Electronics, Control and Signal Processing*, 2002.
- [152] R. .Fallahi, A. M. Pooyan, and H. Mohammadnejad, “Application of morphological operations in human brain ct image with svm,” *Proceedings of Third International conference on Bioinformatics and Biomedical Engineering*, pp. 1–4, 2009.
- [153] M. O.R., M. C R, and B. W, “A Max-Min Measure for Image Texture Analysis,” *IEEE Trans. Comput.*, vol. Vol. C-26, pp. 408–414, 1977.
- [154] K. L. Chan and K. McCarty, “Aspects of the statistical of medical ultrasound images,” *IEE Colloquium on Ultrasound Instrumentation*, pp. 1–3, 1990.
- [155] S. Skorton, D.J.and Collins, N. J., P. N.G., J. Bean, and R. Kerber, “Quantitative texture analysis in two-dimensional echocardiography :application to the diagno-

- sis of experimental myocardial contusion.,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 68, pp. 217–223, 1983.
- [156] W. Nailon and S. McLaughlin, “Comparative study of textural analysis techniques to characterize tissue from intravascular ultrasound,” *Proceedings of IEEE International Conference of Image Processing*, pp. 303–305, 1996.
- [157] Q. Wei and Y. Hu, “A study on using texture analysis methods for identifying lobar fissure regions in isotropic ct images,” (IEEE, EMBS, USA), pp. 3537–3540, Sept. 2009.
- [158] A. S. Mark and S. Liwen, “Efficient Image Texture Analysis and Classification for Prostate Ultrasound Diagnosis,” *Proc. 2005 IEEE Comput. Syst. Bioinforma. Conf. Work.*, pp. 7–8, 2005.
- [159] O. S. Al-Kadi and D. Watson, “Texture analysis of aggressive and nonaggressive lung tumor ce ct images,” *IEEE Transactions on Biomedical Engineering*, vol. 55, pp. 1822–1830, 2008.
- [160] D. R. C. Philips, D. Li and J. Furst, “Directional invariance of co-occurrence matrices within the liver,” *Proceedings of the International Conference on Biocomputation, Bioinformatics and Biomedical Technologies*, pp. 29–34, July 2008.
- [161] B. K. R. Kale, M.C.and Clymer, J. Heverhagen, S. R. Sammet, S. M.V., and Knopp, “Multispectral co-occurrence with three random variables in dynamic contrast enhanced magnetic resonance imaging of breast cancer,” *IEEE Transactions on Medical Imaging*, vol. 27, pp. 1425–1431, 2008.
- [162] M. Pereira, M. Schaefer, and B. M. J, “Remote expert system of support the prostate cancer diagnosis,” *Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.*, vol. 5, pp. 3412–3415, 2004.

- [163] S. S. Mohamed, M. M. A. Salama, M. Kamel, E. F. El-Saadany, K. Rizkalla, and J. Chin, “Prostate cancer multi-feature analysis using trans-rectal ultrasound images,” *Phys. Med. Biol.*, vol. 50, no. 15, pp. N175–N185, 2005.
- [164] A. Tabesh, M. Teverovskiy, H.-y. Pang, V. P. Kumar, and D. Verbel, “Multifeature Prostate Cancer Diagnosis and Gleason Grading of Histological Images,” vol. 26, no. November, pp. 1366–1378, 2007.
- [165] E. Alexandratou, D. Yova, D. Gorpas, P. Maragos, G. Agrogiannis, and N. Kavantzias, “Texture analysis of tissues in Gleason grading of prostate cancer,” vol. 6859, p. 04, 2008.
- [166] Y. Zhan, D. Shen, S. Member, J. Zeng, L. Sun, G. Fichtinger, J. Moul, and C. Davatzikos, “Targeted Prostate Biopsy Using Statistical Image Analysis,” *IEEE Trans. Med. Imaging*, vol. 26, no. 6, pp. 779–788, 2007.
- [167] The Prostate Cancer Charity, *How prostate cancer is diagnosed*. 2010.
- [168] Prostate Cancer Taskforce, *Diagnosis and Management of Prostate Cancer in New Zealand Men: Recommendations from the Prostate Cancer Taskforce*. 2012.
- [169] S. Liao, Y. Gao, J. Lian, and D. Shen, “Sparse patch-based label propagation for accurate prostate localization in CT images,” *IEEE Trans. Med. Imaging*, vol. 32, no. 2, pp. 419–434, 2013.
- [170] M. Sadeghi, *Towards Prevention and Early Diagnosis of Skin Cancer: Computer-Aided Analysis of Dermoscopy Images*. Ph.D dissertation, Iran University of Science and Technology, Iran, 2009.

- [171] R. Dobrescu, M. Dobrescu, S. Mocanu, and D. Popescu, "Medical images classification for skin cancer diagnosis based on combined texture and fractal analysis," *WSEAS Transactions on Biology and Biomedicine*, vol. 7, pp. 223–232, 2010.
- [172] L. Ballerini, X. Li, R. B. Fisher, B. Aldridge, and J. Rees, "Content-based image retrieval of skin lesions by evolutionary feature synthesis," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6024 LNCS, no. PART 1, pp. 312–319, 2010.
- [173] R. B. Aswin, J. A. Jaleel, and S. Salim, "Implementation of ANN Classifier using MATLAB for Skin Cancer Detection," 2013.
- [174] M. G. Bincy, S. R. Muddanna, A. Kumar, N. Suvarna, and J. S. DSouza vol. 6, no. 1.
- [175] R. K. Megalingam, R. Venumadhav, A. P. K, A. Mahadevan, and T. Charly, "Kinect Based Wireless Robotic Coconut Tree Climber," pp. 201–206, 2013.
- [176] R. K. Megalingam, T. C. Kattakayam, T. Harikrishna Menon, and R. V. Madhav, "CocoBot: A kinect based coconut tree climber," *Int. J. Appl. Eng. Res.*, vol. 7, no. 11 SUPPL., pp. 1497–1502, 2012.
- [177] G. Jose and P. Manimegalai, "Identification of Precise Object among Various Objects using Sparse Coding," *International Journal of Computer Applications (0975 8887)*, vol. 154, no. 10, pp. 24–28, 2016.
- [178] V. Murugan and T. Devakumar, "Identification of Matured Coconut and its location using OPENCV," *International Journal of Computer Applications (0975 8887)*, pp. 242–247, 2016.

- [179] T. S. Wibowo, I. A. Sulistijono, and A. Risnumawan, “End-to-end coconut harvesting robot,” *Proc. - 2016 Int. Electron. Symp. IES 2016*, pp. 444–449, 2017.
- [180] W. H. Wolf, “Hardware Software Co-design of Embedded systems,” *Proceedings of IEEE*, vol. 8, no. 7, pp. 967–989, 1994.
- [181] J. Takalo, J. Kriinen, P. Pivi, and T. Ihme, “Challenges of software-hardware co-design Prestudy in TWINS project ,” *White paper*, 2008.
- [182] M. Kiran, K. M. War, L. M. Kuan, and L. K. Meng, “Implementing image processing algorithms using ”Hardware in the loop” approach for Xilinx FPGA,” *Int. Conf. Electron. Des. (ICED 2008), Penang*, pp. 1–6, 2008.
- [183] P. SivaramaPrasad and K. Srinivasa Rao, “Hardware and Software Codesign for Computer Screen Image Processing Applications using FPGA,” *Int. J. Comput. Appl.*, pp. 7–11, 2012.
- [184] S. Tagzout, K. Achour, and O. Djekoune, “Hough transform algorithm for FPGA implementation,” *Signal Processing*, vol. 81, pp. 1295–1301, 2001.
- [185] M. Fleury, R. Self, and A. Downton, “Development of a fine-grained parallel KarhunenLoe‘ve transform,” *Journal of Parallel and Distributed Computing*, vol. 64, pp. 520–1465, 2004.
- [186] M. Maamoun, M. Neggazi, A. Meraghni, and D. Berkani, “VLSI Design of 2-D Discrete Wavelet Transform for Area-Efficient and High-Speed Image Computing,” *International Journal of Computer and Information Engineering*, vol. 2, no. 9, pp. 3056–3061, 2008.

- [187] P. Kumar Biswal, P. Mondal, and S. Banerjee, “Parallel architecture for accelerating affine transform in high-speed imaging systems,” *J Real-Time Image Proc*, vol. 8, pp. 69–79, 2013.
- [188] C. Lin, L. Tao, and H. Keung Kwan, “Parallel-computing-based implementation of fast algorithms for discrete Gabor transform,” *IET Signal Processing*, vol. 9, no. 7, pp. 546–552, 2015.
- [189] A. Asaduzzaman, A. Martinez, and A. Sepehri, “A Time-Efficient Image Processing Algorithm for Multi-core/Manycore Parallel Computing,” *Proceedings of the IEEE SoutheastCon 2015*, 2015.
- [190] R. Archana, *Neural Network based Modeling, Characterization and Identification of Chaotic Systems in Nature*. PhD thesis, Cochin University of Science and Technology, Kochi, India, 2015.

Appendix A

Texture Features

A.1 GLCM and Haralick Features

Suppose an image to be analyzed is rectangular and has N_x resolution cells in the horizontal direction and N_y resolution cells in the vertical direction. Each resolution cell is quantized to N_g gray tone levels. Let the Horizontal Spatial Domain be $L_x = 1, 2, \dots, N_x$ and Vertical Spatial Domain be $L_y = 1, 2, \dots, N - y$ and Set of quantized gray tones be $G = 1, 2, \dots, N_g$. The texture content information is specified by the matrix of relative frequencies $P_{i,j}$ with two neighboring pixels separated by a distance d occur on the image, one with gray level i and other with gray level j . Such matrices of gray level co-occurrence frequencies are a function of the angular relationship and distance between neighboring pixels.

Formally for angles quantized to 45° intervals, the unnormalized

frequencies are defined as shown in the equations below.

$$\begin{aligned}
 P(i, j, d, 0^\circ) &= \#\{((k, l)(m, n)) \in (L_y \times L_x) \times (L_y \times L_x) \\
 &\quad |k - m = 0, |l - n| = d, \\
 &\quad I(k, l) = i, I(m, n) = j\} \quad (\text{A.1})
 \end{aligned}$$

$$\begin{aligned}
 P(i, j, d, 45^\circ) &= \#\{((k, l)(m, n)) \in (L_y \times L_x) \times (L_y \times L_x) \\
 &\quad k - m = d, l - n = -d, \text{ or } k - m = -d, l - n = d \\
 &\quad I(k, l) = i, I(m, n) = j\} \quad (\text{A.2})
 \end{aligned}$$

$$\begin{aligned}
 P(i, j, d, 90^\circ) &= \#\{((k, l)(m, n)) \in (L_y \times L_x) \times (L_y \times L_x) \\
 &\quad |k - m| = d, l - n = 0, \\
 &\quad I(k, l) = i, I(m, n) = j\} \quad (\text{A.3})
 \end{aligned}$$

$$\begin{aligned}
 P(i, j, 0^\circ) &= \#\{((k, l)(m, n)) \in (L_y \times L_x) \times (L_y \times L_x) \\
 &\quad k - m = d, l - n = d \text{ or } k - m = -d, l - n = -d, \\
 &\quad I(k, l) = i, I(m, n) = j\} \quad (\text{A.4})
 \end{aligned}$$

A set of 14 textural features extracted from each of the gray level co-occurrence matrices is defined by Haralick *et al.*[11]. The following equations define these features.

Notation

$p(i, j)$ $(i, j)^{th}$ entry in a normalized gray level co-occurrence matrix, $=P(i, j)/R$.

$p_x(i)$ i^{th} entry in the marginal probability matrix obtained by summing the rows of $p(i, j) = \sum_{j=1}^{N_g} P(i, j)$.

$$P_y(j) = \sum_{i=1}^{N_g} p(i, j)$$

$$p_{x+y}(k) = \sum_{i=1}^{N_g} \sum_{\substack{j=1 \\ i+j=k}}^{N_g} p(i, j), \quad k = 2, 3, \dots, 2N_g.$$

$$p_{x-y}(k) = \sum_{i=1}^{N_g} \sum_{\substack{j=1 \\ |i-j|=k}}^{N_g} p(i, j), \quad k = 0, 1, \dots, N_g - 1.$$

Textural Features

1. *Angular Second Moment* : $f_1 = \sum_i \sum_j \{p(i, j)\}^2$.
2. *Contrast* : $f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{\substack{i=1 \\ |i-j|=n}}^{N_g} \sum_{j=1}^{N_g} p(i) \right\}$
3. *Correlation* : $f_3 = \frac{\sum_i \sum_j (ij)p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}$ where μ_x , μ_y , σ_x and σ_y are the means and standard deviations of p_x and p_y .
4. *Sum of squares : Variance* $f_4 = \sum_i \sum_j (i - \mu)^2 p(i, j)$.
5. *Inverse Difference Moment* : $f_5 = \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i, j)$.
6. *Sum Average* : $f_6 = \sum_{i=2}^{2N_g} ip_{x+y}(i)$.
7. *Sum Variance* : $f_7 = \sum_{i=2}^{2N_g} (i - f_6)^2 p_{x+y}(i)$.
8. *Sum Entropy* : $f_8 = - \sum_{i=2}^{2N_g} P_{x+y}(i) \log p_{x+y}(i)$
9. *Entropy* : $f_9 = - \sum_i \sum_j jp(i, j) \log(p(i, j))$
10. *Difference Variance* : $f_{10} = \text{variance of } p_{x-y}$
11. *Difference Entropy* : $f_{11} = - \sum_{i=0}^{N_g-1} p_{x-y}(i) \log \{p_{x-y}(i)\}$.
- 12,13. *Information Measures of correlation:*

$$f_{12} = \frac{HXY - HXY1}{\max\{HX, HY\}}$$

$$f_{13} = (1 - \exp[-2.0(HXY2 - HXY)])^{1/2}$$

$$HXY = - \sum_i \sum_j p(i, j) \log(p(i, j))$$

where HX and HY are entropies of p_x and p_y , and

$$HXY1 = - \sum_i \sum_j p(i, j) \log(p_x(i)p_y(j))$$

$$HXY2 = - \sum_i \sum_j p_x(i)p_y(j) \log(p_x(i)p_y(j))$$

14. *Maximal Correlation coefficient:*

$$f_{14} = (\text{Second largest eigenvalue of } Q)^{1/2}$$

A.2 GLRL matrix and texture features

The metrics for texture analysis computed from GLRL matrices are termed as Short run low gray-level emphasis, Short run high gray-level emphasis, Long run low gray-level emphasis, Long run high gray-level emphasis, Gray level non uniformity and Run length Non uniformity. Short Run Emphasis and Long Run Emphasis emphasizes short run and long run respectively. When runs are equally distributed throughout the gray levels, Gray level non uniformity measure will have a low value. Run length Non uniformity measure will have a long value when the runs are equally distributed throughout the lengths. The following expressions give the five textural features extracted from GLRL matrices.

Notation

$p(i, j)$ $(i, j)^{th}$ entry in gray level run length matrix.

N_g be the number of gray levels in the picture.

N_r be the number of different run lengths that occur.

P be the number of points in the picture.

GLRL features

1.

Short Runs Emphasis, RF1 =

$$\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} \frac{p(i, j)}{j^2} / \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} p(i, j)$$

2.

Long Runs Emphasis, RF2 =

$$\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} j^2 p(i, j) / \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} p(i, j)$$

3.

Gray Level NonUniformity, RF3 =

$$\sum_{i=1}^{N_g} \left(\sum_{j=1}^{N_r} p(i, j) \right)^2 / \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} p(i, j)$$

4.

Run Length NonUniformity, RF4 =

$$\sum_{j=1}^{N_g} \left(\sum_{i=1}^{N_r} p(i, j) \right)^2 / \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} p(i, j)$$

5.

Run Percentage, RF5 =

$$\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} p(i, j) / P$$

A.3 Wavelet Transform based Texture features

Wavelet transform [62] is a multiresolution technique, implemented as pyramid or tree structure, by applying separable filter bank, $h_{LL}(k, l) = h(k)h(l)$, $h_{LH}(k, l) = h(k)g(l)$, $h_{HL}(k, l) = g(k)h(l)$, and $h_{HH}(k, l) = g(k)g(l)$. h and g are low pass and high pass filters respectively. An image can be decomposed into four sub images by convolving the image with these filters. The four sub images characterize the frequency information of the image in the LL, LH, HL, and HH frequency regions, respectively. The 2-D pyramidal structured wavelet transform(PSWT) recursively decomposes the image in the LL regions(Fig. A.1), whereas the 2-D

wavelet packet transform decomposes all frequencies to achieve a full decomposition(Fig. A.2). The tree structured wavelet transform is the algorithmic approach to implement the decomposition of a function in terms of wavelet packet basis.

LL ₃	HL ₃	HL ₂	HL ₁
LH ₃	HH ₃		
LH ₂		HH ₂	
LH ₁		HH ₁	

Fig. A.1: Three level 2-D PSWT decomposition of 128x128 image.

LL ₃	HL ₃	LL ₃	HL ₃	LL ₃	HL ₃	LL ₃	HL ₃
LH ₃	HH ₃	LH ₃	HH ₃	LH ₃	HH ₃	LH ₃	HH ₃
LL ₃	HL ₃	LL ₃	HL ₃	LL ₃	HL ₃	LL ₃	HL ₃
LH ₃	HH ₃	LH ₃	HH ₃	LH ₃	HH ₃	LH ₃	HH ₃
LL ₃	HL ₃	LL ₃	HL ₃	LL ₃	HL ₃	LL ₃	HL ₃
LH ₃	HH ₃	LH ₃	HH ₃	LH ₃	HH ₃	LH ₃	HH ₃
LL ₃	HL ₃	LL ₃	HL ₃	LL ₃	HL ₃	LL ₃	HL ₃
LH ₃	HH ₃	LH ₃	HH ₃	LH ₃	HH ₃	LH ₃	HH ₃

Fig. A.2: Three level 2-D Wavelet Packet decomposition of 128x128 image

Wavelet based texture analysis focuses on extracting the energy values from the sub images that characterizes the texture image. A commonly used energy signature for characterizing texture, mean deviation signature is explained in [65] .

If the sub image is of size $M \times N$, its energy signature (mean deviation signature) can be represented as,

$$e = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |x(i, j)| \quad (A.5)$$

where $x(i,j)$ is the pixel value of the sub image.

Appendix **B**

Patient Proforma for CT Based Texture Analysis in Prostate Disease

1. Name of Patient
2. Age of Patient
3. Clinical Symptoms with Duration
4. Physical Examination Findings
5. DRE Findings
6. Abdomen Ultra Sound Scan
7. TRUS Report
8. X-Ray KUB
9. CT Scan
10. MRI

11. PSA

Treatment Adopted

Histopathology Report

Appendix C

Primitive Symbols used in Visual Representation of 2-D DFT coefficients based on 2×2 Data

Let the 2×2 data be X ,

$$X = \begin{bmatrix} x_{00} & x_{01} \\ x_{10} & x_{11} \end{bmatrix} \quad (\text{C.1})$$

Primitive symbols are derived from the mapping relation between 2×2 data and 2×2 DFT [15].

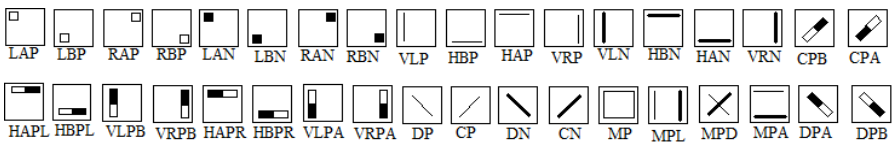


Fig. C.1: Primitive Symbols based on 2×2 Data

In order to understand the primitive symbols, in general the following rule may be applied:

- A white square implies that the data at that point is to be added.
- A black square implies that data at that point is to be subtracted.
- Thin lines indicate that the data involved are to be added.
- Bold lines indicate that the data involved are to be subtracted.

The alphabets in the mnemonics stands for:

- L: Left
- R: Right
- H : Horizontal
- V : Vertical
- A: Above
- B: Below
- P: Positive
- N: Negative
- M: Matrix
- D: Diagonal
- C: Cross Diagonal

The meaning of few primitive symbols and the corresponding mnemonics used in the visual representation based on the 2×2 data matrix are given below:

1. Symbol named LAP(Left Above Positive) indicates the element in the $(0, 0)^{th}$ position of the 2×2 data matrix is taken with a positive sign for UMRT coefficient computation and the rest of the data not considered.

$$LAP = x_{00}$$

2. Symbol named RBN(Right Below Negative) indicates the $(1, 1)^{th}$ position element of the 2×2 matrix is taken with a negative sign for the computation and the rest of the data not considered.

$$RBN = x_{11}$$

3. Symbol named HAP (Horizontal Above Positive) indicates that two data points on the upper row of the 2×2 matrix at positions $(0, 0)$ and $(0, 1)$ are taken with a positive sign and the rest are not considered for coefficient computation.

$$HAP = x_{00} + x_{01}.$$

Appendix **D**

Poof of Theorem 5.3

D.1 Theorem 5.3

One type of pattern C, D, E, or F and/or its sign reversed form will be present in the visual representation of SMRT coefficients and the type of the pattern depends on whether the sequency c_1 and/or c_2 is even or odd.

D.2 Proof

D.2.1 Case 1: c_1 and c_2 even or zero

Suppose the data at (n_1, n_2) has to be added in the computation of a particular SMRT coefficient. From expression of SMRT Kernel given in equation 1.4: For $c_1 \leq c_2$, then

$$((n_1 \cdot c_1 \cdot (1 + 2 \cdot i_2) + n_2 \cdot c_2))_N = c_1 \cdot i_1 \quad (\text{D.1})$$

else for $c_1 > c_2$, then

$$((n_1.c_1.(1 + 2.i_1) + n_2.c_2))_N = c_2.i_2 \quad (D.2)$$

When c_1 and c_2 even or zero, they can be written as $c_1 = 2.t_1$ and $c_2 = 2.t_2$ where t_1 and t_2 can be any whole number. Then for the data point M space apart which is at $(n_1, n_2 + M)$:

For $c_1 \leq c_2$

$$((n_1.c_1.(1+2.i_2)+(n_2+M).c_2))_N = (((n_1.c_1.(1+2.i_2)+(n_2).c_2))_N + M.c_2))_N = (c_1.i_1 + M.2.t_2))_N = c_1.i_1$$

For $c_1 > c_2$

$$((n_1.c_1.(1+2.i_1)+(n_2+M).c_2))_N = (((n_1.c_1.(1+2.i_1)+(n_2).c_2))_N + M.c_2))_N = (c_2.i_2 + M.2.t_2))_N = c_2.i_2$$

Similarly for data point $(n_1 + M, n_2)$ and $(n_1 + M, n_2 + M)$ also we can prove equations D.1 and D.2 remains same.

Thus it can be concluded that if the data at (n_1, n_2) is to be added, then all the other three data M-spacing apart are also to be added.

Now, suppose the data at (n_1, n_2) has to be subtracted in the computation of a particular SMRT coefficient. From expression of SMRT Kernel given in equation 1.4:

For $c_1 \leq c_2$, then

$$((n_1.c_1.(1 + 2.i_2) + n_2.c_2))_N = c_1.i_1 + M \quad (D.3)$$

else for $c_1 > c_2$, then

$$((n_1.c_1.(1 + 2.i_1) + n_2.c_2))_N = c_2.i_2 + M \quad (D.4)$$

Then for the data point M space apart which is at $(n_1, n_2 + M)$:

For $c_1 \leq c_2$

$$((n_1.c_1.(1+2.i_2)+(n_2+M).c_2))_N = (((n_1.c_1.(1+2.i_2)+(n_2).c_2))_N +$$

$$M.c_2))_N = (c_1.i_1 + M + M.2.t_2))_N = c_1.i_1 + M$$

For $c_1 > c_2$

$$((n_1.c_1.(1+2.i_1)+(n_2+M).c_2))_N = (((n_1.c_1.(1+2.i_1)+(n_2).c_2))_N + M.c_2))_N = (c_2.i_2 + M + M.2.t_2))_N = c_2.i_2 + M$$

Similarly for data point $(n_1 + M, n_2)$ and $(n_1 + M, n_2 + M)$ also we can prove equations D.3 and D.4 remains same.

Thus it is understood that if the data at (n_1, n_2) is to be subtracted, then all the other three data at M-spacing are also to be subtracted.

The addition or subtraction of the four points M-space apart can be defined as the element $C(n_1, n_2)$ of C matrix. Hence when the sequencies are both even, the C matrix is involved in the computation.

D.2.2 Case 2: c_1 and c_2 odd

Suppose the data at (n_1, n_2) has to be added in the computation of a particular SMRT coefficient. Then equations D.1 and D.2 are applicable.

When c_1 and c_2 odd, they can be written as $c_1 = 2.t_1 + 1$ and $c_2 = 2.t_2 + 1$ where t_1 and t_2 can be any whole number.

Then for the data point M space apart which is at $(n_1, n_2 + M)$:

For $c_1 \leq c_2$

$$((n_1.c_1.(1+2.i_2)+(n_2+M).c_2))_N = (((n_1.c_1.(1+2.i_2)+(n_2).c_2))_N + M.c_2))_N = (c_1.i_1 + M.(2.t_2 + 1))_N = c_1.i_1 + M$$

For $c_1 > c_2$

$$((n_1.c_1.(1+2.i_1)+(n_2+M).c_2))_N = (((n_1.c_1.(1+2.i_1)+(n_2).c_2))_N + M.c_2))_N = (c_2.i_2 + M.(2.t_2 + 1))_N = c_2.i_2 + M$$

Hence, for data point $(n_1, n_2 + M)$ equations D.3 and D.4 holds, implies data at the point to be subtracted in the computation of SMRT coefficient.

Similarly for data point $(n_1 + M, n_2)$ and $(n_1 + M, n_2 + M)$ it can be proved that equations D.3 and D.4 and equations D.1 and D.2 respectively are applicable.

Thus it can be concluded that if (c_1, c_2) is odd then the data at (n_1, n_2) and $(n_1 + M, n_2 + M)$ are to be added and data at $(n_1, n_2 + M)$ and $(n_1 + M, n_2)$ are to be subtracted.

Now, suppose the data at (n_1, n_2) has to be subtracted in the computation of a particular SMRT coefficient. Then equations D.3 and D.4 are applicable.

Then for the data point M space apart which is at $(n_1, n_2 + M)$:

For $c_1 \leq c_2$

$$((n_1 \cdot c_1 \cdot (1 + 2 \cdot i_2) + (n_2 + M) \cdot c_2))_N = (((n_1 \cdot c_1 \cdot (1 + 2 \cdot i_2) + (n_2) \cdot c_2))_N + M \cdot c_2)_N = (c_1 \cdot i_1 + M + M \cdot (2 \cdot t_2 + 1))_N = c_1 \cdot i_1$$

For $c_1 > c_2$

$$((n_1 \cdot c_1 \cdot (1 + 2 \cdot i_1) + (n_2 + M) \cdot c_2))_N = (((n_1 \cdot c_1 \cdot (1 + 2 \cdot i_1) + (n_2) \cdot c_2))_N + M \cdot c_2)_N = (c_2 \cdot i_2 + M + M \cdot (2 \cdot t_2 + 1))_N = c_2 \cdot i_2$$

Hence, for data point $(n_1, n_2 + M)$ equations D.1 and D.2 holds good, which means data at the point has to be added.

Similarly for data point $(n_1 + M, n_2)$ and $(n_1 + M, n_2 + M)$ it can be proved that equations D.1 and D.2 and equations D.3 and D.4 respectively are applicable.

Thus it can be concluded that if (c_1, c_2) is odd then the data at (n_1, n_2) and $(n_1 + M, n_2 + M)$ are to be subtracted and data at $(n_1, n_2 + M)$ and $(n_1 + M, n_2)$ are to be added.

The addition of data points (n_1, n_2) and $(n_1 + M, n_2 + M)$ to be subtracted from data $(n_1, n_2 + M)(n_1 + M, n_2)$ can be defined as the element $F(n_1, n_2)$ of F matrix. Hence when the sequences are both odd, the F matrix is involved in the computation.

D.2.3 Case 3: c_1 even and c_2 odd

Suppose the data at (n_1, n_2) has to be added in the computation of a particular SMRT coefficient. Then equations D.1 and D.2 are applicable.

When c_1 even and c_2 odd, they can be written as $c_1 = 2.t_1$ and $c_2 = 2.t_2 + 1$ where t_1 and t_2 can be any whole number.

Then for the data point M space apart which is at $(n_1, n_2 + M)$:

For $c_1 \leq c_2$

$$((n_1.c_1.(1+2.i_2)+(n_2+M).c_2))_N = (((n_1.c_1.(1+2.i_2)+(n_2).c_2))_N + M.c_2))_N = (c_1.i_1 + M.(2.t_2 + 1))_N = c_1.i_1 + M$$

For $c_1 > c_2$

$$((n_1.c_1.(1+2.i_1)+(n_2+M).c_2))_N = (((n_1.c_1.(1+2.i_1)+(n_2).c_2))_N + M.c_2))_N = (c_2.i_2 + M.(2.t_2 + 1))_N = c_2.i_2 + M$$

Hence, for data point $(n_1, n_2 + M)$ equations D.3 and D.4 holds, implies data at the point to be subtracted in the computation of SMRT coefficient.

For data point $(n_1 + M, n_2)$ equations D.1 and D.2 holds, which tells that the data there has to be added.

Similarly for data point $(n_1 + M, n_2 + M)$ it can proved that equations D.3 and D.4 are applicable, which means the data has to be subtracted.

Thus it can be concluded that if c_1 is even and c_2 is odd then the data at (n_1, n_2) and $(n_1 + M, n_2)$ are to be added and data at $(n_1, n_2 + M)$ and $(n_1 + M, n_2 + M)$ are to be subtracted.

Now, suppose the data at (n_1, n_2) has to be subtracted in the computation of a particular SMRT coefficient. Then equations D.3 and D.4 are applicable.

Then for the data point M space apart which is at $(n_1, n_2 + M)$:

For $c_1 \leq c_2$

$$((n_1.c_1.(1+2.i_2)+(n_2+M).c_2))_N = (((n_1.c_1.(1+2.i_2)+(n_2).c_2))_N + M.c_2))_N = (c_1.i_1 + M + M.(2.t_2 + 1))_N = c_1.i_1$$

For $c_1 > c_2$

$$((n_1.c_1.(1+2.i_1)+(n_2+M).c_2))_N = (((n_1.c_1.(1+2.i_1)+(n_2).c_2))_N + M.c_2))_N = (c_2.i_2 + M + M.(2.t_2 + 1))_N = c_2.i_2$$

Hence, for data point $(n_1, n_2 + M)$ equations D.1 and D.2 holds good, which means data at the point has to be added.

For data point $(n_1 + M, n_2)$ equations D.3 and D.4 holds, which tells that the data there has to be subtracted.

Similarly for data point $(n_1 + M, n_2 + M)$ it can be proved that equations D.1 and D.2 are applicable, which means the data has to be added.

Thus it can be concluded that if c_1 is even and c_2 is odd then the data at (n_1, n_2) and $(n_1 + M, n_2)$ are to be subtracted and data at $(n_1, n_2 + M)$ and $(n_1 + M, n_2 + M)$ are to be added.

The addition of data points (n_1, n_2) and $(n_1 + M, n_2)$ to be subtracted from data $(n_1, n_2 + M)(n_1 + M, n_2 + M)$ can be defined as the element $E(n_1, n_2)$ of E matrix. Hence when the sequences are even odd, the E matrix is involved in the computation.

D.2.4 Case 3: c_1 odd and c_2 even

Suppose the data at (n_1, n_2) has to be added in the computation of a particular SMRT coefficient. Then equations D.1 and D.2 are applicable.

When c_1 odd and c_2 even, they can be written as $c_1 = 2.t_1 + 1$ and $c_2 = 2.t_2$ where t_1 and t_2 can be any whole number.

Then for the data point M space apart which is at $(n_1 + M, n_2 + M)$:

For $c_1 \leq c_2$

$$(((n_1 + M).c_1.(1 + 2.i_2) + (n_2 + M).c_2))_N = (((((n_1.c_1.(1 + 2.i_2) + (n_2).c_2))_N + M.c_2))_N = (c_1.i_1 + M.(2.t_2)))_N = c_1.i_1$$

For $c_1 > c_2$

$$(((n_1.c_1.(1+2.i_1)+(n_2+M).c_2))_N = (((((n_1.c_1.(1+2.i_1)+(n_2).c_2))_N + M.c_2))_N = (c_2.i_2 + M.(2.t_2)))_N = c_2.i_2$$

Hence, for data point $(n_1, n_2 + M)$ equations D.1 and D.2 holds, implies data at the point to be added in the computation of SMRT coefficient.

For data points $(n_1 + M, n_2)$ and $(n_1 + M, n_2 + M)$ equations D.3 and D.4 holds, which tells that the data there has to be subtracted.

Thus it can be concluded that if c_1 is odd and c_2 is even then the data at (n_1, n_2) and $(n_1, n_2 + M)$ are to be added and data at $(n_1 + M, n_2)$ and $(n_1 + M, n_2 + M)$ are to be subtracted.

Now, suppose the data at (n_1, n_2) has to be subtracted in the computation of a particular SMRT coefficient. Then equations D.3 and D.4 are applicable.

Then for the data point M space apart which is at $(n_1, n_2 + M)$:

For $c_1 \leq c_2$

$$(((n_1.c_1.(1+2.i_2)+(n_2+M).c_2))_N = (((((n_1.c_1.(1+2.i_2)+(n_2).c_2))_N + M.c_2))_N = (c_1.i_1 + M.(2.t_2 + 1)))_N = c_1.i_1 + M$$

For $c_1 > c_2$

$$(((n_1.c_1.(1+2.i_1)+(n_2+M).c_2))_N = (((((n_1.c_1.(1+2.i_1)+(n_2).c_2))_N + M.c_2))_N = (c_2.i_2 + M.(2.t_2 + 1)))_N = c_2.i_2 + M$$

Hence, for data point $(n_1, n_2 + M)$ equations D.3 and D.4 holds good, which means data at the point has to be subtracted.

For data point $(n_1 + M, n_2)$ and $(n_1 + M, n_2 + M)$ equations D.1 and D.2 holds, which tells that the data there has to be added.

Thus it can be concluded that if c_1 is odd and c_2 is even then the data at (n_1, n_2) and $(n_1, n_2 + M)$ are to be subtracted and data at $(n_1 + M, n_2)$ and $(n_1 + M, n_2 + M)$ are to be added.

The addition of data points (n_1, n_2) and $(n_1, n_2 + M)$ to be subtracted from data $(n_1 + M, n_2)(n_1 + M, n_2 + M)$ can be defined as the element $D(n_1, n_2)$ of E matrix. Hence when the sequences are odd even, the D matrix is involved in the computation.

Hence proved the theorem 5.3

Appendix **E**

Modular Arithmetic

E.1 Definition of Congruent Modulo

It is said that two integers x and y are *congruent modulo* m , where m is also a positive integer, if $y - x$ is divisible by m .

$$x \equiv y \pmod{m} \iff x - y = m \cdot k \quad (\text{E.1})$$

k is some integer.

The equivalence relation E.1 has many solutions and the smallest solution is $x \equiv y \pmod{m}$.

E.2 Inverses in Modular Arithmetic

An inverse to x *modulo* m is a integer is integer y such that,

$$xy \equiv 1 \pmod{m} \quad (\text{E.2})$$

Let x and m be positive integers such that $\gcd(x, m) = 1$. Then x has an inverse modulo m , and it is unique.

Appendix **F**

4×4 SMRT coefficients

F.1 Visual Representation of 4×4 SMRT

The visual representation of 4×4 SMRT coefficients is shown in Fig. F.1.

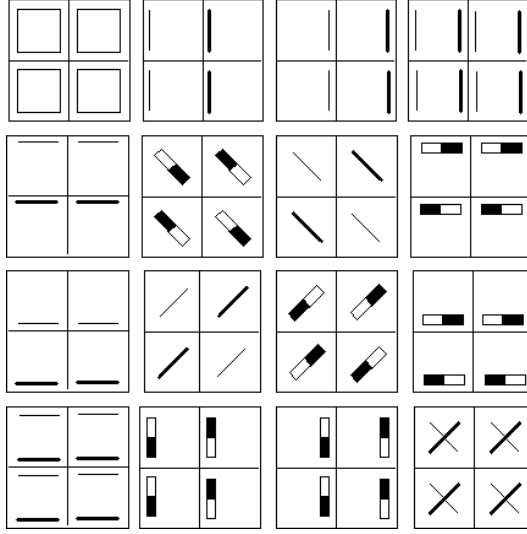


Fig. F.1: Visual Representation of 4×4 SMRT based on 2×2 Data

F.2 Algorithm for Computation of 4×4 SMRT based on 2×2 Data

In the algorithm 4×4 input data matrix, X is partitioned into 2×2 matrices and S is the output 4×4 SMRT matrix.

Computations in L1

L1G1

For $0 \leq i, j < M = 2$

$$LS(i, j) = x(2i, 2j) + x(2i + 1, 2j)$$

$$RS(i, j) = x(2i, 2j + 1) + x(2i + 1, 2j + 1)$$

$$\begin{aligned}
TS(i, j) &= x(2i, 2j) + x(2i, 2j + 1) \\
BS(i, j) &= x(2i + 1, 2j) + x(2i + 1, 2j + 1) \\
DiS(i, j) &= x(2i, 2j) + x(2i + 1, 2j + 1) \\
CS(i, j) &= x(2i + 1, 2j) + x(2i, 2j + 1)
\end{aligned}$$

L1G2

For $0 \leq i, j < M = 2$

$$\begin{aligned}
LD(i, j) &= x(2i, 2j) - x(2i + 1, 2j) \\
RD(i, j) &= x(2i, 2j + 1) - x(2i + 1, 2j + 1) \\
TD(i, j) &= x(2i, 2j) - x(2i, 2j + 1) \\
BD(i, j) &= x(2i + 1, 2j) - x(2i + 1, 2j + 1) \\
DiD(i, j) &= x(2i, 2j) - x(2i + 1, 2j + 1) \\
CD(i, j) &= -x(2i + 1, 2j) + x(2i, 2j + 1)
\end{aligned}$$

Computations in L2

L2G1

For $0 \leq i < M = 2$

$$\begin{aligned}
CLS(i) &= \sum_{j=0}^1 LS(j, i) & CRS(i) &= \sum_{j=0}^1 RS(j, i) \\
CLD(i) &= \sum_{j=0}^1 LD(j, i) & CRD(i) &= \sum_{j=0}^1 RD(j, i)
\end{aligned}$$

L2G2

For $0 \leq i < M = 2$

$$\begin{aligned}
RTS(i) &= \sum_{j=0}^1 TS(i, j) & RTD(i) &= \sum_{j=0}^1 TD(i, j) \\
RBS(i) &= \sum_{j=0}^1 BS(i, j) & RBD(i) &= \sum_{j=0}^1 BD(i, j)
\end{aligned}$$

L2G3

For $0 \leq i, j < M = 2$

$$\begin{aligned} SLR(i, j) &= LS(i, j) + RS(i, j) & DTB(i, j) &= LD(i, j) + RD(i, j) \\ DLR(i, j) &= LS(i, j) - RS(i, j) & DDiC(i, j) &= LD(i, j) - RD(i, j) \end{aligned}$$

L2G4

For $0 \leq i, j < M = 2$

$$\begin{aligned} L(i, j) &= DiS(i, j) & O(i, j) &= DiD(i, j) \\ P(i, j) &= CS(i, j) & Q(i, j) &= CD(i, j) \end{aligned}$$

Computations in L3

L3G1

Sequency 0,0

$$S(0, 0) = \sum_{i=0}^1 \sum_{j=0}^1 SLR(i, j)$$

Sequency 0,2

$$S(0, 3) = \sum_{i=0}^1 \sum_{j=0}^1 DRL(i, j)$$

Sequency 2,0

$$S(3, 0) = \sum_{i=0}^1 \sum_{j=0}^1 DTB(i, j)$$

Sequency 2,2

$$S(3, 3) = \sum_{i=0}^1 \sum_{j=0}^1 DDiC(i, j)$$

L3G2

Sequency 0,1

$$S(0,1) = CLS(0) - CLS(1) \quad S(0,2) = CRS(0) - CRS(1)$$

Sequency 1,0

$$S(1,0) = RTS(0) - RTS(1) \quad S(2,0) = RBS(0) - RBS(1)$$

Sequency 1,2

$$S(1,3) = RTD(0) - RTD(1) \quad S(2,3) = RBD(0) - RBD(1)$$

Sequency 2,1

$$S(3,1) = CLD(0) - CLD(1) \quad S(3,2) = CRD(0) - CRD(1)$$

L3G3

Sequency 1,1

$$S(1,1) = DiD(0,0) - DiD(0,1) + DiD(1,0) - DiD(1,1)$$

$$S(1,2) = DiS(0,0) - DiS(0,1) + DiS(1,0) - DiS(1,1)$$

$$S(2,1) = CD(0,0) - CD(0,1) + CD(1,0) - CD(1,1)$$

$$S(2,2) = CS(0,0) - CS(0,1) + CS(1,0) - CS(1,1)$$

F.3 Algorithm for Computation of 4×4 SMRT based on M-spacing Data

For $0 \leq i < 3, 0 \leq j < 1$,

$$A(i,j) = x(i,j) + x(i,j+2), \quad B(i,j) = x(i,j) - x(i,j+2)$$

L2

For $0 \leq i, j \leq 1$

$$\begin{aligned} C(i, j) &= A(i, j) + A(i + 2, j), & E(i, j) &= B(i, j) + B(i + 2, j) \\ D(i, j) &= A(i, j) - A(i + 2, j), & F(i, j) &= B(i, j) - B(i + 2, j) \end{aligned}$$

L3

L3G1

Sequency 00

$$S(0, 0) = C(0, 0) + C(1, 0) + C(0, 1) + C(1, 1)$$

Sequency 02

$$S(0, 3) = C(0, 0) + C(1, 0) - C(0, 1) - C(1, 1)$$

Sequency 20

$$S(3, 0) = C(0, 0) + C(0, 1) - C(1, 0) - C(1, 1)$$

Sequency 22

$$S(3, 3) = C(0, 0) - C(0, 1) + C(1, 1) - C(1, 0)$$

L3G2

Sequency 01

$$S(0, 1) = E(0, 0) + E(1, 0)$$

$$S(0, 2) = E(0, 1) + E(1, 1)$$

Sequency 21

$$S(3, 1) = E(0, 0) - E(1, 0)$$

$$S(3, 2) = E(0, 1) - E(1, 1)$$

L3G3

Sequency 10

$$S(1, 0) = D(0, 0) + D(0, 1)$$

$$S(2, 0) = D(1, 0) + D(1, 1)$$

Sequency 12

$$S(1, 3) = D(0, 0) - D(0, 1)$$

$$S(2, 3) = D(1, 0) - D(1, 1)$$

L3G4

Sequency 11

$$S(1, 1) = F(0, 0) - F(1, 1)$$

$$S(2, 1) = F(0, 1) + F(1, 0)$$

$$S(1, 2) = F(0, 0) + F(1, 1)$$

$$S(2, 2) = F(0, 1) - F(1, 0)$$

Appendix **G**

Computational Illustration of $N \times N$ Hybrid SMRT Architecture, N a power of 2

The algorithmic steps of $N \times N$ SMRT computation based on hybrid architecture is illustrated using an 8×8 Matrix:

Layer 0

1	2	3	2	1	0	2	1
0	3	1	3	2	1	1	1
1	2	3	0	0	2	1	3
3	1	0	2	0	1	1	3
0	2	3	1	2	1	2	1
1	2	0	1	3	1	0	1
1	1	2	2	0	0	2	2
2	2	0	0	1	3	1	2

Input Matrix, X
L1G1

Fig. G.1: Input 8×8 Matrix

Layer 1**L1G1**

$$LS = \begin{bmatrix} 1 & 4 & 3 & 3 \\ 4 & 3 & 0 & 2 \\ 1 & 3 & 5 & 2 \\ 3 & 2 & 1 & 3 \end{bmatrix} \quad RS = \begin{bmatrix} 5 & 5 & 1 & 2 \\ 3 & 2 & 3 & 6 \\ 4 & 2 & 2 & 2 \\ 3 & 2 & 3 & 4 \end{bmatrix} \quad TS = \begin{bmatrix} 3 & 5 & 1 & 3 \\ 3 & 3 & 2 & 4 \\ 2 & 4 & 3 & 3 \\ 2 & 4 & 0 & 4 \end{bmatrix} \quad BS = \begin{bmatrix} 3 & 4 & 3 \\ 4 & 2 & 1 \\ 3 & 1 & 4 \\ 4 & 0 & 4 \end{bmatrix}$$

L1G2

$$LD = \begin{bmatrix} 1 & 2 & -1 & 1 \\ -2 & 3 & 0 & 0 \\ -1 & 3 & -1 & 2 \\ -1 & 2 & -1 & 1 \end{bmatrix} \quad RD = \begin{bmatrix} -1 & -1 & -1 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 2 & -3 & 0 \end{bmatrix} \quad TD = \begin{bmatrix} -1 & 1 \\ -1 & 3 \\ -2 & 2 \\ 0 & 0 \end{bmatrix}$$

$$BD = \begin{bmatrix} -3 & -2 & 1 & 0 \\ 2 & -2 & -1 & -2 \\ -1 & -1 & 2 & -1 \\ 0 & 0 & -2 & -1 \end{bmatrix}$$

L1G3

$$A = \begin{bmatrix} 2 & 2 & 5 & 3 \\ 2 & 4 & 2 & 4 \\ 1 & 4 & 4 & 3 \\ 3 & 2 & 1 & 5 \\ 2 & 3 & 5 & 2 \\ 4 & 3 & 0 & 2 \\ 1 & 1 & 4 & 4 \\ 3 & 5 & 1 & 2 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 2 & 1 & 1 \\ -2 & 2 & 0 & 2 \\ 1 & 0 & 2 & -3 \\ 3 & 0 & -1 & -1 \\ -2 & 1 & 1 & 0 \\ -2 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & -1 & -1 & -2 \end{bmatrix}$$

Layer 2

L2G1

$$SLR = LS + RS = \begin{bmatrix} 6 & 9 & 4 & 5 \\ 7 & 5 & 3 & 8 \\ 5 & 5 & 7 & 4 \\ 6 & 4 & 4 & 7 \end{bmatrix} \quad DLR = LS - RS = \begin{bmatrix} -4 & -1 & 2 & 1 \\ 1 & 1 & -3 & -4 \\ -3 & 1 & 3 & 0 \\ 0 & 0 & -2 & -1 \end{bmatrix}$$

$$DTB = LD + RD = \begin{bmatrix} 0 & 1 & -2 & 1 \\ -1 & 1 & 1 & 0 \\ -1 & 3 & -1 & 2 \\ -2 & 4 & -4 & 1 \end{bmatrix} \quad DDiC = LD - RD = \begin{bmatrix} 2 & 3 & 0 & 1 \\ -3 & 5 & -1 & 0 \\ -1 & 3 & -1 & 2 \\ 0 & 0 & 2 & 1 \end{bmatrix}$$

L2G2

$$CLS = \text{column sum of } LS = [9 \quad 12 \quad 9 \quad 10]$$

$$CRS = \text{column sum of } RS = [15 \quad 11 \quad 9 \quad 14]$$

$$CLD = \text{column sum of } LD = [-3 \quad 10 \quad -3 \quad 4]$$

$$CRD = \text{column sum of } RD = [-1 \quad -1 \quad -3 \quad 0]$$

L2G3

$$RTS = \text{row sum of } TS = [12 \quad 12 \quad 12 \quad 10]$$

$$RTD = \text{column sum of } TD = [2 \quad -2 \quad 2 \quad 0]$$

$$RBS = \text{column sum of } BS = [12 \quad 11 \quad 9 \quad 11]$$

$$RBD = \text{column sum of } BD = [-4 \quad -3 \quad -1 \quad -3]$$

L2G4

$$C = \begin{bmatrix} 4 & 5 & 10 & 5 \\ 6 & 7 & 2 & 6 \\ 2 & 5 & 8 & 7 \\ 6 & 7 & 2 & 7 \end{bmatrix} \quad D = \begin{bmatrix} 0 & -1 & 0 & 1 \\ -2 & 1 & 2 & 2 \\ 0 & 3 & 0 & -1 \\ 0 & -3 & 0 & 3 \end{bmatrix} \quad E = \begin{bmatrix} -2 & 3 & 2 & 1 \\ -4 & 3 & 0 & 2 \\ 2 & 1 & 2 & -3 \\ 4 & -1 & -2 & -3 \end{bmatrix}$$

$$F = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & -1 & 2 & -3 \\ 2 & 1 & 0 & 1 \end{bmatrix}$$

Layer 3

L3G1

Computation of elements belonging to sequency combinations (0,0), (0,M), (M,0) and (M,M)

For sequency combination (0,0), the row and column index of SMRT element is $n_1, n_2=[0,0]$

Hence, $S(0,0)$ = sum of all elements of SLR matrix = 89.

For sequency combination (0,M), the row and column index of SMRT element is $n_1, n_2=[0, N-1]$

Hence, $S(0,N-1)$ = sum of all elements of DLR matrix = -9.

For sequency combination (M,0), the row and column index of SMRT element is $n_1, n_2=[N-1,0]$

Hence, $S(N-1,0)$ = sum of all elements of DTB matrix = 3.

For sequency combination (M,M), the row and column index of SMRT element is $n_1, n_2=[N-1,N-1]$

Hence, $S(N-1,N-1)$ = sum of all elements of SLR matrix = 13.

L3G2

The algorithm depicted in fig. 6.2 is illustrated using the following steps:

Computation of elements belonging to Sequencies $\{(0,1), (0,2), \dots, (0,2^{v-1})\}$, $\{(1,0), (2,0), \dots, (2^{v-1},0)\}$, $\{(M,1), (M,2), \dots, (M,2^{v-1})\}$, $\{(1,M), (2,M), \dots, (2^{v-1},M)\}$

Initialization:

$M=4$

$v = \log_2 M = 2$

Case 1: Sequencies $\{(0,1), (0,2), \dots, (0,2^{v-1})\}$

For sequency combination (0,1)

Initialization:

$$c_1 = 0, c_2 = 1$$

$$n_1, n_2 = (0,1)$$

$$\text{Row packet size, } s = \frac{M}{c_2} = \frac{4}{1} = 4$$

$$\text{No. of iterations in loop 1, } i = \frac{s}{2} = 2$$

k varies from 0 to i-1

$$\text{No. iterations in loop 2} = c_2 = 1$$

loop1, iteration 1, k=0:

$$S(0, 1) = CLS(k) - CLS(k + \frac{M}{2c_2} - 1) = CLS(0) - CLS(2) = 9 - 9 = 0$$

$$S(0, 2) = CRS(k) - CRS(k + \frac{M}{2c_2} - 1) = CRS(0) - CRS(2) = 15 - 9 = 6$$

$$n_2 = 1 + 2 = 3$$

loop1, iteration 2=i, k=i-1=1:

$$S(0, 3) = CLS(k) - CLS(k + \frac{M}{2c_2} - 1) = CLS(1) - CLS(3) = 12 - 10 = 2$$

$$S(0, 4) = CRS(k) - CRS(k + \frac{M}{2c_2} - 1) = CRS(1) - CRS(3) = 11 - 14 = -3$$

For sequency combination (0,2)

Initialization:

$$c_1 = 0, c_2 = 2$$

$$n_1, n_2 = (0,5)$$

$$\text{Row packet size, } s = \frac{M}{c_2} = \frac{4}{1} = 2$$

$$\text{No. of iterations in loop 1, } i = \frac{s}{2} = 1$$

k varies from 0 to i-1

$$\text{No. iterations in loop 2} = c_2 = 2$$

loop1, iteration 1, k=0:

loop2, iteration 1

$$S(0, 5) = CLS(k) - CLS(k + \frac{M}{2c_2} - 1) = CLS(0) - CLS(1) = 9 - 12 = -3$$

$$S(0, 6) = CRS(k) - CRS(k + \frac{M}{2c_2} - 1) = CRS(0) - CRS(1) = 15 - 11 = 4$$

$$k = k + \frac{M}{2c_2} = 0 + 2 = 2$$

loop2, iteration 1

$$S(0, 5) = S(0, 5) + CLS(k) - CLS(k + \frac{M}{2c_2} - 1) = -3 + CLS(2) -$$

$$CLS(4) = -3 + 9 - 10 = -4$$

$$S(0, 6) = S(0, 6) + CRS(k) - CRS(k + \frac{M}{2c_2} - 1) = 4 + CRS(2) -$$

$$CRS(4) = 4 + 9 - 14 = -1$$

In a similar manner the elements of other sequency combinations in the group is calculated as follows:

Sequency 1,0

$$S(1, 0) = RTS(0) - RTS(2) = 12 - 12 = 0$$

$$S(2, 0) = RBS(0) - RBS(2) = 12 - 9 = 3$$

$$S(3, 0) = RTS(1) - RTS(3) = 12 - 10 = 2$$

$$S(4, 0) = RBS(1) - RBS(3) = 11 - 11 = 0$$

Sequency 2,0

$$S(5, 0) = RTS(0) - RTS(1) + RTS(2) - RTS(3) = 12 - 12 + 12 - 10 = 2$$

$$S(6, 0) = RBS(0) - RBS(1) + RBS(2) - RBS(3) = 12 - 11 + 9 - 11 = -1$$

Sequency 1,4

$$S(1, 7) = RTD(0) - RTD(2) = 2 - 2 = 0$$

$$S(2, 7) = RBD(0) - RBD(2) = -4 + 1 = -3$$

$$S(3, 7) = RTD(1) - RTD(3) = -2 - 0 = -2$$

$$S(4, 7) = RBD(1) - RBD(3) = -3 + 3 = 0$$

Sequency 2,4

$$S(5, 7) = RTD(0) - RTD(1) + RTD(2) - RTD(3) = 2 + 2 + 2 - 0 = 6$$

$$S(6, 7) = RBD(0) - RBD(1) + RBD(2) - RBD(3) = -4 + 3 - 1 + 3 = 1$$

Sequency 4,1

$$S(7, 1) = CLD(0) - CLD(2) = -3 + 3 = 0$$

$$S(7, 2) = CRD(0) - CRD(2) = -1 + 3 = 2$$

$$S(7, 3) = CLD(1) - CLD(3) = 10 - 4 = 6$$

$$S(7, 4) = CRD(1) - CRD(3) = -1 - 0 = -1$$

Sequency 4,2

$$S(7, 5) = CLD(0) - CLD(1) + CLD(2) - CLD(3) = -3 - 10 - 3 - 4 = -20$$

$$S(7, 6) = CRD(0) - CRD(1) + CRD(2) - CRD(3) = -1 + 1 - 3 - 0 = -3$$

L3G3

Sequency 1,1

The general algorithm for computation and placement of elements in L3G3, L3G4, L3G5 and L3G6 is given in section 6.2. L3G3 computes ODD ODD sequency combination. The steps for L3G3 is given below:

- a. Row and column indices of starting element of the sequency packet is initialized, $n_1=1$ and $n_2=1$
- b. Odd sequencies $c_1=1$ and $c_2=1$
- c. Sequency packet size is computed as $\frac{M}{c_1} \times \frac{M}{c_2} = M \times M$
- d. Since $c_1 = c_2$, steps 1 to 14 are performed.
 1. Sequency ratio, $m = \frac{c_2}{c_1} = 1$

2. Column identifier k is initialized as 1 to $\frac{N}{c_2}$ in steps of 2. k=1 to 8 in steps of 2.
3. The row and column indices of elements in the M-spacing matrix, F are calculated as, r=1 and c= $M - \frac{M}{c_2} + 1$ respectively. Hence in step 1, r=0 and c=0.

Iteration 1

4. Check whether column index is in the range of F matrix. c =0 is within range.
5. Sign of F matrix element to be used for the computation of S element is decided from the computation of expression 6.3
 $((r \times c_1(1 + 2 \times i_2) + c \times c_2))_N - c_2 \times i_1 = 0$, hence sign is positive.
6. S(1,1)=F(0,0)
7. $c = c - \frac{M}{c_2} = 0 - M = -M$
8. Steps 4 to 6 are repeated until all F matrix elements in a row involved in the computation of a particular coefficient are completed. As $c_1 = 1$, only one iteration
9. r=1, c=3
10. Steps 4 to 9 are repeated M times

Iteration 2

4. c is within range
5. sign: r=1, c=3, $c_1=1$, $c_2=1$, $i_1=0$ and $i_2=0$. Hence expression 6.3 is M and hence sign is negative.
6. S(1,1)=S(1,1)-F(1,3)
- 7 and 8 same as above.
9. r=2, c=6

Iteration 3

4. c not in range, hence c= c-M=6-4=2
5. sign:r=2, c=2, $c_1=1$, $c_2=1$, $i_1=0$ and $i_2=0$. Hence expression 6.3 is M and hence sign is negative
- 6.S(1,1)=S(1,1)-F(2,2)
- 7 and 8 same as above.
9. r=3, c=5

Iteration 4

4. c not in range, hence c= c-M=5-4=1

5. sign:r=3, c=1, c₁=1, c₂=1, i₁=0 and i₂=0. Hence expression

6.3 is M and hence sign is negative

$$6.S(1,1)=S(1,1)-F(3,1)$$

7 and 8 same as above.

$$9.r =4, c=8$$

10. M iterations completed.

S(1,1) = F(0,0)-F(1,3)-F(2,2)-F(3,1) is caclulated.

$$S(1,1) = F(0,0) - F(1,3) - F(2,2) - F(3,1) = 2 - 2 - 2 - 1 = -3$$

11. Steps to compute the next element in the sequency packet

n₁=2, n₂=1. S(2,1) is computed as

$$S(2,1) = F(0,1)+F(1,0)-F(2,3)-F(3,2) = 1+0-(-3)-0 = 4$$

$$S(3,1) = F(0,2) + F(1,1) + F(2,0) - F(3,3) = 0 + 1 + 0 - 1 = 0$$

$$S(4,1) = F(0,3) + F(1,2) + F(2,1) + F(3,0) = 1 + 0 - 1 + 2 = 2$$

$$S(1,2) = F(0,0) - F(1,1) + F(2,2) - F(3,3) = 2 - 1 + 2 - 1 = 2$$

$$S(2,2) = F(0,1) - F(1,2) + F(2,3) + F(3,0) = 1 - 0 - 3 + 2 = 0$$

$$S(3,2) = F(0,2) - F(1,3) - F(2,0) + F(3,1) = 0 - 2 - 0 + 1 = -1$$

$$S(4,2) = F(0,3)+F(1,0)-F(2,1)+F(3,2) = 1+0-(-1)+0 = 2$$

$$S(1,3) = F(0,0) + F(1,3) - F(2,2) + F(3,1) = 2 + 2 - 2 + 1 = 3$$

$$S(2,3) = F(0,1)-F(1,0)-F(2,3)+F(3,2) = 1-0-(-3)+0 = 4$$

$$S(3,3) = F(0,2) - F(1,1) + F(2,0) + F(3,3) = 0 - 1 + 0 + 1 = 0$$

$$S(4,3) = F(0,3) - F(1,2) + F(2,1) - F(3,0) = 1 - 0 - 1 - 2 = -2$$

$$S(1,4) = F(0,0) + F(1,1) + F(2,2) + F(3,3) = 2 + 1 + 2 + 1 = 6$$

$$S(2,4) = F(0,1) + F(1,2) + F(2,3) - F(3,0) = 1 + 0 + (-3) - 2 = -4$$

$$S(3,4) = F(0,2) + F(1,3) - F(2,0) - F(3,1) = 0 + 2 - 0 - 1 = 1$$

$$S(4,4) = F(0,3) - F(1,0) - F(2,1) - F(3,2) = 1 - 0 - (-1) - 0 = 2$$

Similar steps are involved in the computation of L3G4, L3G5 and L3G6.

L3G4

Sequency 1,2

$$S(1,5) = D(0,0) - D(0,2) - D(2,1) + D(2,3) = 0 - 0 - 3 + (-1)$$

$$S(2,5) = D(1,0) - D(1,2) - D(3,1) + D(3,3) = -2 - 2 - (-3) + 3 = 2$$

$$S(3,5) = D(0,1) - D(0,3) - D(2,2) + D(2,0) = -1 - 1 - 0 + 0 = -2$$

$$S(4,5) = D(1,1) - D(1,3) + D(3,0) - D(3,2) = 1 - 2 + 0 - 0 = -1$$

$$S(1,6) = D(0,0) - D(0,2) + D(2,1) - D(2,3) = 0 - 0 + 3 - (-1) = 4$$

$$S(2,6) = D(1,3) - D(1,1) + D(3,0) - D(3,2) = 2 - 1 + 0 - 0 = 1$$

$$S(3,6) = D(0,1) - D(0,3) - D(2,0) + D(2,2) = -1 - 1 - 0 + 0 = -2$$

$$S(4,6) = D(1,0) - D(1,2) + D(3,1) - D(3,3) = -2 - 2 + (-3) - 3 = -10$$

L3G5

Sequency 2,1

$$S(5,1) = E(0,0) - E(1,2) - E(2,0) + E(3,2) = -2 - 0 - 2 + (-2) = -6$$

$$S(5,2) = E(0,1) - E(1,3) - E(2,1) + E(3,3) = 3 - 2 - 1 + (-3) = -3$$

$$S(5, 3) = E(0, 2) + E(1, 0) - E(2, 2) - E(3, 0) = 2 + (-4) - 2 - 4 = -8$$

$$S(5, 4) = E(0, 3) + E(1, 1) - E(2, 3) - E(3, 1) = 1 + 3 - (-3) + (-1) = 8$$

$$S(6, 1) = E(0, 0) + E(1, 2) - E(2, 0) - E(3, 2) = -2 + 0 - 2 - (-2) = -2$$

$$S(6, 2) = E(0, 1) + E(1, 3) - E(2, 1) - E(3, 3) = 3 + 2 - 1 - (-3) = 7$$

$$S(6, 3) = E(0, 2) - E(1, 0) - E(2, 2) + E(3, 0) = 2 - (-4) - 2 + 4 = 8$$

$$S(6, 4) = E(0, 3) - E(1, 1) - E(2, 3) + E(3, 1) = 1 - 3 - (-3) + (-1) = 0$$

L3G6 Sequency 2,2

$$S(5, 5) = C(0, 0) - C(0, 2) - C(1, 1) + C(1, 3) - C(2, 0) + C(2, 2) + C(3, 1) - C(3, 3) = 4 - 10 - 7 + 6 - 2 + 8 + 7 - 7 = -1$$

$$S(6, 5) = C(0, 1) - C(0, 3) + C(1, 0) - C(1, 2) - C(2, 1) + C(2, 3) - C(3, 0) + C(3, 2) = 5 - 5 + 6 - 2 - 5 + 7 - 6 + 2 = 2$$

$$S(5, 6) = C(0, 0) - C(0, 2) + C(1, 1) - C(1, 3) - C(2, 0) + C(2, 2) - C(3, 1) + C(3, 3) = 4 - 10 + 7 - 6 - 2 + 8 - 7 + 7 = 1$$

$$S(6, 6) = C(0, 1) - C(0, 3) - C(1, 0) + C(1, 2) - C(2, 1) + C(2, 3) + C(3, 0) - C(3, 2) = 5 - 5 - 6 + 2 - 5 + 7 + 6 - 2 = 2$$

The L3 output is as given below:

L3

89	0	6	2	-3	-4	-1	-9
0	-3	2	3	6	-4	4	0
3	4	0	4	-4	2	1	3
2	0	-1	0	1	-2	-2	-2
0	2	2	-2	2	-1	-10	0
2	-6	-3	-8	8	-1	1	6
-1	-2	7	8	0	2	2	1
3	0	2	6	-1	-20	-3	13

SMRT Matrix, S

Fig. G.2: Output SMRT Matrix

Appendix H

Shift invariance in sequence packets of SMRT

The shift property of SMRT matrix is analyzed by shifting the data row wise/column wise. After shifting the original data row wise/ column wise and analyzing the sequency packets (fig: H.1), it is clear that shifting has predicatble effect inside the sequency packets. This will be more clear from the fig: H.2.

The shift property of SMRT can be concluded as follows:

Suppose c_1, c_2 are the sequencies and if row shift is indicated by m and column shift by n , (where m positive for right shift and negative for left shift. Similarly n positive for down shift and negative for up shift.) then the shift in sequency packet of SMRT matrix is $(c_1 \times m + c_2 \times n)$. When $c_1 > c_2$, shift is row wise and $c_1 < c_2$, shift is column wise. As it is a shift variant transform it can be compared with other shift variant transforms like wavelet transforms.

0,0	0,1	0,2	0,4
1,0	1,1	1,2	1,4
2,0	2,1	2,2	2,4
4,0	4,1	4,2	4,4

Fig. H.1: Sequency Packets in 8x8 SMRT

44	78	47	45	48	56	60	79
142	137	150	119	136	144	142	162
191	235	204	223	215	231	212	225
202	215	205	216	206	235	212	227
209	243	205	236	205	230	217	241
204	223	198	206	214	217	211	176
196	227	199	226	212	220	190	189
189	167	182	203	197	201	196	161

8x8 input matrix

1163	-56	-9	-50	14	-20	125	-353
-132	66	41	-50	-91	7	-21	55
-517	57	-47	39	-13	-26	41	3
77	31	94	-45	-60	-15	43	-27
222	-7	-3	-57	-49	-75	-20	-100
-115	24	68	-18	-21	-79	15	-16
-433	56	-20	-34	-127	-46	-40	49
-357	-24	101	2	-22	-8	-13	-307

SMRT matrix for the 8x8 matrix

189	167	182	203	197	201	196	161
44	78	47	45	48	56	60	79
142	137	150	119	136	144	142	162
191	235	204	223	215	231	212	225
202	215	205	216	206	235	212	227
209	243	205	236	205	230	217	241
204	223	198	206	214	217	211	176
196	227	199	226	212	220	190	189

8x8 input matrix shifted one row down, m=1, n=0

1163	-56	-9	-50	14	-20	125	-353
-222	7	47	-57	-13	75	-41	100
-132	66	-94	50	-30	7	-43	55
-517	57	3	-39	-49	-26	20	3
77	31	41	49	91	-15	-21	-27
433	18	21	24	68	46	-40	-49
-115	-34	-127	-66	20	-79	-15	-16
357	24	-101	-2	22	8	13	307

SMRT matrix for the shifted 8x8 matrix with m=1, n=0

60	79	44	78	47	45	48	56
142	162	142	137	150	119	136	144
212	225	191	235	204	223	215	231
212	227	202	215	205	216	206	235
217	241	209	243	205	236	205	230
211	176	204	223	198	206	214	217
190	189	196	227	199	226	212	220
196	161	189	167	182	203	197	201

8x8 input matrix shifted one row down, m=0, n=2

1163	60	-14	-56	-9	20	-125	-353
-132	-31	-94	49	30	-7	21	55
-517	7	3	57	49	26	-41	3
77	66	41	-50	-91	15	-43	-27
222	57	-47	39	-13	75	20	-100
-115	18	21	24	68	79	-15	-16
-433	34	127	66	-20	46	40	49
-357	-2	22	-24	101	8	13	-307

SMRT matrix for the shifted 8x8 matrix with m=0, n=2

137	150	119	136	144	142	162	142
235	204	223	215	231	212	225	191
215	205	216	206	235	212	227	202
243	205	236	205	230	217	241	209
223	198	206	214	217	211	176	204
227	199	226	212	220	190	189	196
167	182	203	197	201	196	161	189
78	47	45	48	56	60	79	44

8x8 input matrix shifted one row down, m=1, n=1

1163	-9	-50	14	66	125	20	353
-517	31	-41	49	-91	-75	-41	-3
77	-7	47	57	-13	-7	-43	27
222	-66	-94	-50	-30	26	20	100
1329	-57	3	39	-49	15	-21	55
-433	-21	-24	-68	18	79	15	-49
1152	127	56	-20	-34	46	-40	-16
357	-101	-2	22	-24	13	-8	-307

SMRT matrix for the shifted 8x8 matrix with m=1, n=1

Fig. H.2: Shift variance in sequency packets of 8x8 SMRT

