

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600



Université d'Ottawa • University of Ottawa

Wavelet Based Coding and Indexing of Images and Video

Mrinal Kumar Mandal, M.A.Sc

A Thesis

submitted to the School of Graduate Studies and Research

in partial fulfillment of the requirements

for the Degree of

Ph.D. in Electrical Engineering

Ottawa-Carleton Institute of Electrical and Computer Engineering

School of Information Technology and Engineering

Faculty of Engineering

University of Ottawa

October, 1998

©Mrinal Kumar Mandal, Ottawa, Canada



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-36782-7

To my family

Acknowledgements

It is my pleasure to acknowledge and thank all persons who have influenced me in the course of this research. First, I express my gratitude to both of my advisors Dr. Sethuraman Panchanathan and Dr. Tyseer Aboulnasr for introducing me to the exciting field of wavelets and image compression and for their continued support and encouragement during my thesis work.

I wish to express my gratitude to Prof. K. R. Rao, Department of Electrical Engineering, University of Texas at Arlington, for examining the thesis meticulously, and giving valuable comments.

Special thanks are due to Dr. D. Coll, Department of Systems and Computer Engineering, Carleton University and Dr. T. Yeap, School of Information Technology and Engineering, University of Ottawa, whose valuable comments have helped in the progress of this thesis.

I would like to thank all the past and present members of the Visual Computing and Communications Laboratory, especially Fayez Idris, and Nadjia Gamaz for their help and cooperation.

My special thanks are due to all the support staff members of School of Information Technology and Engineering for their help, especially Michele Roy and Lucette Lepage.

The generous financial support of Canadian Commonwealth Fellowship Plan and NSERC that made this research possible is also gratefully acknowledged.

I am truly grateful to my beloved wife Rupasri and my family for their consistent support, without which this work would not have been possible.

Abstract

The multitude of visual media based applications demands sophisticated compression and indexing techniques for efficient storage, transmission, and retrieval of images and video. Wavelet transform has emerged as a powerful tool for efficient compression of images and video sequences. However, there is a need for significant enhancements in the motion estimation process in order to design an efficient wavelet-based video coder. More importantly, there has been little work done in the area of image and video indexing in the wavelet domain. These are crucial for consideration of wavelets as a potential candidate for multimedia applications. Hence there is an impending need for investigating joint compression and indexing approaches in the wavelet transform domain which is the principal focus of this thesis.

In this thesis, we have proposed several novel coding and indexing techniques in the wavelet domain. The proposed coding techniques emphasize efficient motion estimation techniques for a wavelet-based video coder, which include bi-directional motion estimation, fine-to-coarse motion estimation, and variable block-size motion estimation. The proposed indexing techniques include moment-based indexing, fast wavelet histogram indexing, indexing based on distribution of wavelet coefficients, illumination invariant indexing, and robust video segmentation. An efficient video storage and archival system has been developed by combining all proposed techniques. The novelty of the coding and indexing techniques is that both employ a set of key features that describe the content of the visual data. This results in superior performance with lower storage space requirement, and reduced computational complexity.

Table of Contents

Acknowledgements		iii
Abstract		iv
Table of Contents		v
List of Figures		x
List of Tables		xv
List of Abbreviations		xvii
List of Mathematical Symbols and Notations		xx
1. Introduction		1
1.1 Motivation and Problem Statement		3
1.1.1 Motivation		3
1.1.2 Problem Statement		4
1.2 Thesis Contributions		6
1.2.1 Publications		8
1.3 Outline of the Thesis		8
2. Review of Image and Video Compression Techniques		10
2.1 Digital Image and Video Signal		11
2.1.1 Image Model		11
2.1.2 Video Data Formats		12
2.2 Image Compression Techniques		12
2.2.1 Fundamentals of Data Compression Techniques		12
2.2.2 Popular Image Coding Techniques		14
2.2.3 Image Compression Standards		17
2.3. Video Compression Techniques		18
2.3.1 Conventional Motion Estimation Techniques		19
2.3.1.1 Evaluation Criteria for Motion Estimation Techniques		22
2.3.2 Video Compression Standards		23

2.4 Wavelets in Image and Video Compression	26
2.4.1 Theory of Wavelets/Subbands	26
2.4.2 Wavelet Coding of Images	32
2.4.3 Wavelet-based Video Coding	37
2.4.4 Evaluation of Coding Performance	42
2.5 Summary	43
3. Review of Image and Video Indexing Techniques	44
3.1 Image Indexing in Pixel Domain	46
3.1.1 Histogram	47
3.1.2 Color	49
3.1.3 Texture	50
3.1.4 Shape/Sketch	51
3.1.5 Spatial Relationships	52
3.1.6 Moments	52
3.1.6.1 1-D Moments of Histogram	53
3.1.6.2 2-D Moments	55
3.2 Image Indexing in the Compressed Domain	56
3.2.1 Discrete Cosine Transform	58
3.2.2 Subband/Wavelet	59
3.3 Illumination Invariant Indexing	62
3.4 Video Indexing in Pixel Domain	63
3.4.1 Video Segmentation in Pixel Domain	64
3.5 Video Indexing in Compressed Domain	66
3.5.1 Video Segmentation using DCT Coefficients	67
3.5.2 Video Segmentation using Subband Coefficients	67
3.5.3 Video Segmentation using Motion Vectors	68
3.6 Feature Similarity	70
3.7 Evaluation Criteria for Image Indexing Techniques	73
3.8 Integrated Coding and Indexing	75

3.9 Summary	76
4. Video Compression using Wavelets	77
4.1 Baseline Video Coder	78
4.1.1 Bit Allocation and Quantization	79
4.1.1.1 Scheme-1	80
4.1.1.2 Scheme-2	81
4.1.1.3 Scheme-3	81
4.1.1.4 Performance Evaluation of Schemes 1-3	83
4.2 Translation Variance of DWT Coefficients	85
4.3 Proposed Adaptive Thresholding (AMRME) Technique	89
4.3.1 Coding Performance	91
4.4 Bi-directional Motion Estimation (BMRME) Technique	94
4.4.1 Technique-1 (BMRME-1)	95
4.4.2 Technique-2 (BMRME-2)	97
4.4.3 Computational Complexity of BMRME Techniques	99
4.4.4 Performance of BMRME Techniques	102
4.5 Adaptive Motion Estimation Techniques	104
4.5.1 Adjustable Resolution Selection (ARS) Technique	105
4.5.1.1 Motion Estimation Performance	108
4.5.1.2 Choice of Threshold Factors	111
4.5.1.3 Coding Performance	114
4.5.2 Adaptive Bit Allocation (ABA) Technique	116
4.5.2.1 Coding Performance of the ABA Technique	120
4.5.3 Bi-directional ARS/ABA Technique	122
4.6 Comparison of the Proposed Techniques	124
4.7 Summary	128
5. Image Indexing using Histograms, Moments and Wavelets	129
5.1 Indexing by Histogram and Moments	130
5.1.1 Evaluation of Histogram/Moment-based Technique	130
5.1.1.1 Difference of Image Histogram	130
5.1.1.2 Difference of Reconstructed Histograms	133

5.1.1.3 Difference of Moments of Histogram	135
5.1.1.4 Difference of 2-D Moments of Image	138
5.1.1.5 Indexing of Texture Images	140
5.1.2 Histogram Representation by Legendre Moments	143
5.1.3 Comparison of Computational Complexity	147
5.1.4 Section Summary	150
5.2 Image Indexing in the Wavelet Domain	150
5.2.1 Performance of Indexing Techniques based on DWT Coefficients	151
5.2.1.1 Fast Multiresolution Image Querying Technique (FMIQT)	152
5.2.1.2 Wavelet Histogram Technique (WHT)	153
5.2.1.3 Fast Wavelet Histogram Techniques (FWHT)	155
5.2.2 Image Indexing based on Distribution of Wavelet Coefficients	166
5.2.3 Joint Moment and Wavelet Technique	171
5.2.4 Section Summary	176
5.3 Summary	177
6. Illumination Invariant Image Indexing	179
6.1 Translation and Scale Invariant Moments	180
6.2 Wavelet-based Indexing	185
6.3 Joint Moment and Wavelet-based Indexing	189
6.4 Computational Complexity of the Proposed Techniques	190
6.5 Performance of Illumination Invariant Indexing Techniques	192
6.6 Summary	202
7. Joint Coding and Indexing of Video in Wavelet Domain	203
7.1 Video Indexing Requirements	204
7.2 Potential Features for Abrupt Transition Detection	205
7.2.1 Image Histogram and it's Moments	206
7.2.2 DWT Coefficients	207
7.2.3 Wavelet Band Parameters	209
7.2.4 Motion Vectors	210
7.2.4.1 Detecting Abrupt Transition using Motion Vectors	212

7.2.5 Indexing Performance of Individual Features	212
7.3 Fast Video Segmentation	214
7.4 Joint Coding and Indexing	216
7.4.1 Video Coding	216
7.4.2 Video Segmentation and Query Matching	217
7.5 Summary	223
8. Summary, Conclusions and Future Research Directions	224
8.1 Summary and Conclusions	224
8.2 Future Research Directions	226
References	228
Appendix	240
A Wavelet Bases and Filter Coefficients	240
B Image and Video Databases	242

List of Figures

1.1	Overview of the Thesis	7
2.1	Block diagram of a transform coding scheme	16
2.2	Block matching motion estimation process	21
2.3	Block diagram of MPEG video encoder	25
2.4	Example of a group of pictures (GOP) used in MPEG	27
2.5	Schematic of Mallat's tree algorithm	29
2.6	Schematic of 1-D wavelet decomposition and reconstruction	30
2.7	2-D wavelet transform	31
2.8	Three stage wavelet transformed image	32
2.9	Wavelet pyramid of 4 levels	33
2.10	A wavelet-based image coding scheme	35
2.11	A typical wavelet-based video encoder	38
2.12	Multiresolution motion estimation	41
2.13	Typical coding performance of MRME techniques on Pingpong sequence	43
3.1	A schematic of image archival and retrieval system	45
3.2	Various methods in content based image indexing in the pixel domain	46
3.3	An image with five different orientations	48
3.4	Histograms of five images shown in Figure 3.3	49
3.5	Original and reconstructed <i>pdfs</i>	54
3.6	A case where direct comparison of histogram is likely to fail	56
3.7	Block diagram of a compressed domain indexing system	57
3.8	Various methods in content based image indexing in the compressed domain	57

3.9	Image indexing and retrieval using transform coefficients	58
3.10	Wavelet histogram generation	61
3.11	Shot detection by twin comparison technique	66
4.1	A typical wavelet-based video encoder	78
4.2	Block diagram of a typical wavelet-based video decoder	80
4.3	Generalized Gaussian density functions with different parameters	83
4.4	Coding performance of three quantization schemes for various images	84
4.5	Effect of aliasing on the motion estimation in the lowpass and highpass wavelet bands	88
4.6	Coding performance of AMRME techniques for different thresholds	92
4.7	Performance comparison of MRME-Z and AMRME Techniques	94
4.8	Schematic of BMRME-1 technique	97
4.9	Schematic of BMRME-2 technique	100
4.10	Performance comparison of MRME and BMRME techniques	103
4.11	Video encoder employing adjustable resolution selection (ARS) technique followed by AMRME technique	107
4.12	Frames 1 and 2 of Pingpong sequence	109
4.13	Motion vectors corresponding to frame-2 of pingpong sequence, estimated by AMRME technique	110
4.14	Motion vectors corresponding to frame-2 of pingpong sequence, estimated by ARS technique	112
4.15	Choice of threshold factors for fine ME in ARS (L-1) technique	113
4.16	Performance of fine motion estimation using original and reconstructed DWT coefficients	115
4.17	Performance of ARS technique with motion estimation at various levels of wavelet pyramid	116

4.18	Performance comparison of AMRME, MRME C, and ARS techniques	117
4.19	Video coder employing ARS followed by ABA Techniques	120
4.20	Typical motion blocks in ARS, and ABA+ARS technique	121
4.21	Coding performance of ABA+ARS technique	122
4.22	Performance comparison of ARS+MRME and ARS+BMRME techniques	125
4.23	Peak signal-to-noise ratio (PSNR) values of reconstructed sequence	126
4.24	Coding performance of the proposed techniques	127
5.1	Images retrieved using histogram-based technique	131
5.2	Histograms of the images shown in Fig. 5.1a, 5.1e and 5.1i	132
5.3	Performance of different Metrics for indexing	133
5.4	Indexing performance of various moments with L^1/L^2 metric	137
5.5	Indexing performance of various moments with L^2 metric on IDB1	138
5.6	Indexing performance of various moments	139
5.7	Performance of various 2-D moments for indexing	140
5.8	Reconstructed images of Fig. 3.3a using 2-D regular moments	141
5.9	Performance of histogram technique for indexing texture images	141
5.10	Images retrieved using histogram technique - first query	142
5.11	Images retrieved using histogram technique - second query	143
5.12	Legendre polynomials of order 0-15	145
5.13	Performance comparison of regular and Legendre moments for indexing (based on IDB1 image database)	147
5.14	Location of high amplitude DWT coefficients	154
5.15	Schematic of wavelet histogram generation at Level-1	158
5.16	Schematic of wavelet histogram generation at Level-2	159
5.17	Schematic of wavelet histogram generation at Level-3	160

5.18	Wavelet histograms of the image shown in Fig. 3.3a	163
5.19	Wavelet histograms of the image shown in Fig. 5.11a	164
5.20	Comparison of indexing performance of various wavelet histogram techniques	165
5.21	Comparison of band histogram of two different images (Fig. 5.1c and Fig. 5.1e)	168
5.22	Typical histogram of a wavelet highpass subband and its approximation	169
5.23	Comparison of subband parameters of Fig. 5.1c and Fig. 5.1e	170
5.24	Histogram of the original image and the corresponding lowpass subimage of Fig. 3.3a-b	170
5.25	Indexing performance of Legendre moments and wavelet parameters	173
5.26	Images retrieved from IDB2 using LGM+WP technique - first query	174
5.27	Images retrieved from IDB2 using LGM+WP technique - second query	175
5.28	Example of failure of LGM+WP technique	176
5.29	Tolerance (ρ) versus retrieval efficiency (η_R)	177
6.1	An image with different illumination levels and their histograms	181
6.2	Mapping of a gray level histogram [0,255] to Legendre interval [-1,1]	184
6.3	Change of standard deviation parameter with scale factor	188
6.4	Change of shape parameter with scale factor	188
6.5	Retrieval efficiency of various indexing techniques on color {Y} images of IDB3	196
6.6	Retrieval efficiency of various indexing techniques on color {Y,Cb,Cr} images of IDB3	198
6.7	Retrieval efficiency of various indexing techniques on {Y} images of IDB4	198
6.8	Retrieval efficiency of various indexing techniques on color {Y,Cb,Cr} images of IDB4	199
6.9	A typical query image	199

6.10	First six retrieved images from IDB2 employing 3-D RHT technique for query image shown in Fig. 5.38	200
6.11	First six retrieved images from IDB2 employing TSI-LGM+WP technique for query image shown in Fig. 5.38	201
6.12	Tolerance (ρ) versus retrieval efficiency of TSI-LGM+WP technique	201
7.1	A schematic of video archival and retrieval system	205
7.2	DoIH and DoLMH variation with respect to frames in Beverly Hills sequence ..	208
7.3	DoDWTC variation with respect to frames in Beverly Hills sequence	209
7.4	DoSPWB variation with respect to frames in Beverly Hills sequence	210
7.5	DoSDWB variation with respect to frames in Beverly Hills sequence	211
7.6	Flowchart of the proposed fast video segmentation technique in the wavelet domain	217
7.7	Coding performance of Beverly Hills sequence	219
7.8	Representative frames (RF) from Beverly Hills sequence	221
7.9	Query image	222
7.10	Retrieved representative frames from Beverly Hills sequence	222
7.11	Histogram comparison of query image and representative frames 13 and 10	223

List of Tables

2.1	Digital video data formats		12
2.2	Computational complexity of motion estimation algorithms		41
3.1	The capacity of histogram space with a distance threshold T_H		50
4.1	Computational complexity of motion estimation algorithms		101
4.2	Number of motion vectors and TFLAGS in MRME/BMRME algorithms		101
4.3	Computational complexity of various motion estimation techniques		108
4.4	SNR of motion compensation, between frame 1 and 2 of Salesman sequence, at various subbands		111
5.1	SNR of reconstructed pdf's using finite number of moments		135
5.2	Various moments (on grid [0:1:255]) of histogram of image shown in Fig. 3.3a		136
5.3	Various moments (on grid [-1:2/255:1]) of histogram of image shown in Fig. 3.3a		136
5.4	Computational complexity of feature generation for various indexing techniques		148
5.5	Computational complexity of feature comparison for various indexing techniques		149
5.6	Retrieval efficiency of FMIQT and WHT techniques		152
5.7	Complexity of computing wavelet histogram in WHT technique		156
5.8	Complexity of computing wavelet histogram in Level-1		158
5.9	Complexity of computing wavelet histogram in Level-2		159
5.10	Complexity of computing wavelet histogram in Level-3		160
5.11	Complexity of FWHT technique at Level-2		161
5.12	Performance versus complexity of WHT and FWHT techniques		166
5.13	Performance improvement using wavelet band information		171

5.14	Relative weights of various feature parameters employed for performance evaluation	172
6.1	Typical relative weights of various feature parameters	191
6.2	Computational complexity of feature generation for TSI indexing techniques	192
6.3	Computational complexity of feature comparison for TSI indexing techniques	193
6.4	Performance of ratio histogram on IDB1 and IDB2 database of images	195
6.5	Performance of Wavelet Parameter (WP) Technique	196
7.1	Shots with abrupt transition in Beverly Hills sequence	206
7.2	Detection of camera breaks in Beverly Hills sequence using various algorithms	213
7.3	Detection of AT's using motion vectors at different levels	214
7.4	Detection of camera breaks in Beverly Hills sequence using FVST technique	218

List of Abbreviations

ABA	Adaptive Bit Allocation
ARS	Adjustable Resolution Selection
AMRME	Adaptive Multiresolution Motion Estimation
ATM	Asynchronous Transfer Mode
BABS	Biggest allowed block size (for motion estimation)
BHD	Block Histogram Difference Technique for Video Indexing
BVD	Block Variance Difference Technique for Video Indexing
BMRME	Bi-directional Multiresolution Motion Estimation
CCIR	Comite Consultatif International de Radiodiffusion
CCITT	International Consultative Committee for Telephone and Telegraph
CIF	Common Intermediate Format
CNM	Central Moments
CNNM	Central Normalized Moments
DCT	Discrete Cosine Transform
DOCM	Difference of Central Moments
DOCNM	Difference of Central Normalized Moments
DOIH	Difference of Image Histogram
DOIHL	Difference of Image Histogram by Lee's Method
DOIHWP	Difference of Image Histogram by as well as difference of Wavelet Parameters
DOIWH	Difference of Image as well as Wavelet Histogram
DOH	Difference of Histogram Technique for Video Indexing
DOHHWB	Difference of Histograms of Highpass Wavelet Bands
DOLMH	Difference of Legendre Moments of Histograms
DOLMWP	Difference of Legendre Moments as well as Wavelet Parameters.
DST	Discrete Sine Transform
DPCM	Differential Pulse Code Modulation
DFD	Displace Frame Difference

DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
EC	Evaluation Criterion
FLOP	Floating Point Operation
GGD	Generalized Gaussian Distribution
HDTV	High Definition Television
HI	High Illumination
HODF	Histogram of Difference Frame Technique for Video Indexing
HVS	Human Visual System
IDB1	Image Database - 1 (for general image)
IDB2	Image Database - 2 (for texture image)
IDB3	Image Database - 3 (images with different illumination)
IDB4	Image Database - 4 (images with different illumination)
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
ITU	International Telecommunication Union
JPEG	Joint Photographic Experts Group
FMIQT	Indexing Technique proposed by Jacob <i>et al.</i>
KLT	Karhunen-Loeve Transform
LGM	Legendre Moments
LGM+WP	Joint Legendre Moment and Wavelet Parameter Technique
LI	Low Illumination
MAE	Mean Absolute Error
MAD	Mean Absolute Difference
MAXAD	Maximum Allowed Displacement (for motion estimation)
MAXAR	Maximum Allowed Refinements (for ME)
MD	Mahalanabis Distance
MDRGM	Mahalanabis Distance of Regular Moments
ME	Motion Estimation
MPEG	Motion Picture Experts Group

MRME	Multiresolution Motion Estimation
MRME-C	Multiresolution Motion Estimation Technique of Conklin <i>et al.</i>
MRME-U	Multiresolution Motion Estimation Technique of Uz <i>et al.</i>
MRME-Z	Multiresolution Motion Estimation Technique of Zhang <i>et al.</i>
MSE	Mean Square Error
MWHT	Modified Wavelet Histogram Technique
NB	Normal Block
PCM	Pulse Code Modulation
PR	Perfect Reconstruction
PSNR	Peak Signal to Noise Ratio
QMF	Quadrature Mirror Filter
RGM	Regular Moments
RGNM	Regular Normalized Moments
RHT	Ratio Histogram Technique
SB	Special Block
SBC	Subband Coding
SBD	Subband Decomposition
STFT	Short Time Fourier Transform
TCG	Transform Coding Gain
TSI	Translation and Scale Invariant
TSI-M	TSI Moments
TSI-LGM	TSI Legendre Moments
VLSI	Very Large Scale Integration
VQ	Vector Quantization
WH	Wavelet Histogram
WHT	Wavelet Histogram Technique
<i>bpp</i>	bit-per-pixel
<i>pdf</i>	probability density function

Mathematical Notations

a, b	Scale and location parameters of wavelet transform
A_k	Weight of the standard deviation of the k th wavelet band
α	Degree of illumination change
B_k	Weight of the shape parameter of the k th wavelet band
β_k	k th order normalized central moment
β_{Q_k}	k th order normalized central moment of histogram of image Q
β_{pq}	(p, q) th order 2-D normalized central moment
B_F	Buffer Space required to store an Image Frame
C	A candidate image
c	Feature vector corresponding to a candidate image C
$c_{p,q}$	Lowpass DWT coefficients corresponding to p th scale and q th location
$d_{p,q}$	Highpass DWT coefficients corresponding to p th scale and q th location
D	Total distortion (for video compression)
δ_k	Quantization step-size in wavelet band k
Δ	Horizontal/vertical motion search area
η_R	Retrieval efficiency
η_K	k th order TSI regular moment
$f(x)$	An 1-D continuous function
$f(x, y)$	A 2-D continuous function
ϕ_k	k th Hu's moment
$\phi(x)$	1-D scaling function
γ	Shape parameter of GGD function
$\Gamma(.)$	Gamma function
G_{MP}	Gain of motion prediction/compensation
$g[.]$	Highpass filter coefficient corresponding to mother wavelet

$H(\cdot)$	Laplace/z-transform of $h[\cdot]$
$h[\cdot]$	Lowpass filter coefficient corresponding to mother wavelet
$h_i[\cdot]$	Histogram function
I	Original image
$\hat{I}$	Reconstructed image
$i(x, y)$	image function at (x, y) location
$i_{DFD}(\cdot)$	Displaced frame difference
J	Lagrange cost function
K_1	Number of operations equivalent to a logarithm.
K_2	Number of operations equivalent to a convolution.
L_M	Lagrange multiplier
$l(x, y)$	Illumination at (x, y) location
M_k	k th order regular moment
M_{pq}	(p, q) th order 2-D regular moment
$\overline{m}_h$	Centre of mass of an image histogram
μ_k	k th order central moment
μ_{pq}	(p, q) th order 2-D central moment
N_B	Total number of channels (or bands) for wavelet histogram (WH) generation
N_U	Total number of channels to be downsampled for WH generation
N_G	No. of histogram gray levels
N_{GR}	No. of bins in 3-D ratio histogram
N_M	Number of moments employed for indexing
N_P	Number of pixels in an image
N_R, N_C	Number of rows/columns in an image
N_U	Total number of channels to be upsampled for WH generation
N_{WB}	Number of wavelet bands (<i>i.e.</i> corresponding parameter) used in image indexing
$v(\cdot)$	Inverse transform kernels
Ω	Forward unitary matrix for 1-D transform

$\omega(.)$	Forward transform kernels
$p(.)$	Probability density function
Π	Dead zone interval (for quantization)
Q	Query image
q	Feature vector corresponding to the query image Q
ρ	Tolerance in image retrieval
R	Bit-rate for video transmission
$r(x, y)$	Reflectance at (x, y) location
S	Size (<i>i.e.</i> number of images) of an image database.
Σ_H	Histogram space
Σ_E^K	K-dimensional Euclidean Space
σ	Standard deviation
$\Psi(x, y)$	Wavelet basis function
$\psi(y)$	1-D wavelet function
T_H	Distance threshold for histogram comparison
T_B	Threshold for AMRME technique
τ	Threshold factor for AMRME technique
t	Time
θ	Individual transform coefficients
Θ	Transform coefficient matrix
$\hat{u}, \hat{v}$	Horizontal and vertical components of motion vectors
$V_m^n(.)$	Multiresolution motion vectors corresponding to scale m and direction n
$W_{2,4,8}^{H,V,D}$	Wavelet bands corresponding to different scales and directions
z	A wavelet coefficient
$\bar{z}$	Quantized wavelet coefficient
$\hat{z}$	Reconstructed wavelet coefficient
ξ	Computational Complexity

Chapter 1

Introduction

Visual computing and communications are becoming increasingly important with the advent of broadband networks (e.g., ISDN and ATM), and low cost VLSI technology. Many new application areas, such as digital television, teleconferencing, multimedia communications [27], transmission and storage of remote sensing images [23, 114], image and video data bases, mobile multimedia, and archiving medical images are feasible with current technology. However, visual information (image, video, *etc.*) needs high bandwidth for transmission and large storage space for archival.

To meet the growing need for data compression and for ensuring compatibility, the International Standard Organization (ISO) has recently established the JPEG [120] and MPEG-1 [27, 44], and MPEG-2 [45] standards for image and video compression, respectively [27, 45, 91]. These standards are based on discrete cosine transform (DCT) of small image blocks, and are very effective in reducing the spatial redundancy in images. However, DCT-based coding has the drawbacks of blockiness and aliasing distortion at high compression ratios. In addition, it also lacks content access functionality. Hence, ISO is presently working towards efficient image and video coding standards, such as JPEG-2000, and MPEG-4. We note that JPEG-2000 [46], expected to be completed by March 2000, will provide object-oriented and progressive transmission functionalities, superior low bit-rate

performance, error resilience, and open architecture for image coding. MPEG-4 [42, 55], expected to be completed by January 1999, will employ sophisticated techniques for efficient coding and manipulation of images and video sequences.

A critical requirement for visual databases is efficient indexing of images and video clips present in the database. The main objective of researchers in this area is to develop techniques for storage and retrieval of images based on their content. These techniques should be such that the retrieved output must include all images and video frames of similar content in the database. The technique should be of low complexity to facilitate fast retrieval. Image matching and retrieval techniques are generally based on image feature vectors, such as histogram, color, texture, and shape, and are employed in the spatial domain. To meet the growing need for content-based indexing and retrieval, ISO has recently proposed to establish a standard known as *Multimedia Content Description Interface* (which is also known as 'MPEG-7') that is expected to be formalized by November 2001 [47].

The two key requirements namely, compression and indexing, thus far have been developed independently. However, a superior performance is expected when the compression parameters are also used for indexing [133]. Recently, selected techniques have been proposed for indexing in the MPEG framework [63, 97].

Wavelet theory has emerged in the last few years as a powerful technique for nonstationary signal analysis. One of the main contributions of wavelet theory is to relate the discrete time filter used in the implementation of the wavelet transform with the theory of continuous-time function space. Wavelets offer a wide variety of useful features in image and signal processing. The hierarchical structure of DWT provides the multiresolution capability for image processing. As a consequence, wavelets are popular in scalable image

and video communications. The DWT basis functions can be designed to have good time localization, which is important in visual signal processing. DWT also provides lower aliasing distortion, blocking artifacts, and mosquito noise in image/video compression applications. In addition, wavelets can be efficiently implemented in VLSI, and the computational complexity increases linearly with the data size. As a result, wavelets are being used extensively in the area of the image/video compression. Several proposals submitted for JPEG-2000 and MPEG-4 standards are based on wavelet-based techniques. However, the full potential of wavelets in the field of video compression and indexing is yet to be explored and requires detailed investigation.

1.1 Motivation and Problem Statement

1.1.1 Motivation

The multitude of visual media-based applications demands sophisticated compression and indexing techniques for efficient storage, transmission and retrieval of images and video. Wavelet transform has emerged as a powerful tool for efficient compression of images and video sequences. In addition, wavelet transform supports features such as scalability and fast random access. Hence future image and video standards such as JPEG-2000 and MPEG-4 are likely to employ wavelet transform for compression. A variety of wavelet-based image and video compression techniques has been reported in the literature. However, there needs to be significant enhancements in the motion estimation process in order to design an efficient wavelet-based video coder. More importantly, there has been little work done in the area of image and video indexing in the wavelet domain. These are crucial for consideration of wavelets as a potential candidate for multimedia applications, including standards such as

MPEG-4 and MPEG-7. Hence, there is an impending need for investigating joint compression and indexing approaches in the wavelet transform domain. This is the principal focus of this thesis.

1.1.2 Problem Statement

In this thesis we investigate wavelet-based joint compression and indexing schemes for images and video. The key areas of research include:

Motion Estimation/Video Compression

We note that motion estimation is an important step in a video compression system. However, the estimation of motion vectors is a computationally expensive task. Due to the real-time constraint (10-30 frames/sec), a motion estimation algorithm must have low complexity, and yet provide accurate motion estimates. In this thesis, we address the following issues.

1. Translation Invariance: A potential limitation restricting the use of DWT is that the DWT is not translation invariant. In this thesis, we investigate how the translation variance affects motion estimation performance, and techniques to improve the coding performance.
2. Bi-directional Motion Estimation: Most techniques proposed in the wavelet domain are based on uni-directional motion estimation. We propose techniques for bi-directional motion estimation adapted to wavelet domain.
3. Hierarchical Motion Estimation: We investigate various hierarchical motion estimation techniques, employing fine-to-coarse or coarse-to-fine approach, to obtain a superior coding performance.

4. Spatio-temporal Bit allocation: Efficient distribution of bits for encoding motion vectors (R_{Motion}) and error frames (R_{DFD}) is crucial to obtain a superior coding performance. Motion estimation (*i.e.*, temporal prediction) can be improved by reducing the block-sizes, increasing the search area, and employing sub-pixel motion estimation. However, this degrades the quality of the reconstructed error frames (assuming the bit-rate to be constant). We investigate the optimum allocation of R_{Motion} and R_{DFD} to achieve the best performance.

Indexing

The indexing techniques based on the classical pattern recognition algorithms are generally computationally expensive. To achieve near real-time performance, an indexing technique must provide a good retrieval performance, yet be computationally simple. Current indexing techniques utilize various image features such as histograms, moments, and color. These techniques in general do not exploit the characteristics of the coding scheme. One objective of this thesis is to find features in both spatial and wavelet domains for more efficient indexing of image and video. The following issues are addressed in this thesis.

1. Histogram: Although, histogram comparison is a popular technique in indexing applications, direct comparison of histograms is computationally expensive. We propose histogram-based techniques that not only improve the indexing performance but also result in a lower computational complexity.
2. Wavelet Features: The goal is to derive feature vectors from wavelet coefficients which provide a superior indexing performance.
3. Illumination Invariance: Most of the techniques proposed in the literature provide a good performance when the query and target images are acquired under similar

illumination conditions. Here, we attempt to develop indexing techniques that are robust to changes in illumination.

Joint Coding and Indexing

Here, we will investigate how to design a joint video compression and indexing system. Emphasis will be placed on extracting features that can be employed for both compression and indexing.

An overview of the thesis is shown in Fig. 1.1 illustrating the relationship among the research subtopics.

1.2 Thesis Contributions

The overall contribution of this thesis is the design of novel and efficient joint compression and indexing schemes in wavelet domain. The main contributions are the development of various techniques to achieve superior coding and indexing performance. The proposed techniques include:

Motion Estimation/Video Compression

- An adaptive-thresholding technique (section 4.3), and a bi-directional ME technique (section 4.4).
- An adaptive resolution selection technique (section 4.5.1).
- An adaptive bit-allocation technique (section 4.5.2).

Image Indexing

- An image indexing technique based on Legendre moments of image histogram (section 5.1.2).

- A fast wavelet-histogram technique for indexing (section 5.2.1.1).
- An indexing technique based on distribution of wavelet coefficients (section 5.2.2).
- An illumination invariant technique based on translation and scale-invariant moments and wavelet features (section 6.3).

Joint Coding and Indexing of Video

- A fast video segmentation technique (section 7.3).
- A joint video coding and indexing scheme (section 7.4).

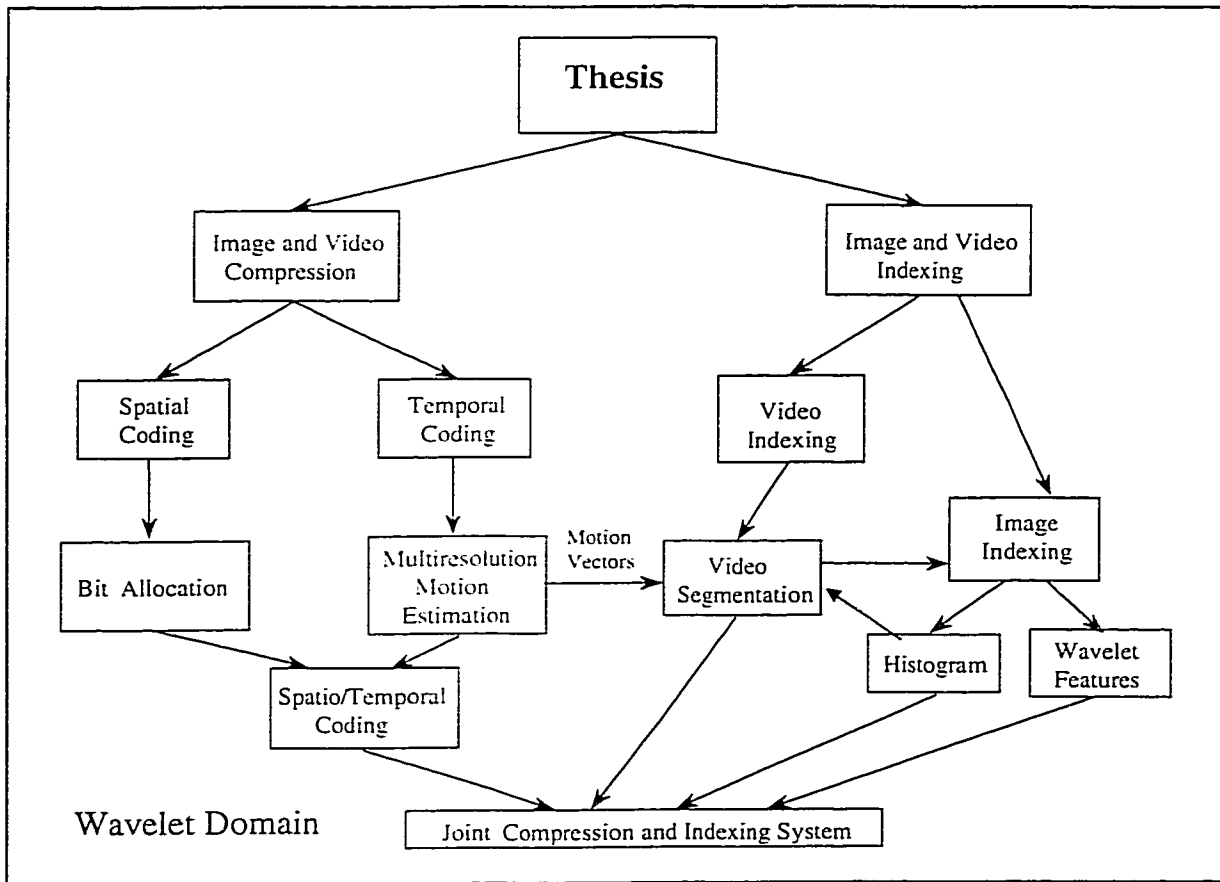


Figure 1.1. Overview of the Thesis

1.2.1 Publications

In addition to a critical review [72] of the state of the art in image and video indexing in the compressed domain, the contributions of this thesis have appeared in several refereed international journals [67, 68, 73] and conference proceedings [70, 71, 74, 75].

1.3 Outline of the Thesis

The two main issues addressed in this thesis are compression and indexing of images and video. Chapter 2 presents a review of current image and video compression techniques. Image coding techniques are presented in section 2.2. This is followed by a review of video indexing techniques in section 2.3. We then introduce wavelets in section 2.4. The chapter concludes with a summary.

In chapter 3, a review of current image indexing techniques is presented. We present image indexing techniques based on pixel domain features in section 3.1. A review of compressed domain indexing techniques is presented in section 3.2. Illumination invariant techniques are discussed in section 3.3. Review of video indexing techniques in pixel and compressed domains is presented in sections 3.4 and 3.5, respectively. Feature similarity and the evaluation criteria for indexing are presented in sections 3.6 and 3.7, respectively. The issue of joint coding and indexing techniques is addressed in section 3.8, which is followed by a summary.

Wavelet-based video compression techniques are presented in chapter 4. A typical wavelet-based video compression (encoder and decoder) scheme is outlined in section 4.1. The limitation of motion estimation in wavelet domain is discussed in section 4.2. An adaptive thresholding technique for motion estimation is presented in section 4.3. A bi-

directional motion estimation technique is then proposed in section 4.4. Two adaptive techniques are proposed in section 4.5. An adjustable resolution selection scheme is proposed in section 4.5.1, which is followed by an adaptive bit-allocation technique in section 4.5.2. The chapter concludes with a summary.

Image indexing techniques are presented in chapter 5. In section 5.1, indexing techniques employing histogram and moments are discussed. The proposed wavelet-based image indexing techniques are presented in section 5.2. Detailed performance analysis and computational complexity are presented in the respective sections. The chapter concludes with a summary.

Illumination invariant image indexing techniques are presented in chapter 6. Sections 6.1 and 6.2 detail indexing techniques based on invariant moments, and wavelets, respectively. The joint moment and wavelet-based technique and its complexity are presented in section 6.3 and section 6.4, respectively. The performance of various illumination invariant techniques are presented in section 6.5, which is followed by the chapter summary.

Joint coding and indexing of video is presented in chapter 7. Section 7.1 details the requirements of a typical video indexing system. Section 7.2 presents various features that are useful for video segmentation. The performance of these features are evaluated in section 7.2.5. A fast video segmentation technique is presented in section 7.3. Section 7.4 details the coding and indexing performance of the proposed system with a real TV sequence.

Conclusions and future research directions are presented in chapter 8. This is followed by bibliography and appendices.

Chapter 2

Review of Image and Video Compression Techniques

Image and video compression is the process of reducing the amount of data needed for its representation with an acceptable subjective quality. This is generally achieved by reducing the statistical or temporal redundancy present in an image or video. In addition, the properties of human visual system can be exploited to further increase the compression ratio. Most image and video compression techniques are based on results from information theory first formulated by Shannon [98].

In this chapter, we review the basic concepts of image and video compression techniques. A simple image model and few video data formats are discussed in section 2.1. A brief review of image compression techniques and image compression standards is presented in section 2.2. Section 2.3 presents a review of video compression techniques, associated motion estimation techniques, and video compression standards. A review of wavelet based image and video compression techniques is presented in section 2.4. We then conclude the chapter with a summary.

2.1 Digital Image/Video Signal

2.1.1 Image Model

An image can be defined [29] as a two-dimensional light intensity function $i(x, y, t)$, where the amplitude of the function at any spatial coordinate (x, y) provides the intensity (brightness) of the image at that point at a particular time t . The function $i(x, y, t)$ can be represented as a function of two components: i) the amount of source light incident on the scene and ii) the amount of light reflected by the objects in the scene. These are referred to as *illumination* component, $l(x, y, t)$ and *reflectance* component, $r(x, y, t)$, respectively. Thus, an image function can be represented as:

$$i(x, y, t) = l(x, y, t)r(x, y, t) \quad (2.1)$$

where $0 < l(x, y, t) < \infty$ and $0 < r(x, y, t) < 1$. The simple image model described above is known as the *coefficient model* [29]. We note that Eq. 2.1, where the image intensity function is directly proportional to the illumination component, holds true only for sensors with narrow band sensitivity [25].

Images can be monochrome or color. A monochrome image is generally represented in terms of the instantaneous luminance of the light field defined by $i(x, y, t)$, while a color image is represented in terms of a set of tristimulus values that are linearly proportional to the amounts of red ($R(x, y, t)$), green ($G(x, y, t)$), and blue ($B(x, y, t)$) light.

We note that an image refers to a still picture at a specific value of t , while a video consists of a sequence of images ordered in time. For digital image or video, x and y are also discrete.

2.1.2 Video Data Formats

To ensure compatibility among data from different applications, a common data format is required. CCIR (International Consultative Committee for Radio) recommendation 601 defines a digital video format for 525-line (NTSC) and 625-line (PAL) TV systems [113]. This standard is intended to facilitate international exchange of programs. The parameters of the CCIR 601 standards are tabulated in Table 2.1. We note that the raw data rate for this format is 165 Mbits/s. Since this rate is too high for most applications, CCITT (International Consultative Committee for Telephone and Telegraph) Specialist Group (SGXV) has proposed a new digital video format, called the Common Intermediate Format (CIF). The parameters of the CIF and QCIF (Quarter CIF) format are also shown in Table 2.1 [113]. We note that the CIF format is progressive and requires approximately 37 Mbit/s.

Table 2.1
Digital Video Data Formats

Parameters	CCIR-601 (4:2:2)		CIF/QCIF (4:2:0)	
	525-line/60 Hz NTSC	625-line/50 Hz PAL/SECAM	CIF	QCIF
Luminance, Y	720 × 480	720 × 576	352 × 288	176 × 144
Chrominance (U, V)	360 × 240	360 × 288	176 × 144	88 × 72
Field/Frame rate	59.94	50	29.97	29.97
Interlacing	2:1	2:1	1:1	1:1

2.2 Image Compression Techniques

2.2.1 Fundamentals of Image Compression Techniques

In general, image data is highly redundant. High compression ratio can therefore be achieved by exploiting spatial, structural and knowledge redundancies. In the following, we briefly discuss each of them.

Spatial redundancy: It refers to the correlation between neighboring pixels in an image or video frame. This intra-image or intra-frame redundancy is typically removed by employing compression techniques such as predictive coding, and transform coding.

Structural redundancy: We note that the image is originally a projection of 3-D objects onto a 2-D plane. Therefore, if the image is encoded using structural image models that take into account the 3-D properties of the scene, a high compression ratio can be achieved. For example, a segmentation coding approach that considers an image as an assembly of many regions and encodes the contour and texture of each region can efficiently exploit the *structural* redundancy in an image/video sequence.

Knowledge redundancy: When the object to be coded is limited in its scope, a common knowledge can be associated with it at both the encoder and decoder. The encoder can then transmit only the necessary information (*i.e.* the change) required for reconstruction. For example, in the case of a videophone application, the image sequence to be coded is usually limited to parts of a human body, namely the head and shoulder.

In addition to redundancy reduction, the human visual system (HVS) properties can also be exploited to improve the subjective quality of image/video signal [3, 20, 51]. Some of the HVS properties that are useful in image/video compression are:

- Greater sensitivity to distortion in dark areas in images.
- Greater sensitivity to distortion in smooth areas compared to areas with sharp changes (*i.e.*, areas with higher spatial frequencies).
- Lower sensitivity to faster moving objects in a scene.
- Greater sensitivity to signal changes in the luminance component compared to the chrominance component in color images.

2.2.2 Popular Image Coding Techniques

There are two kinds of image compression techniques: i) lossless techniques, ii) lossy techniques. In lossless compression techniques, the statistical redundancy is exploited in such a way that the entire process is reversible *i.e.* the original image can be fully recovered. However, it results in a low compression ratio (typically 2 to 3). Lossy compression techniques, on the other hand, achieve a high compression ratio, but with some loss of information. These techniques are generally based on predictive coding, transform coding, vector quantization, *etc.* In the following, we briefly discuss these coding techniques.

Predictive Coding

Predictive coding exploits the redundancy related to the predictability and smoothness in an image [52]. For example, an image having a constant gray level can be fully predicted from the gray level value of its first pixel. In images with multiple gray levels, the gray level of an image pixel can be predicted with high accuracy from the values of its neighboring pixels. Prediction error is then encoded instead of the original pixels, and a high compression ratio is thus achieved. *Differential pulse code modulation* is the basic compression scheme used in predictive coding techniques.

Vector Quantization

A fundamental result of Shannon's rate-distortion theory is that better performance can always be achieved by coding vectors instead of scalars. Let the K -dimensional Euclidean space be denoted by Σ_E^K . A vector quantizer (VQ) can then be defined [60] as a mapping Q of Σ_E^K into a finite subset Y of Σ_E^K , where Y is the set of reproduction vectors, and is called a VQ codebook. At the transmitter and receiver, an identical codebook exists whose entries

contain combinations of pixels in a block. At the encoder, each data vector is matched with or approximated by a codeword in the codebook, and the address or index of that codeword is transmitted instead of the data vector itself. At the decoder, the index is mapped back to the codeword, and the codeword is used to represent the original data vector.

The major drawback of vector quantization is that it is highly image-dependent and its computational complexity grows exponentially with the vector dimension. In addition, it is difficult to design a good codebook that is representative of all the possible occurrences of pixel combinations in a block.

Transform Coding

In transform coding, the image data is first transformed from spatial to an alternate domain by a unitary transform and then encoded by scalar or vector quantization techniques. A unitary transform is a reversible linear transform whose kernel describes a set of complete orthonormal basis functions. The objective of transform coding is to decorrelate the original signal and to repack the energy into fewer coefficients.

Let an $N_R \times N_C$ image be denoted by

$$I = [i(m, n)] \quad 0 \leq m \leq N_R - 1, 0 \leq n \leq N_C - 1$$

The forward and inverse transforms are defined as

$$\theta(k, l) = \sum_{m=0}^{N_R-1} \sum_{n=0}^{N_C-1} \omega(k, l; m, n) i(m, n) \quad 0 \leq k \leq N_R - 1, 0 \leq l \leq N_C - 1 \quad (2.2)$$

$$\hat{i}(m, n) = \sum_{k=0}^{N_R-1} \sum_{l=0}^{N_C-1} v(m, n; k, l) \theta(k, l) \quad 0 \leq m \leq N_R - 1, 0 \leq n \leq N_C - 1 \quad (2.3)$$

where $\omega(\cdot)$ and $v(\cdot)$ are the forward and inverse transform kernels.

In most practical cases, the 2-D kernels are separable and symmetric so that the 2-D kernel can be expressed as the product of the two 1-D orthogonal basis functions. Hence, the image transformation can be done in two stages: i) by taking the unitary transform (1-D) of each row of the image array and then ii) taking the transform of each column of the intermediate result. A typical transform coding scheme is shown in Fig. 2.1.

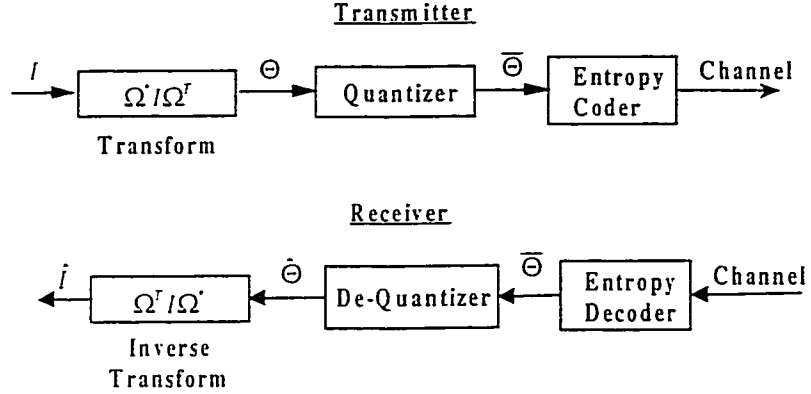


Figure 2.1. Block diagram of a transform coding scheme, I : Original image, $\hat{I}$: Reconstructed image, Ω : Forward unitary matrix for 1-D transform, Θ : Transform coefficient matrix, $\bar{\Theta}$: Quantized transform coefficient matrix, $\hat{\Theta}$: Reconstructed transform coefficient matrix.

The transformation of an image results in a set of coefficients that are generally nonstationary in nature. The transform will be statistically *optimal* for image coding if it satisfies the following two criteria: (i) there should not be any correlation among the coefficients *i.e.*, the autocorrelation matrix should be diagonal, and (ii) it should pack the energy in as few coefficients as possible. The unitary transform that satisfies both criteria is the Karhunen-Loeve transform (KLT) [49]. However, KLT is image dependent and has a higher computational complexity. Therefore, image independent sub-optimal transforms such as discrete cosine, Fourier, and Hadamard transforms are used in practice. Among all the sub-optimal transforms, the rate-distortion performance of the discrete cosine transform

(DCT) is closest to KLT [49]. Hence, DCT has been adopted as the compression technique in image and video coding standards, such as JPEG, MPEG, H.261, and H.263.

Orthonormal transforms have two properties that are very useful in image coding applications [49]. Firstly, they satisfy Parseval's relation *i.e.* the total energy in the frequency domain is equal to that in the spatial domain. Secondly, the mean square reconstruction error (in spatial domain) is equal to the mean square quantization error (in frequency domain). These two properties are very helpful in designing a *minimum square error* (MSE) quantizer.

2.2.3 Image Compression Standards

To ensure compatibility among various image compression schemes, several image compression standards have been proposed by the International Standard Organization (ISO) and the International Telecommunications Union (ITU). ITU (CCITT) G3/G4 standards have been developed for fax image transmission, and are being used in all fax machines. It has been recognized that the ITU G3/G4 codes cannot effectively deal with some images, especially digital halftones. To address this problem, JBIG [33] standard has been proposed by a joint committee of ISO and ITU. The JBIG standard employs adaptive arithmetic coding technique and outperforms ITU-G4 by as much as 30%. This standard also accommodates progressive transmission.

In order to encode continuous tone (gray scale or color) images, CCITT and ISO collaborated to develop the most popular and comprehensive JPEG (Joint Photographic Experts Group) standard. JPEG standard provides a framework for high quality compression and reconstruction of continuous-tone gray scale and color images for a wide range of applications [120]. The standard specifies details of the compression and decompression algorithms for various application environments. It has four modes of operation that

generally cover most image compression environments. These are - i) baseline sequential, ii) progressive coding, iii) hierarchical coding, and iv) lossless coding [120]. The baseline sequential mode provides a simple and efficient algorithm that is adequate for most image coding applications. The other modes are employed for more sophisticated applications.

Presently, the ISO committee is pursuing a new coding system which is known as JPEG 2000. This incoming standard will significantly improve the coding performance compared to the present JPEG standard. In addition, it will provide new functionalities, such as object manipulation, both lossless and lossy compression methods, error resilience, and open architecture. This standard is expected to be operational by 2000 A.D.

2.3 Video Compression Techniques

Considering video as a sequence of image frames, the image coding techniques can be applied to video frames individually to achieve compression. This is done in Motion JPEG (we note that this not an international standard), where each frame is coded using the JPEG standard algorithm. However, this technique is not efficient since it does not exploit the temporal redundancy among the neighboring frames. We note that the differences between successive frames are usually very small and are essentially due to object, or camera motion. To exploit the temporal redundancy, several techniques such as conditional replenishment, adaptive predictive coding and predictive coding with motion compensation can be used. Among these approaches, predictive coding with motion compensation is generally used for low bit-rate video coding applications. Here, a video frame is predicted from a previous reference frame. The magnitude of the prediction error is reduced by a technique known as motion estimation. In this technique, the objects in the current frame are first displaced to

their estimated positions in the previous frames, and then the subtraction of two frames is performed. This produces a *difference* frame with much less information compared to simple inter frame difference. The only information required to be transmitted in the case of motion compensated frame difference is just the values of the motion vectors for each object while a substantial amount of information is needed if a simple frame difference is transmitted. It has been shown that coding of motion compensated frames generally result in a 25-35% lower bit-rate compared to coding simple frame differences [50] despite the overhead of the motion vectors. In the following sections, a brief review of various motion estimation techniques and video compression standards is provided.

2.3.1 Conventional Motion Estimation Techniques

Motion estimation techniques are generally based on block matching [10, 50, 78, 104]. Here, an image is divided into a number of small blocks on the assumption that the pixels within a block belong to a rigid body, and thus have the same motion activity. In the following, we will briefly discuss a few block-based methods.

We start with a video signal $i(x, y; k)$, where (x, y) denotes the spatial coordinate, and k denotes the time. The goal is to find a mapping $d(x, y; k)$ that would help reconstruct $i(x, y; k)$ from $i(x, y; k \pm p)$, where p is a small integer. We assume a restrictive motion model where an image is assumed to be composed of rigid objects in translational motion on a plane.

$$i(x, y; k) = i((x, y) - d(x, y; k), k - p) \quad (2.4)$$

We also expect homogeneity in time, *i.e.*,

$$i(x, y; k) = i((x, y) + d(x, y; k), k + p) \quad (2.5)$$

In a block-based scheme, these assumptions are expected to be valid for all points within a block b using the same displacement vector d_b . These assumptions are easily justified when the blocks are much smaller than the objects, and temporal sampling is sufficiently dense.

In a block matching motion estimation scheme, each frame is divided into non-overlapping rectangular blocks of size $K \times L$. Each block in the present frame is then matched to a particular block in the previous frame(s) to find the horizontal and vertical displacements of that block. This is illustrated in Fig. 2.2, where the maximum allowed displacement in the vertical and horizontal directions are, respectively Δ_u and Δ_v . The most frequently used block matching criteria are *the mean absolute difference* (MAD) and *mean squared error* (MSE). The optimum motion vector (having two components $\hat{u}$ and $\hat{v}$) can be expressed using the MAD and MSE criteria respectively as:

$$(\hat{u}, \hat{v}) = \arg \min_{\substack{(u,v) \in \mathbb{Z}^2 \\ |u| \leq \Delta_u, |v| \leq \Delta_v}} \sum_{x=0}^{K-1} \sum_{y=0}^{L-1} |i_{DFD}(x, y; u, v)| \quad (2.6)$$

$$(\hat{u}, \hat{v}) = \arg \min_{\substack{(u,v) \in \mathbb{Z}^2 \\ |u| \leq \Delta_u, |v| \leq \Delta_v}} \sum_{x=0}^{K-1} \sum_{y=0}^{L-1} (i_{DFD}(x, y; u, v))^2 \quad (2.7)$$

where,

$$i_{DFD}(x, y; u, v) = i(x, y; k) - i(x - u, y - v; k - 1) \text{ and} \quad (2.8)$$

$\mathbb{Z}$ is the set of all integer numbers

In Eqs. 2.6 and 2.7, $(u, v) \in \mathbb{Z}^2$ signifies that the motion vectors have one-pixel accuracy. More accurate ME is possible by estimating motion vectors at a fraction-pixel accuracy. We note that the computational complexity of full search algorithm (FSA) is very high. The total number of possible displacement values, with one-pixel accuracy, is $(2\Delta_u + 1)(2\Delta_v + 1)$.

Hence, The complexity of the technique is on the order of $4\Delta_u\Delta_v$ operations/pixel. Several techniques have been proposed to reduce the complexity of motion estimation algorithms. Most of these techniques are based on the assumption that the matching criterion (*i.e.*, the error) increases monotonically as the search moves away from the direction of minimum distortion. These algorithms are faster compared to FSA, however, they may converge to a local optimum, which corresponds to an inaccurate prediction of the motion vectors. We now present a few selected block-matching techniques.

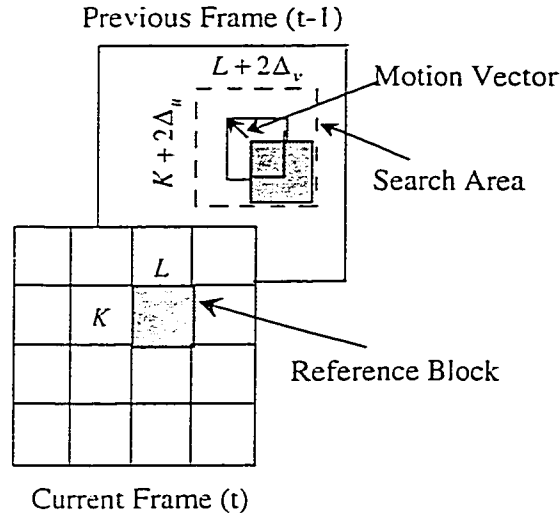


Figure 2.2. Block matching motion estimation process

2-D logarithmic search

The 2-D logarithmic search introduced by Jain and Jain [50] is an extension of the logarithmic search in one dimension. In each step of this algorithm, search is performed only at the five locations that include a middle point and four points in the two main directions, horizontal and vertical. The location that provides the minimum DFD is considered the center of the five locations used in the next step. If the optimum is at the center of the five locations, the search area is decreased by half, otherwise the search area remains identical to

that of the previous step. The procedure continues in recursive manner until the distance between the search points is reduced to 3×3 . In the final step, all the nine locations are searched and the position of the minimum distortion is selected as the x and y components of the motion vectors. This algorithm reduces the number of calculations from $(2p+1)^2$ required by the full search (when $\Delta_u = \Delta_v = p$) to only $2 + 7 \log_2 p$.

Increasing Accuracy Search

This algorithm is based on the convexity hypothesis criterion [56]. In the first step, the displacement vectors are tested on nine points coarsely spaced around the center of the search area. After determining the displacement vector that minimizes the DFD, the next level of search is pursued with increasing accuracy, until a single-pixel accuracy is obtained. This algorithm reduces the number of calculations from $(2p+1)^2$ required by the full search (when $\Delta_u = \Delta_v = p$) to only $1 + 8 \log_2 p$.

Conjugate Direction Search

This algorithm was proposed by Srinivasan and Rao [104]. Starting at the center of the block, the vertical direction is kept fixed while the horizontal direction is varied to find the point of minimum distortion. From this minimum location, the horizontal direction is kept constant while the vertical is varied to find the minimum in the vertical direction. The maximum number of searches using this technique is $(2p+3)$.

2.3.1.1 Evaluation Criteria for Motion Estimation Techniques

A motion estimation algorithm is evaluated using two factors - i) the motion compensation efficiency, and ii) the computational complexity. The motion compensation efficiency can be

measured in terms of the prediction gain. The prediction gain for an image block can be defined as:

$$G_{MP} = \frac{\text{Energy of the original image block}}{\text{Motion compensated residual energy}} \quad (2.9)$$

When the motion compensation is adequate, the residual energy will be small resulting in a high prediction gain G_{MP} .

To facilitate real time implementation, the computational complexity of a motion estimation algorithm should be low. The computational complexity is proportional to the number of points tested by an algorithm for a given search area. However, for real-time hardware implementation, the number of required sequential steps can also be important since some of these can be evaluated in parallel.

2.3.2 Video Compression Standards

Several video compression standards have been developed recently for different applications. The CCITT has recommended a standard, called H.261, for video telephony over ISDN lines at $p \times 64$ Kbits/s [77]. Typically, video conferencing using CIF format requires 384 Kbits/s, which corresponds to $p = 6$. This standard has been specifically developed for teleconferencing applications where movements of the subjects (*i.e.*, the persons) are very small. Recently, the International Standards Organization (ISO) has proposed the MPEG-1 and MPEG-2 standards for video and audio compression [91]. MPEG-1 [44] specifies a coded representation that can be used for compressing video sequences up to bit rates of 1.5 Mbit/s. It was developed in response to the growing need for a common format for representing compressed video on various digital storage media such as

CD's, DAT's, Winchester disks and optical drives. MPEG-2 has been developed for a target rate of up to 50 Mbits/sec, intended for applications requiring high quality digital video and audio. MPEG-2 video builds upon the completed MPEG-1 standard by supporting interlaced video formats and a number of advanced features including those supporting HDTV. A brief review of MPEG-1 standard is presented now.

MPEG-1

In this standard, a block-based motion compensation is employed to remove the temporal redundancy. Motion compensation is used for both causal prediction of the current picture from a previous picture and for non-causal interpolative prediction from past and future pictures. The residual spatial correlation in the predicted error frames is further reduced by employing block DCT similar to JPEG. Fig. 2.3 shows the schematic of the MPEG encoder.

Because of the conflicting requirements of random access and high compression ratio, the MPEG standard suggests that frames be divided into three categories: I, P and B frames. Intra coded frames (I-frames) are coded without reference to other frames. They provide access points to the coded sequence where decoding can be performed immediately, but are coded with moderate compression ratio. Predictive coded frames (P-frames) are coded more efficiently using motion compensated prediction from a past intra (I-frame), or another P-frame, and are generally used as a reference for further prediction. Bi-directionally predictive coded frames (B-frames) provide the highest degree of compression but require both past and future reference frames for motion compensation. B-frames are never used as references for prediction. The organization of the three frame types in a sequence is very flexible. The choice is left to the encoder and will depend on the requirements of the application. Fig. 2.4

illustrates the relationship among the three different frame types in a group of pictures (GOP).

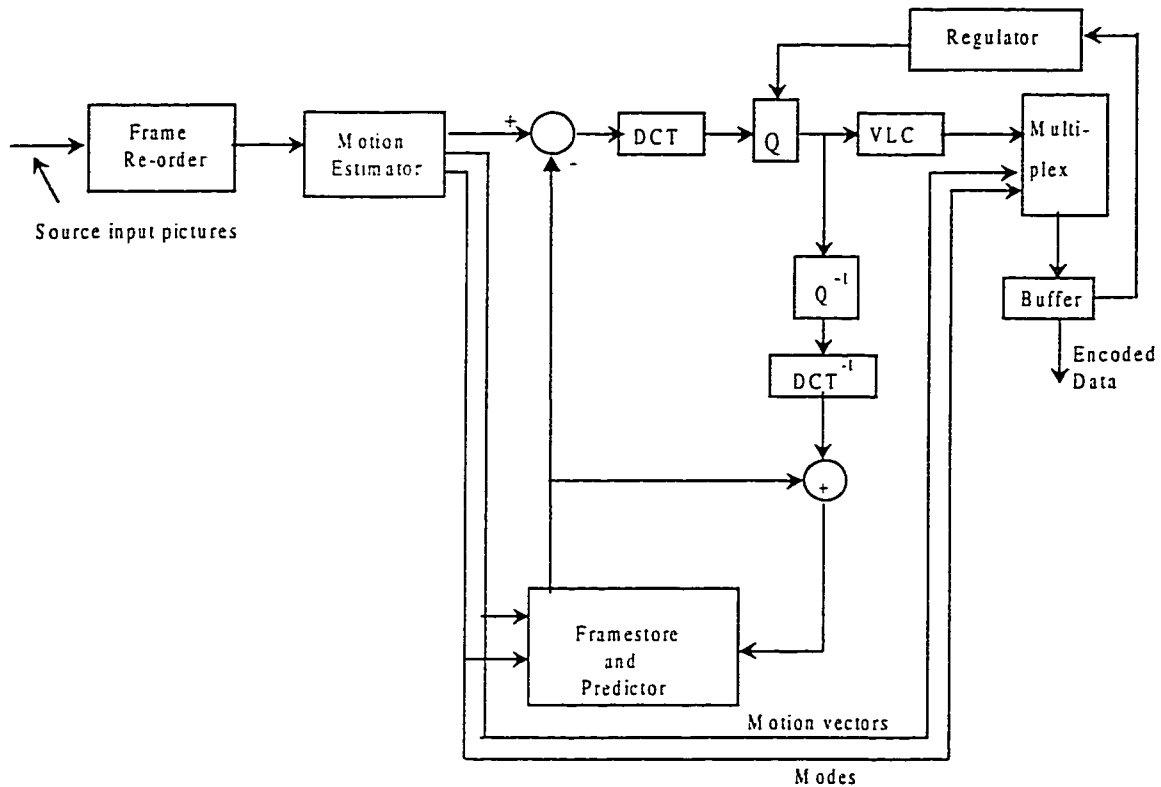


Figure 2.3. Block diagram of MPEG video encoder [27]

MPEG-4

We note that MPEG-1/MPEG-2 standards employ interframe motion compensation and DCT to achieve compression. It is known that DCT-based coding has the drawbacks of blockiness and aliasing distortion at high compression ratios. In addition, MPEG-1/2 standards do not provide content access functionality. Hence, the ISO is presently working towards a new video and audio (both synthetic and natural) coding standard known as MPEG-4 [42, 55]. This incoming standard will provide techniques for storage, transmission and manipulation of textures, images and video data in multimedia environments over wide

range of bit-rates. Here, a scene will be segmented into background and foreground which in turn is represented by video objects. The object-based representation is expected to significantly improve the coding performance. The standard is expected to be operational by January 1999.

2.4 Wavelets in Image and Video Compression

2.4.1 Theory of Wavelets/Subbands

Subband coding was first developed for speech compression and later extended for image coding [125, 126]. In subband coding, an image is first filtered to create a set of subimages or *subbands*, each of which contains a limited range of spatial frequencies. Different subbands can be downsampled, due to their lower bandwidths as compared to the original image, keeping the data rate unchanged. The subbands are then quantized and encoded using one or more coders. Different bit-rates or coding techniques can be used for each subband, thus taking advantage of the properties of the subbands. The coding errors can be distributed over the subbands in a visually optimal manner. The image is reconstructed by upsampling the decoded subbands, applying appropriate filters and adding the reconstructed subbands together. Subband coding is generally implemented using quadrature mirror filters (QMF's) to reduce the aliasing effects in the reconstructed image.

Subband coding is motivated by the idea that the subbands can be coded more efficiently than the full-band image. This is because most of the energy in the subband domain is represented by a few lowpass coefficients. The idea is very similar to that of transform coding. In fact, transform coding and subband coding are two special cases of multirate filterbanks. In practice, DCT, DFT, *etc.*, are used in a block coding approach with a

blocksize of 8×8 or 16×16 . This can be viewed as a filterbank with the decimation factor being the same as the filter length. In subband coding, the subband filter length is generally much larger than the decimation factor, resulting in fewer blocking artifacts in the reconstructed image [4].

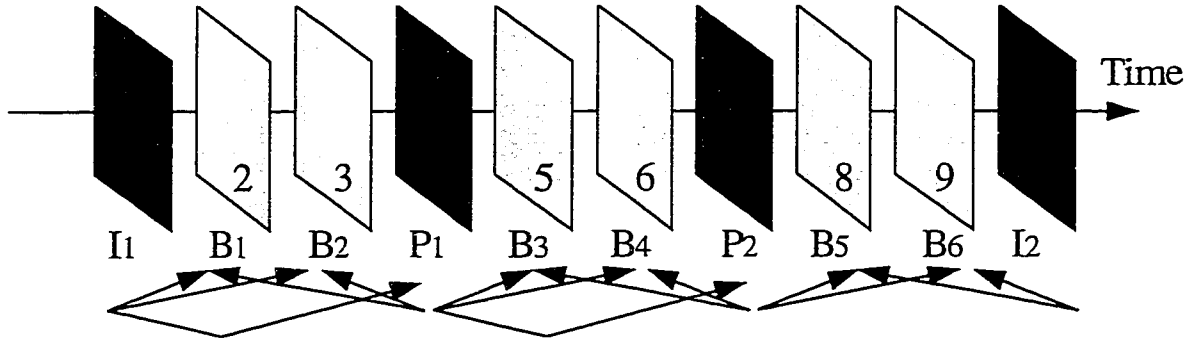


Figure 2.4. Example of a group of pictures (GOP) used in MPEG

Wavelet transform is a special case of subband decomposition, which is defined as follows. Let $L^2(R)$ denote the vector space of measurable, square integrable one-dimensional function. The continuous wavelet transform (CWT) of a function $f(t) \in L^2(R)$ is defined as:

$$F(a,b) = \int_{-\infty}^{\infty} f(t) \Psi_{a,b}^*(t) dt \quad (2.10)$$

where the wavelet basis functions $\Psi_{a,b}(t) \in L^2(R)$ can be expressed as

$$\Psi_{a,b}(t) = a^{-1/2} \Psi\left(\frac{t-b}{a}\right), \quad a \in R^+, b \in R \quad (2.11)$$

These basis functions are called *wavelets* and have at least one vanishing moment. The arguments a and b denote the *scale* and *location* parameters, respectively. The oscillation in the basis functions increases with a decrease in a . The factor $a^{-1/2}$ on the right hand side of

Eq. 2.11 maintains the norm of the wavelet function in different scales. The wavelet transform defined in Eq. 2.10 is highly redundant since a function of one variable t is represented as a function of two variables a and b . The redundancy can be removed by discretizing a and b . When $a = 2^{-k}$ (k is a nonnegative integer) and $b \in \mathbb{Z}$, the transformation is known as dyadic wavelet transform.

It is observed in Eq. 2.11 that the basis functions are dilated and translated versions of the mother wavelet $\Psi(t)$. As a consequence of this, it was shown in [64] that the wavelet coefficients of any scale (or resolution) could be computed from the wavelet coefficients of next higher resolutions. This has facilitated implementation of wavelet transform using a recursive approach that is known as *Mallat's tree algorithm*. Here, wavelet transform of a 1-D signal is calculated by passing it through a lowpass filter (LPF) and a highpass filter (HPF), and by decimating the filters' outputs by a factor of two. This is shown in Fig. 2.5a. Mathematically, this can be expressed as:

$$c_{j+1,k} = \sum_m c_{j,m} h[m - 2k] \quad (2.12)$$

$$d_{j+1,k} = \sum_m c_{j,m} g[m - 2k] \quad (2.13)$$

where

$c_{p,q}$ = Lowpass (or, scaling) coefficient of p th scale at q th location.

$d_{p,q}$ = Highpass (or, wavelet) coefficient of p th scale at q th location.

$h[.]$ = Lowpass filter coefficients corresponding to the mother wavelet

$g[.]$ = Highpass filter coefficients corresponding to the mother wavelet

Similar to the wavelet analysis, the reconstruction of the original fine scale coefficients can be done from the coarser coefficients (see Fig. 2.5b). This can be expressed as follows:

$$c_{j,m} = \sum_k c_{j+1,k} h[m-2k] + \sum_l d_{j+1,l} g[m-2l] \quad (2.14)$$

The schematic of one-stage wavelet decomposition and reconstruction of 1-D signal is shown in Fig. 2.6.

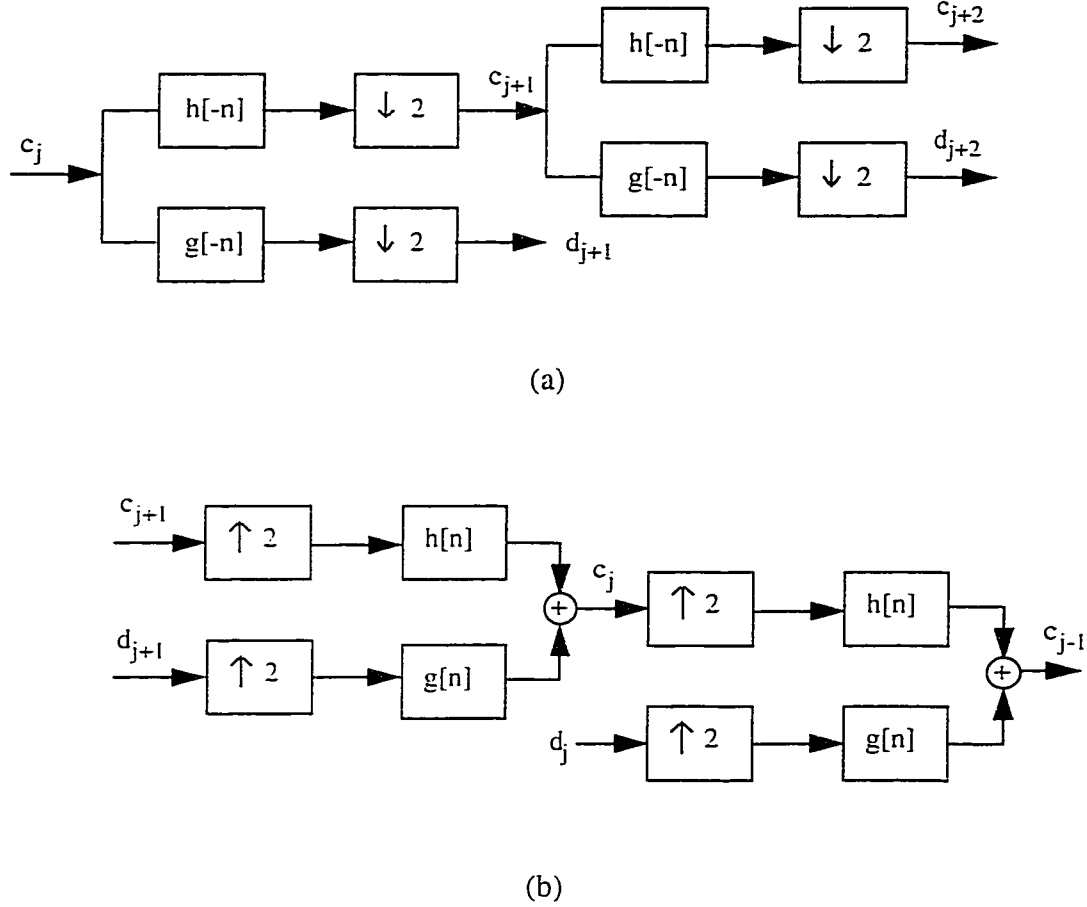


Figure 2.5. Schematic of Mallat's tree algorithm. a) Signal decomposition using analysis filters. b) Signal reconstruction using synthesis filters.

Two-Dimensional Wavelet Transform

A wavelet basis of $L^2(R^2)$ can be constructed using 2-D multiresolution analysis scheme. However, the construction of such wavelets is difficult. A simpler approach [19, 64] has been proposed to construct a 2-D separable orthonormal basis by taking tensor product of two

1-D orthonormal wavelet bases. In this approach, three 2-D wavelet basis functions are defined from their 1-D counterpart, as follows:

$$\Psi^h(x, y) = \phi(x)\psi(y) \quad (2.15a)$$

$$\Psi^v(x, y) = \psi(x)\phi(y) \quad (2.15b)$$

$$\Psi^d(x, y) = \psi(x)\psi(y) \quad (2.15c)$$

where h, v, d stand for *horizontal, vertical and diagonal*, respectively.

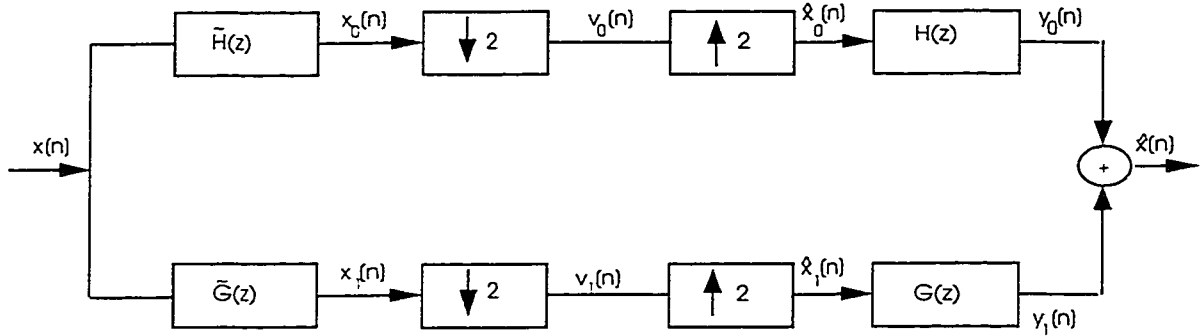


Figure 2.6. Schematic of 1-D wavelet decomposition and reconstruction

In 1-D case, we have seen that each level of decomposition produces two bands corresponding to low and high-resolution data. In case of 2-D wavelet transform, each level of decomposition will produce four bands of data, one corresponding to scaling functions and three corresponding to *horizontal, vertical and diagonal* wavelets. If the original 1-D $\phi(x)$ and $\psi(x)$ have compact support, then the corresponding 2-D scaling and wavelet functions will also have compact support. The filtering can be done on "rows" and "columns" in the two-dimensional array (as shown in Fig. 2.7), corresponding to horizontal and vertical directions in images.

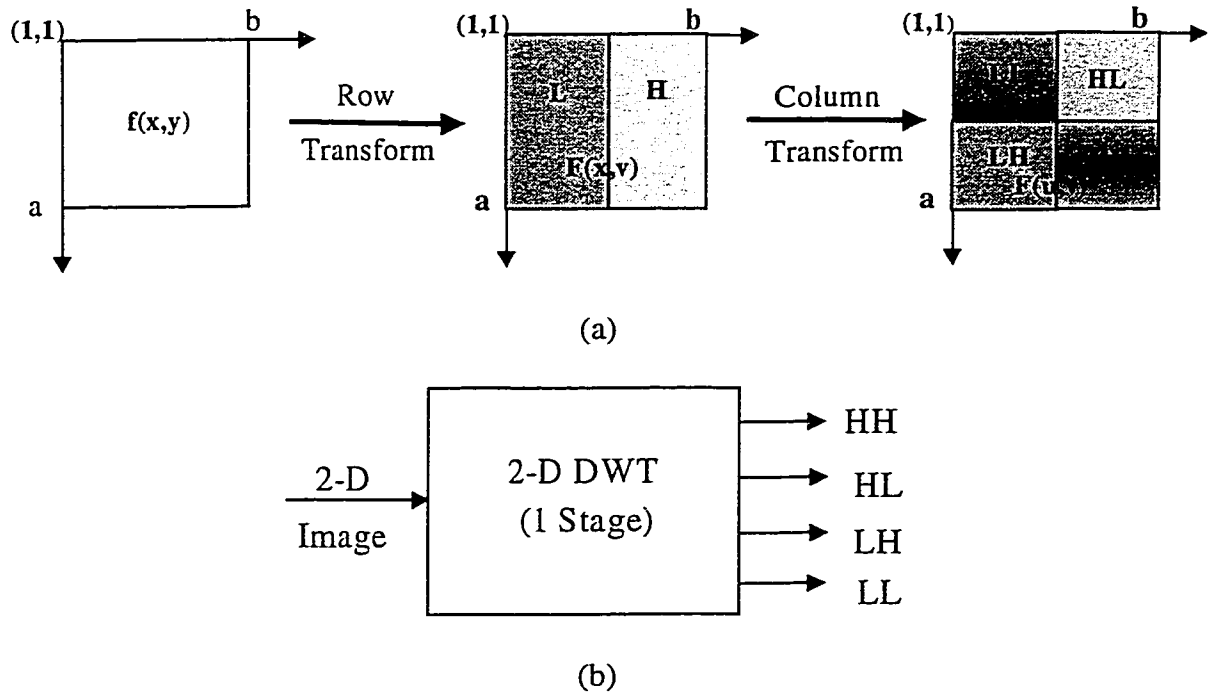


Figure 2.7. 2-D wavelet transform. a) 2-D DWT by 1-D row and column transforms, b) Equivalent block schematic. L: output of lowpass filter after decimation, H: output of highpass filter after decimation.

Fig. 2.8 shows a 3-level wavelet decomposition of an image S_1 of size $a \times b$ pixels. In the first level of decomposition, one low pass subimage (S_2) and three orientation selective highpass subimages (W_2^H, W_2^V, W_2^D) are created. In second level of decomposition, the lowpass subimage is further decomposed into one lowpass and three highpass subimages (W_4^H, W_4^V, W_4^D). This process is repeated on the low pass subimage to form higher-level wavelet decomposition. In other words, DWT decomposes an image into a pyramid structure of subimages (see Fig. 2.9) with various resolutions corresponding to the different scales. The inverse wavelet transform is calculated in the reverse manner, *i.e.* starting from the lowest resolution subimages, the higher resolution images are calculated recursively.

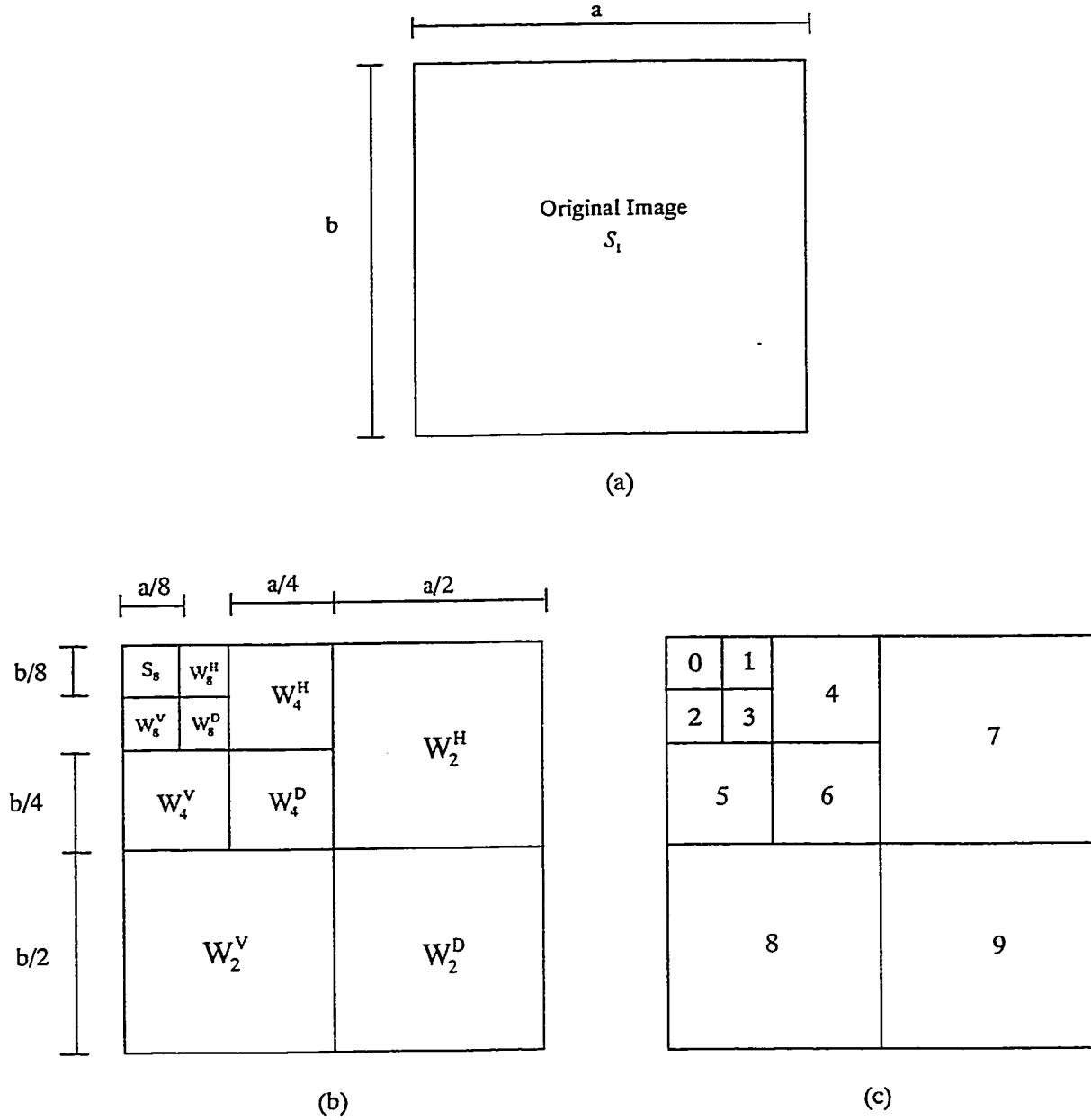


Figure 2.8. Three-stage wavelet transformed image. a) original image, b) Various directional bands, c) Each band has been associated with a number for easy identification.

2.4.2 Wavelet Coding of Images

Recently, wavelets have become very popular in image processing, specifically in coding applications [6, 20, 61, 65, 130] because of several reasons. Firstly, wavelets are efficient in

representing nonstationary signals because of the adaptive time-frequency window. Secondly, they have high decorrelation and energy compaction efficiency. Thirdly, blocking artifacts and mosquito noise are reduced in wavelet-based video coder. Finally, the wavelet basis functions match the human visual system characteristics, resulting in a superior image representation.

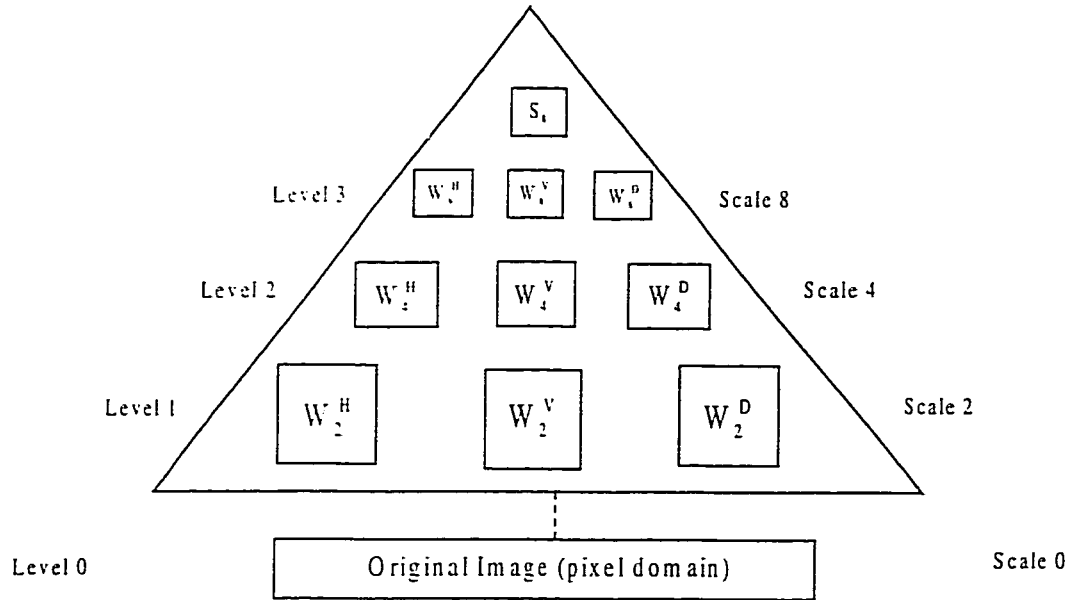


Figure 2.9. Wavelet pyramid of 4 levels.

In addition to the above advantages, wavelets have a direct relation to multiresolution (MR) analysis. MR analysis represents image and video in a scale-space framework where coarse features are large-scale objects and fine-scale features are studied much more locally. It has been shown that wavelets with reasonable time-frequency localization necessarily stem from multiresolution analysis. The multiresolution scheme successfully addresses the following:

1. Signal decomposition for coding
2. Scalable image and video compression

3. Representation well suited for fast random access in digital storage devices
4. Robustness and error recovery.
5. Suitable signal representation for joint source/channel coding.
6. Compatibility with lower resolution representations.

Coding Scheme

Wavelet-based coding techniques can be classified into two categories - i) scalar quantization [130] and ii) vector quantization [6]. Both approaches have their own advantages and disadvantages. It is known that the high frequency coefficients can be modeled fairly accurately with *generalized Gaussian distribution* [12]. Scalar quantizers exploit this while designing their quantization table. On the other hand, it is known that sharp edges are characterized by frequency components of all resolutions. Hence, there will be some residual correlation among coefficients of different scales. Vector quantizers exploit the correlation among coefficients of different scales resulting in a superior coding performance.

Several wavelet-based image compression techniques, with high compression efficiency, have been proposed in the recent literature [95, 99, 128]. However, in this thesis we will employ a simple, but reasonably efficient coding technique [65] that is shown in Fig. 2.10. The main steps of the coding scheme are forward transform, bit allocation, scanning and arithmetic coding.

Transform

The wavelet transform in Fig. 2.10 is basically a 2-D orthogonal or bi-orthogonal transform. The decomposition may be dyadic (only the lowest scale is decomposed recursively), regular (full decomposition) or irregular in nature. The depth of tree is generally

determined by the size of the image and the number of wavelet filter taps. With each decomposition, the number of rows and columns of the lowest passband is halved. For efficient decomposition, the number of rows and columns of the band to be decomposed should not be less than the number of filter taps. In practice, the depth of the tree ranges from 3 to 5.

The computational complexity of DWT is of $O(N)$ where N is the number of data samples. There are several algorithms for computing DWT. However, polyphase decomposition is mostly used because of its simplicity. If the 2-D wavelet transform is implemented using a separable approach, the complexity of decomposing an $N_R \times N_C$ image for J stages will be [65]

$$\xi_{dyadic} = \frac{16N_R N_C L}{3} (1 - 4^{-J}) \text{ FLOP} \quad (2.16)$$

$$\xi_{regular} = 4JLN_R N_C \text{ FLOP} \quad (2.17)$$

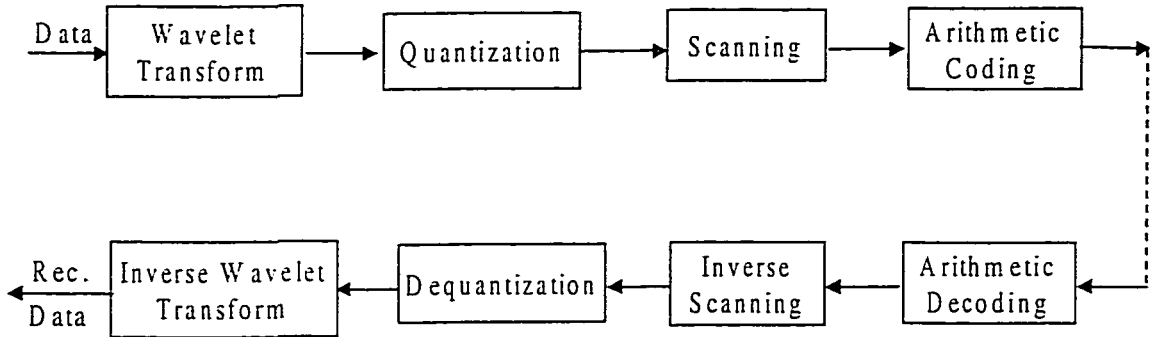


Figure 2.10. A wavelet-based image coding scheme

The complexity shown in Eq. 2.16-2.17 can be reduced by employing more sophisticated filtering techniques. For bi-orthogonal wavelets, the complexity can be further reduced by exploiting the symmetry of the wavelet filters.

Bit Allocation

In order to obtain significant compression, the bit allocation procedure in a subband coding technique generally requires optimization. The most popular criteria for optimization is to minimize the mean square error. In other words, the transform coefficients should be assigned bits depending on their contribution to error variance in the spatial domain. If the signal has $N \times N$ coefficients and the total number of available bits is R , then the optimization problem is to find various $R_{i,j}$, so that

$$D = \sum_{i,j=0}^{N-1} D_{i,j} \text{ is minimized with the constraint}$$

$$R = \sum_{i,j=0}^{N-1} R_{i,j} \quad (2.18)$$

where $D_{i,j}$ is the distortion produced by the (i,j) -th coefficient when $R_{i,j}$ bits are assigned to it. If we use the *pdf* optimized Lloyd-Max quantizer, the total distortion becomes [4, 52]:

$$D = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \gamma_{i,j} 2^{-2R_{i,j}} \sigma_{i,j}^2 \quad (2.19)$$

where $\sigma_{i,j}^2$ is the variance of the (i,j) -th transform coefficient, and $\gamma_{i,j}$ is a performance factor that depends on the *pdf* of the coefficient.

For orthonormal transforms, the above problem can be solved using *Lagrangian optimization* [4]. The solution becomes:

$$R_{i,j} = R_{avg} + \frac{1}{2} \log_2 \frac{\sigma_{i,j}^2}{\left[\prod_{k,l=0}^{N-1} \sigma_{k,l}^2 \right]^{1/(N \times N)}}; \quad i,j = 0,1,\dots,N-1 \quad (2.20)$$

where, R_{avg} is the average number of bits per coefficient, i.e., $R_{avg} = R / (N \times N)$.

Scanning and Arithmetic Coding

After each band is quantized with its corresponding quantization step size, the DWT coefficients are encoded using arithmetic coding. Since, the statistics of different bands vary widely, the bands are generally encoded independently. The remaining nonstationarity within a band is easily handled by an adaptive model. Since adaptive coding is a memory process, the order in which the coefficients are fed into the coder is an important issue. The higher the local stationarity of coefficients the better is the adaptation. We note that various types of scanning, *e.g.* horizontal, zigzag, and Peano-Hilbert scanning [93], are employed in practice.

2.4.3 Wavelet-based Video Coding

Several wavelet-based video coding schemes have been proposed in the literature [8, 9, 13, 28, 34, 54, 82, 117, 135]. A wavelet-based video coding scheme similar to that of MPEG standard can be developed by using wavelets (see Fig. 2.3) instead of DCT. The video sequence is first decorrelated in the temporal direction using motion compensation. Wavelet transform is then applied on the motion compensated error frames to further reduce the remaining spatial correlation. A scalar or vector quantization technique may then be applied to encode the DWT coefficients.

There is an alternative to the above scheme, which is shown in Fig. 2.11. Here, the video frames are first decorrelated spatially using wavelet transform (for 3-5 stages). The temporal redundancy is then removed by compensating motion in various subbands. Gharavi [28] has implemented both schemes and has reported that the motion compensation in the second scheme (DWT followed by motion compensation) does not perform well compared to the first scheme (motion compensation followed by DWT).

Although, motion estimation does not perform well in the wavelet-decomposed video, the second scheme is more flexible in accommodating the coding at multiple resolutions within one coding structure [13]. Once, the frames are decomposed into subbands, the receiver can choose which subimages to use for reconstruction. Hence, the same image can be reconstructed at reduced or full resolution using a subset or all the subimages produced during the coding phase. However, if wavelet decomposition follows motion compensation, the multiresolution scheme loses its flexibility and does not allow for compatibility between different picture formats. Irrespective of the desired signal resolution, all the subbands must be received so as to allow for proper motion compensation.

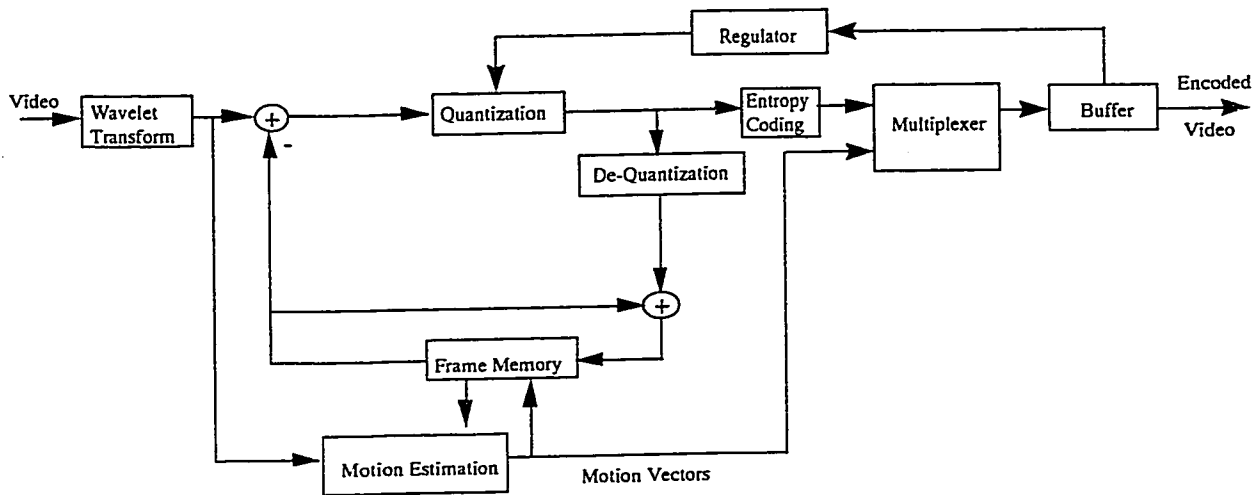


Figure 2.11. A typical wavelet-based video encoder

Subsequently, a few variations of the second scheme have been proposed in the literature to improve the coding performance. The main focus is to reduce the complexity of the motion estimation scheme and improve the bit-rate. We note that the block-matching motion estimation technique is a compute-intensive task. It was discussed in section 2.3.1 that the full search algorithm (FSA) has a complexity on the order of $4\Delta_u\Delta_v$ operations/pixel where

Δ_u and Δ_v are maximum allowed motion search ranges. Several fast methods for block-based motion estimation were described in section 2.3.1. However, these algorithms may converge to a local optimum, which corresponds to the inaccurate prediction of the motion vectors, resulting in a relatively poor performance.

Uz *et al.* [117] proposed a multiresolution motion estimation (MRME-U) scheme which exploits the multiresolution property of the wavelet pyramid in order to reduce the computational complexity of the motion estimation process. In the MRME-U scheme, motion vectors at the highest level of the wavelet pyramid (S_8 in Fig. 2.9) are first estimated using conventional block matching based motion estimation. The motion vectors at the next level (*i.e.*, Level 2 in Fig. 2.9) of the wavelet pyramid are then predicted from the motion vectors of the preceding level (*i.e.* Level 3) which are refined at each step. For example, the motion vectors in W_4^H , W_4^V , and W_4^D are predicted from the motion vectors in W_8^H , W_8^V and W_8^D using the following equation.

$$V_4^o(x, y) = 2V_8^o(x, y) + \Delta_4^o(x, y) \quad (2.21)$$

where $V_i^o(x, y)$ represents the motion vector of the reference block centered at (x, y) for o -orientation ($o \in \{H, V, D\}$) subband at i th scale of pyramid. Similarly, the motion vectors in W_2^H , W_2^V , and W_2^D are predicted using the following equation.

$$V_2^o(x, y) = V_4^o(x, y) + 2V_8^o(x, y) + \Delta_2^o(x, y) \quad (2.22)$$

In this scheme, identical block size ($m \times n$) and search region ($[-\Delta_u, \Delta_u] \times [-\Delta_v, \Delta_v]$) are used for all levels of pyramid. We note that the search area is small (because of decimation and refinement) in this scheme resulting in a reduced complexity. In addition, since the dynamic range of the motion vectors is small, the number of bits needed to encode the motion

vectors will also be less than that needed for full resolution motion vectors. We note that in MRME-U scheme, the number of blocks for motion estimation quadruples in each successive higher resolution subbands (since the block size is same for all subbands). Hence, the corresponding blocks in various subbands do not refer to a particular image area.

Zhang *et al.* [135] have proposed a variable size MRME technique (MRME-Z) where the block sizes are quadrupled and the search areas are reduced in each successive higher resolution subbands. The subimages of Level-3, Level-2 and Level-1 pyramids (in a 3 level decomposition) are divided into small blocks of size $m \times n$, $2m \times 2n$ and $4m \times 4n$, respectively. With this structure, the number of blocks in all subimages is identical. As a result, there is a one-to-one correspondence between the blocks at various levels of wavelet pyramid. Let the maximum allowed displacement (MAXAD) for level-3 (S_8 and W_8 's) subimages be (Δ_u, Δ_v) pixels. The maximum allowed refinements (MAXAR) in level-2 (W_4 's) and level-1 (W_2 's) subimages are then set to $(\Delta_u/2, \Delta_v/2)$ and $(\Delta_u/4, \Delta_v/4)$ pixels, respectively. The refinement of the motion estimation process is shown in Fig. 2.12.

Table 2.2 compares the complexity of the FSA scheme with the MRME (*i.e.*, MRME-U and MRME-Z) techniques. We note that MRME reduces the computational complexity significantly. Although, the MRME-U and MRME-Z have similar complexity, the latter provides a lower bit-rate (since the number of motion vectors is smaller). It has been observed that the MRME-Z provides superior motion compensation at a significantly reduced complexity [135].

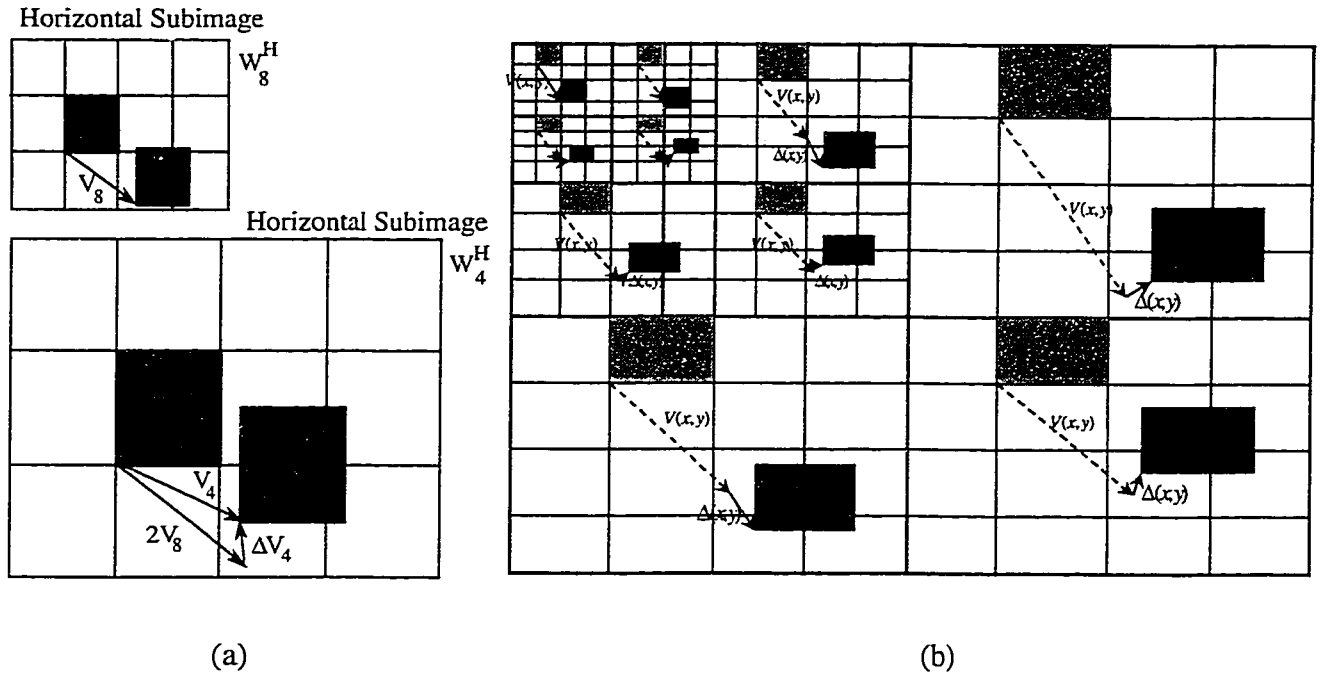


Figure 2.12. Multiresolution motion estimation. a) Level-3 to Level-2, b) Level-3 to Level 1.

Table 2.2

Computational complexity of motion estimation algorithms. The complexity is shown in average operations per pixel. (Δ_u, Δ_v) is the search area for full search ME in pixel domain and (Δ_u, Δ_v) is the search area for MRME at Level-3.

	Computational Complexity	Typical value
Full Search ME	$\approx 4\Delta_u\Delta_v$	1024^+
Multiresolution ME (MRME-U/MRME-Z)	$\approx 0.6\Delta_u\Delta_v + 0.7\Delta_u + 0.7\Delta_v$	15.2^{++}

$^+ \Delta_u = \Delta_v = 16, \quad ^{++} \Delta_u = \Delta_v = 4$

Kim *et al.* [54] have proposed two methods to improve the performance of multiresolution motion estimation. The first method reduces the entropy of the motion information, while the second method reduce the number of motion vectors using merge

operation on the quadtree structure. It has been reported that the two methods provide a significant improvement in performance.

2.4.4 Evaluation of Coding Performance

The performance of a video coder is generally measured in terms of the minimum bit-rate required to obtain a given quality of the reconstructed video. The visual quality of a reconstructed video is generally estimated by visual comparison of original and reconstructed video. However, this is a difficult and tedious process. Hence, in this thesis, the *peak signal to noise ratio* (PSNR) which is defined as:

$$\text{PSNR (in dB)} = 10 \log_{10} \left(\frac{(255)^2}{MSE} \right)$$

is employed as an objective measure of the quality of the reconstructed images. The coding performance of various techniques is compared with respect to bit-rate versus PSNR.

The coding performance of a typical video coder on pingpong sequence is shown in Fig. 2.13. It is observed that the overall bit-rate depends on factors such as entropy of motion vectors and DFD energy. It is difficult to present the coding performance of a video coder over a wide bit-rate using plots shown in Fig. 2.13, as many such plots will be necessary. Hence, in this thesis, we present the entire plot of Fig. 2.13 as a single point (representing the PSNR and bit-rate averaged over all frames) in the rate-distortion plane. For example, the overall bit-rate and PSNR shown in Fig. 2.13 has an average value of 0.45 *bpp* and 26 dB, respectively. Henceforth, the performance of a video coder will be presented using average PSNR and bit-rate.

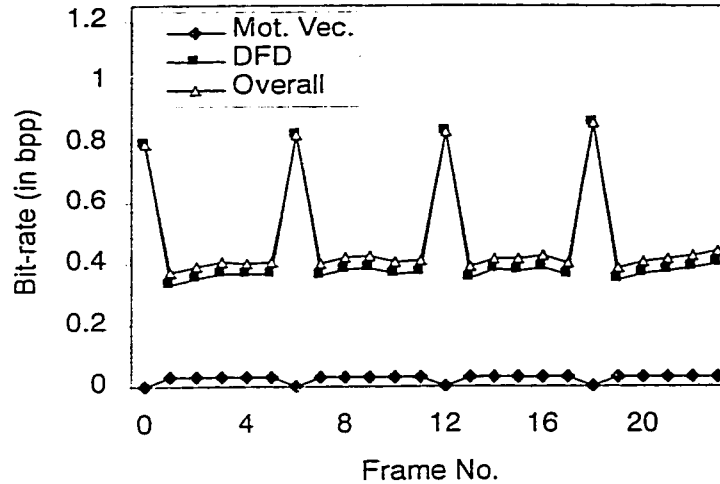


Figure 2.13: Typical coding performance of MRME technique on Pingpong sequence. Average bit-rate (over 24 frames) is 0.45 *bpp*, average PSNR (over all frames) is 26 dB, and average PSNR (over P frames) is 25 dB.

2.5 Summary

In this chapter, we have presented a comprehensive review of various image and video compression techniques. First, we reviewed an image model and several video data formats. This was followed by a review of current image coding techniques. The extension of image compression techniques to video compression was then discussed. We also briefly presented various block-matching motion compensation techniques. This was followed by a discussion of a few international video compression standards such as H.261, and MPEG. A brief review of wavelets, and their application in image and video coding was then presented. In addition, efficient techniques for motion estimation in the wavelet domain were also discussed.

Chapter 3

Review of Image and Video Indexing Techniques

Digital image and video indexing techniques are becoming increasingly important with the recent advances in very large scale integration technology (VLSI), broadband networks (e.g., ISDN and ATM), and image/video compression standards (e.g., JPEG and MPEG). The goal of visual indexing is to develop techniques that provide the ability to store and retrieve visual data based on their content [26]. Some of the potential applications of image and video indexing are: digital libraries [21], multimedia information systems [22], remote sensing and natural resources management [23], movie industry and video on demand [62], law enforcement and criminal investigation. Traditional databases use keywords as labels to quickly access large quantities of text data. However, the representation of visual data with text labels needs a large amount of manual processing. In addition, the retrieval results may not be satisfactory since the query is based on features that may not completely reflect the visual content. Hence, there is a need for novel techniques for content-based indexing of visual media.

A block schematic of a typical image archival and retrieval system is shown in Fig. 3.1. A multidimensional feature vector is generally computed for each image, and indexing is performed based on the similarities of the feature vectors. Since the interpretation or quantification of various features is fuzzy, emphasis is typically placed on the similarity rather than the exactness of the feature vectors. In indexing applications, a feature is selected based on i) its capacity to distinguish between different images, ii) the maximum number of images a query could possibly retrieve, and iii) the amount of computation required to compute the corresponding feature vector.

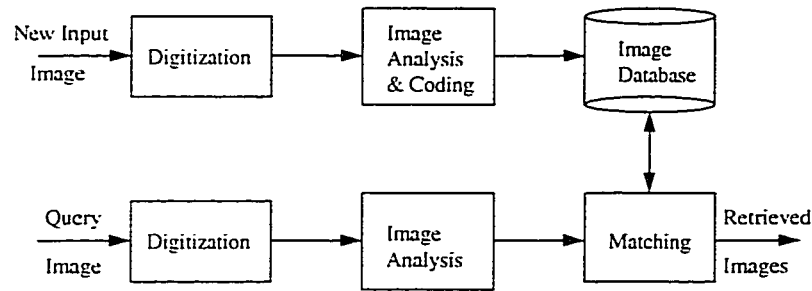


Figure 3.1. A schematic of image archival and retrieval system

Recently, several review papers on indexing have appeared in the literature. Aigrain *et al.* [2] have surveyed approaches for different types of visual content analysis, representation and their application in indexing, retrieval, abstracting and relevance assessment. Idris *et al.* [41] have presented a review of image and video indexing techniques pointing out the advantages and disadvantages of each approach. Ahanger *et al.* [1] have reviewed current research trends in multimedia applications and requirements of future data delivery systems, including a review of selected video segmentation techniques. Recently, we have presented a detailed review.[72] of image and video indexing techniques in the compressed domain.

In this section, we present a review of various indexing techniques. The organization of this section is as follows. A review of pixel domain indexing techniques is presented in section 3.1. In section 3.2, a review of compressed domain image indexing techniques is presented. Illumination invariant techniques are discussed in section 3.3. A review of pixel domain and compressed domain video indexing techniques is presented in sections 3.4, and 3.5, respectively. Various similarity metrics employed in indexing are presented in section 3.6 while the performance evaluation criteria for indexing techniques are presented in section 3.7. Present trends in integrating coding and indexing are discussed in section 3.8. We conclude the chapter with a summary.

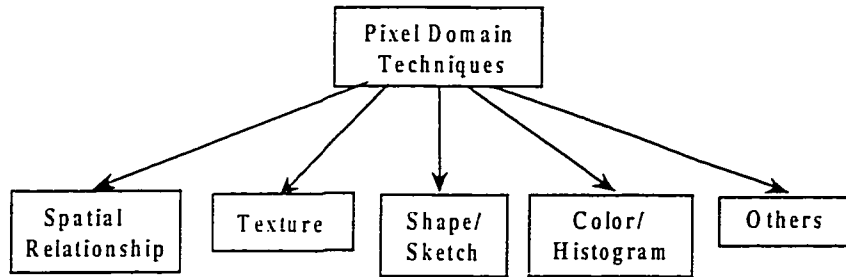


Figure 3.2: Various methods in content based image indexing in the pixel domain

3.1 Image Indexing in Pixel Domain

Pixel domain indexing of visual data is generally based on features such as texture, shape, sketch, histogram, color, and moments. For example, the *Query By Image Content* (QBIC) system developed by IBM [24] retrieves images based on color, texture, shape, and sketch. The *Content-based Retrieval Engine* (CORE) for multimedia information systems proposed by Wu *et al.* [127] employs color and word similarity measures to retrieve images based on

visual content and text annotation, respectively. We now briefly describe various features employed in pixel domain image indexing.

3.1.1 Histogram

The histogram of a digital image is a discrete function $h_i[k] = n_k$, where n_k is the number of pixels in the image with gray level k . The function, $p[k] = n_k/N_p$ gives an estimate of the probability of occurrence of a gray level k , where N_p is the total number of pixels in the image. If the total number of gray levels in an image is N_G , the histogram space Σ_H can be represented as a subset of N_G dimensional vector space:

$$\Sigma_H = \left\{ (h_i[1], \dots, h_i[k]) \mid h_i[k] \geq 0 (1 \leq k \leq N_G), \sum_{k=1}^{N_G} h_i[k] = N_p \right\} \quad (3.1)$$

The histogram of an image provides a global description of the appearance of an image. It has been observed that similar images at similar illumination level have similar gray-level distribution. The gray level distribution is invariant to image rotation and has been seen to change slowly with translation. Thus, the low sensitivity of image histograms to camera and object motion has made it a popular technique for indexing applications [76, 107, 108]. In addition, histogram-based techniques have a lower complexity compared to the classical techniques of pattern recognition, facilitating real time implementation. Figs. 3.3(a)-(e) show a set of 5 images with various camera operations. Fig. 3.4 plots the corresponding histograms that are seen to be very similar. During retrieval, the histogram of the query image is compared to the histograms of all the images in the database. The images with least difference of histograms are retrieved. Henceforth, we denote this as DOIH (difference of image histogram) technique.



(a)



(b)



(c)



(d)



(e)

Figure 3.3. An image with five different orientations

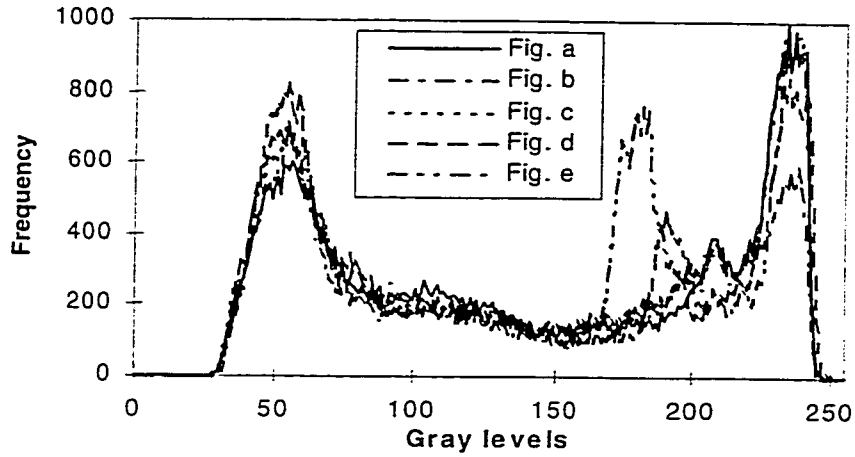


Figure 3.4. Histograms of five images shown in Fig. 3.3

Stricker *et al.* have derived a lower bound for the capacity of histogram space [105]. Given an M -dimensional histogram space Σ_H , and a distance threshold T_H , the capacity of Σ_H is defined as the maximal number of T_H -different histograms (*i.e.*, histograms having distance greater than T_H) that fit into Σ_H . Table 3.1 shows the capacity of the histogram space with respect to distance threshold T_H . It is observed that the capacity is very high, especially for small thresholds. In other words, the probability that histograms of two randomly selected images will be similar, is very small. This is an important result since it emphasizes the applicability of histogram-based techniques.

3.1.2 Color

Color is an important attribute of an image and hence has become popular in image indexing applications [76, 107, 108]. The colors of two images are generally matched by comparing the corresponding histograms of the three color channels (*e.g.*, R-G-B, or Y-I-Q). Stricker *et al.* [107] have proposed to compare the cumulative histogram (*i.e.* distribution

function) of each color channel (R, G and B) in L^1 metric. In some cases, the image histogram is influenced by the image background, which is not desirable. In order to reduce the effect of the background, Swain *et al.* have proposed to use histogram intersection for matching color images [108]. The retrieval performance can be improved by taking into account the location (*i.e.*, spatial information) of the colors in the representation of an image [107]. However, this technique requires the use of efficient segmentation and representation of the subimages.

Table 3.1

The capacity of histogram space with a distance threshold T_H (in L^1 metric). N_G is the number of histogram bins.

	Capacity	
$T_H / (2 * N_G)$	$N_G=64$	$N_G=256$
0.20	6.52E+10	2.48E+31
0.25	1.62E+09	8.87E+25
0.30	4.36E+07	3.24E+20
0.35	2.47E+06	8.02E+15
0.40	5.03E+05	6.30E+12
0.45	3.69E+04	1.47E+10
0.50	1.67E+04	1.43E+08
0.55	2.06E+03	2.58E+06
0.60	1.69E+03	2.48E+05

3.1.3 Texture

Textures are useful in describing the content of an image. This descriptor generally provides some measures such as: smoothness, coarseness, granularity and regularity [49]. Recently, several techniques for image indexing based on texture features have been reported. Picard *et al.* [85] have presented a technique based on Wold decomposition which provides a

description of textures in terms of periodicity, directionality and randomness. A modified set of the Tamura features (coarseness, contrast and directionality) [110] has been used in the QBIC project [24]. Zhang *et al.* [132] have proposed a technique based on a multiresolution autoregressive model, Tamura features, and gray level histogram. Rao *et al.* [90] have studied the relationships between categories of texture images and texture words. Retrieval by texture is useful when the user is interested in retrieving images that are similar to the query image. The main disadvantages of texture based techniques are that i) they are computationally expensive, ii) texture models are not robust, and iii) texture parameters do not correlate well with the human perception.

3.1.4 Shape/Sketch

Shape is an important criterion for matching objects based on their profile and physical structure. In shape-based image indexing [112], the image is first segmented into objects or regions. The shape parameters such as geometrical attributes (*e.g.*, boundary, region), normalized moments, Fourier descriptors are then calculated. The features of the query image are then compared with those of the target images in the database. Some important global shape parameters [49] are – i) *compactness*: a measure of the roundness of an object boundary; ii) *eccentricity*: ratio of the length of the major axis to the length of the minor axis of the object; iii) *corners*: locations on the boundary where the curvature becomes unbounded; and iv) *region convexity*: a measure of the convex hull of the region enclosed by the boundary.

Although, shape parameters are useful, they are difficult to estimate. In many cases, the user has a fuzzy idea about the shape of an image or object. In these instances, a better

approach will be to use sketch as the input [11, 35]. We note that a sketch is an abstract image that contains the outline of objects in the original image. Here, the users provide a rough sketch of the query image. In order to obtain the sketch of the target images, an edge detection operation is performed on all images present in the database. A shrinking and thinning operation is then performed on the edges. The sketch of the query image is then compared with those of the target images in the database based on local and global correlation [35].

3.1.5 Spatial Relationships

In this technique, the content of an image is represented by the objects contained in the image and their spatial relationships [14, 32]. First, an image is segmented and various objects are labeled. The image is then converted into a symbolic picture that is encoded using 2-dimensional (2-D) strings. The 2-D string represents relationships among the objects in the image and is expressed using a set of operators (*e.g.*, left, right, above, etc.). The problem of image retrieval thus becomes a problem of 2-D sequence matching. We note that the generation of a 2-D string is based on object segmentation and recognition which is compute intensive.

3.1.6 Moments

Traditionally, moments have been widely used in pattern recognition applications to describe the geometrical shapes of different objects. They provide fundamental geometric properties (*e.g.*, area, centroid, moment of inertia, skewness, kurtosis) of a distribution [86]. The moments can also be used to represent the *pdf* of pixel intensities of the image [107]. We

note that the *pdf* of pixel intensities is the same as the histogram except for a scale factor which makes the total area under the *pdf* equal to unity. Hence, we use the term *pdf* and *histogram* interchangeably in this thesis wherever appropriate.

3.1.6.1 1-D Moments of Histogram

We recall from section 3.1.1 and 3.1.2 that color and gray level histograms of an image are popular in image indexing applications. The matching process is carried out using a similarity metric, such as L^1 metric. The complexity of the matching process can be reduced by employing dominant features of a histogram, such as moments, Fourier descriptor, and polynomial descriptors. Moments have become a popular descriptor of image histogram. The k th order regular, central, and normalized central moments of a function $f(x)$ are defined as follows.

$$M_k = \int_{-\infty}^{\infty} x^k f(x) dx, \quad k \geq 0, k \in \mathbb{Z} \quad (3.2)$$

$$\mu_k = \int_{-\infty}^{\infty} (x - \bar{x})^k f(x) dx \quad (3.3)$$

$$\beta_k = (\mu_k)^{1/k} \quad k > 0, k \in \mathbb{Z} \quad (3.4)$$

where $\bar{x}$ is the mean of the function $f(x)$. The regular moments M_k are the projections of $f(x)$ onto monomials $\{x^k\}$. The central moments μ_k are projections onto $\{(x - \bar{x})^k\}$ and are thus invariant to translation. We note that the magnitude of moments may grow exponentially with increasing order and hence it is difficult to compare them. This problem can be circumvented by using normalized moments β_k , which are defined in Eq. 3.3.

Generally, a few lowest order moments are sufficient to differentiate the global shapes of the histogram. Fig. 3.5 shows the *pdf* of a typical image and its approximation using 8, 12,

16 regular moments. It is observed that the histogram can be represented efficiently using a few moments. Exploiting this concept, Stricker *et al.* [107] have proposed an indexing technique based on *difference of central moments* (DOCM). The distance measure between two images Q and C are measured using the following [107]:

$$d_{mom}(Q, C) = w_0 |Mean_Q - Mean_C| + \sum_{k=1}^{N_M} w_k |\beta_{Q_k} - \beta_{C_k}| \quad (3.5)$$

where

w_k = weight of the k th moment

β_{Q_k} = k th normalized central moment of histogram of Q

$Mean_Q$ = Mean of image Q , equivalently, the first regular moment of its histogram

N_M = the number of moments employed for indexing

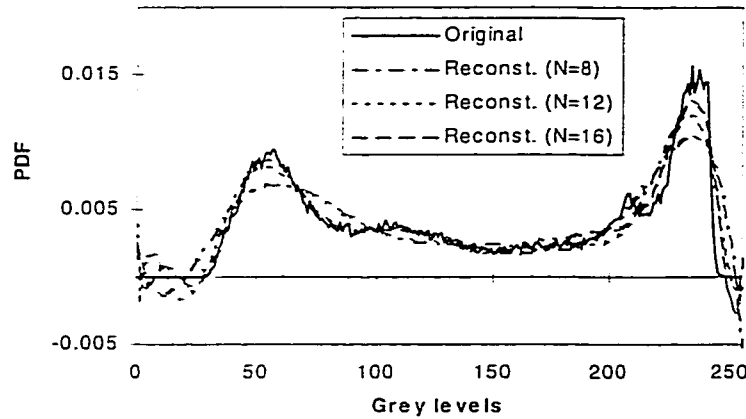


Figure 3.5. Original and reconstructed *pdf*'s. N is the number of moments used for reconstruction.

In addition to reduced complexity, the DOCM technique may provide a superior performance compared to DOIH (difference of image histogram) technique. For example, Fig. 3.6 shows the *pdf* of a test image, where the density is zero for several gray-levels. A small change in the illumination will cause a shift in the *pdf*, which may result in a large

DOIH. However, if moments are used, this problem will be minimal since it will smooth the *pdf*.

3.1.6.2 2-D Moments

The histogram, color and moment techniques described above reduce the description of an image to the specification of 1-D function. In these techniques, the structural property of the images is not exploited. Better performance can be achieved by treating the image as a 2-D function. A 2-D image can be represented by various 2-D descriptors, such as 2-D moments and 2-D DFT. We note that 2-D moments are very popular for pattern recognition. For a 2-D continuous function $f(x,y)$, the regular, central and central normalized moments of order $(p+q)$ are defined as [29]:

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y) dx dy \quad \text{for } p, q = 0, 1, 2, \dots \quad (3.6)$$

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q f(x,y) dx dy \quad (3.7)$$

$$\beta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{(p+q+2)/2}} \quad \text{for } p+q > 1 \quad (3.8)$$

where $\bar{x} = M_{10} / M_{00}$, $\bar{y} = M_{01} / M_{00}$. For image retrieval applications, the retrieval technique should be sophisticated enough to identify images which are rotated, translated and scaled versions of the query image. The regular moments defined in the above equation do not possess any of the three properties. To achieve translation invariance, one can employ 2-D central moments. The normalized moments are used to nullify the effect of exponential growth of moment-magnitudes with increasing order.

Based on theory of algebraic invariants, Hu proposed several rotation, translation, and scale invariant 2-D moments [36]. Example of the Hu's moments includes:

$$\phi_1 = \beta_{2,0} + \beta_{0,2} \quad (3.9)$$

$$\phi_2 = (\beta_{2,0} - \beta_{0,2})^2 + 4(\beta_{1,1})^2 \quad (3.10)$$

$$\phi_3 = (\beta_{3,0} - 3\beta_{1,2})^2 + (\beta_{0,3} - 3\beta_{2,1})^2 \quad (3.11)$$

$$\phi_4 = (\beta_{3,0} + \beta_{1,2})^2 + (\beta_{0,3} + \beta_{2,1})^2 \quad (3.12)$$

All the moments described above, *i.e.*, regular, central, Hu's invariant moments, are non-orthogonal. 2-D Zernike moments are popular in pattern recognition [86, 111] because of their orthogonality. In addition, the magnitude of the Zernike moments are rotation invariant.

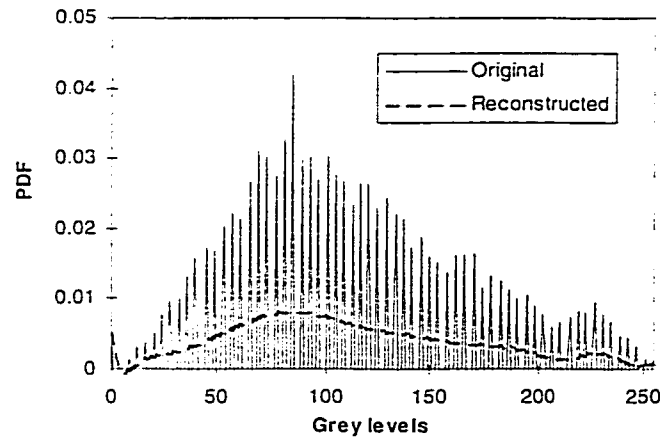


Figure 3.6. A case where direct comparison of histogram is likely to fail, but moment based method will work.

3.2 Image Indexing in the Compressed Domain

The large volume of visual data necessitates the use of compression techniques. Hence, visual data in future multimedia databases is expected to be stored in compressed form. In order to obviate the need to decompress the image data and apply pixel domain indexing

techniques, the indexing should be performed on the compressed data (see Fig. 3.7). Several compressed-domain indexing (CDI) techniques based on compression parameters have been reported in the literature (see Fig. 3.4). CDI techniques can be broadly classified into two categories, namely spatial domain and transform domain techniques. Spatial domain techniques include vector quantization [39, 118] and object oriented techniques [38, 131], whereas transform domain techniques are generally based on DFT [15, 105], KLT [84, 116], DCT [101, 102], and Subbands/Wavelets [16, 48, 103, 122]. We note that hybrid techniques [39, 109] have also been proposed in the literature.

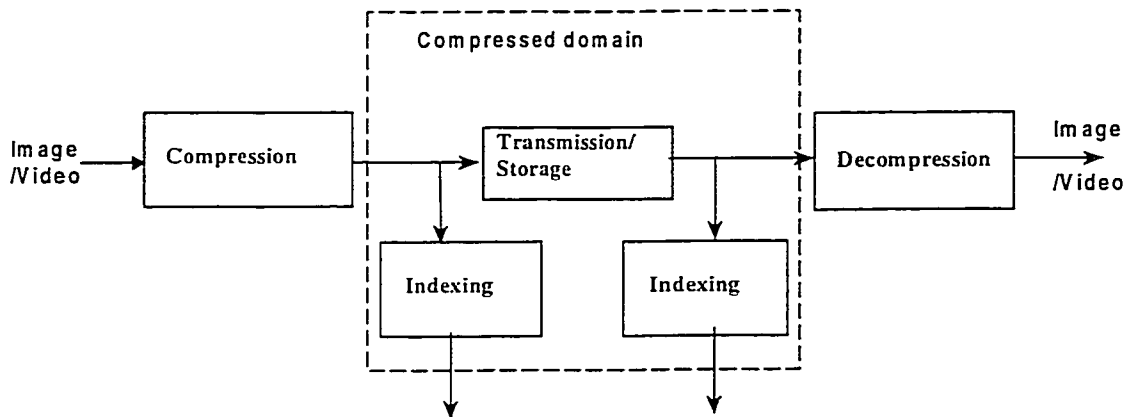


Figure 3.7: Block diagram of a compressed domain indexing system

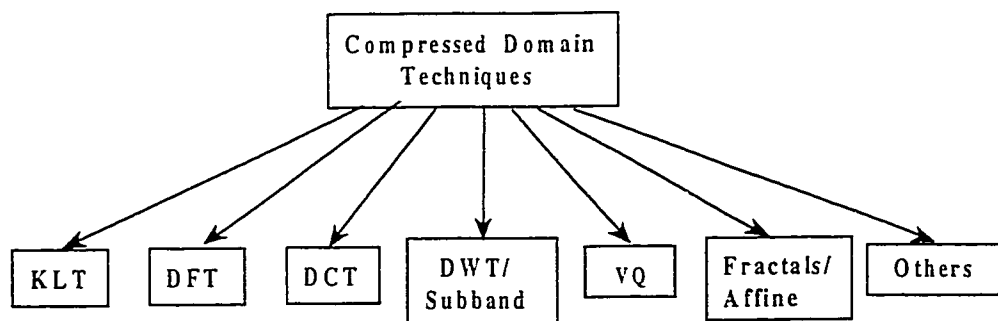


Figure 3.8: Various methods in content based image indexing in the compressed domain

Fig. 3.9 shows a typical schematic of image indexing in the transform domain. Generally, the transform coefficients (or its features) of the query image are compared with the corresponding coefficients (or, features) of a candidate image to find a match. In this respect, various transform-domain techniques are similar. However, each transform has its own idiosyncrasies, and hence has different indexing performance. A detailed review of CDI techniques can be found in [72]. Here, we present a brief review of DCT and wavelet-based techniques.

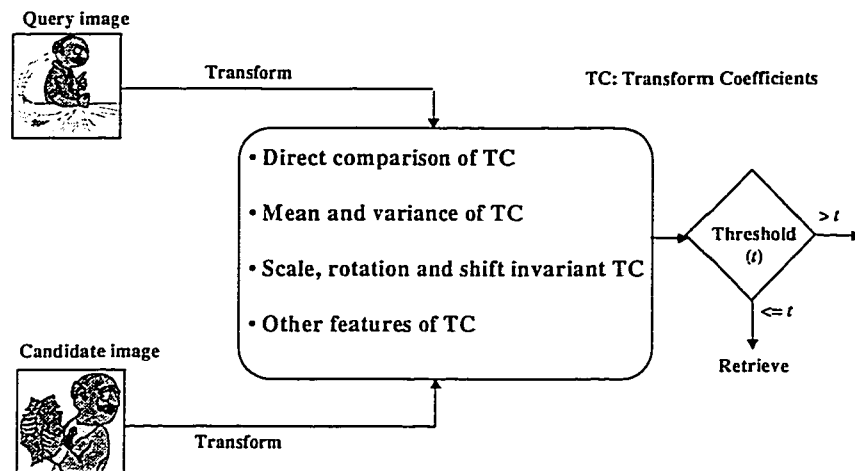


Figure 3.9: Image indexing and retrieval using transform coefficients

3.2.1 Discrete Cosine Transform

We recall from section 2.2.2 that DCT has excellent image compression efficiency and hence has been employed in several compression standards. Several techniques have been proposed in the DCT domain for image indexing. Smith *et al.* [102] have proposed a DCT-based indexing technique, where the image is divided into 4x4 blocks and the DCT is computed for each block resulting in 16 coefficients. The variance and the mean absolute

value of each of these coefficients are calculated over the entire image. The texture of the entire image is then represented by a 32 component feature vector, which is used for indexing. Shneier *et al.* [101] have proposed a technique for image retrieval using JPEG. This technique is based on the mutual relationship between the DCT coefficients of unconnected regions in both the query and target images.

3.2.2 Subbands/Wavelets

Several techniques have recently been proposed for indexing in the subband/wavelet domain. Wavelets have a potential to provide good indexing capability for several reasons. First, indexing can be done hierarchically using multiresolution feature capability. Secondly, edge and shape of objects can be determined efficiently in the wavelet domain. Finally, directional information can be exploited to enhance the indexing performance.

Jacobs *et al.* [48] have proposed a fast multiresolution image querying technique (FMIQT) based on direct comparison of the DWT coefficients. Here, all images are rescaled to 128×128 pixels followed by wavelet decomposition. The average color, the sign (positive and negative) and indices of M (the authors have used a value of 40-60) largest magnitude DWT coefficients of each image are calculated. The indices for all of the database images are then organized into a single data structure for fast image retrieval. Although, a good indexing performance has been reported [48], the index is dependent on the location of DWT coefficients. Hence, the target images which are translated and rotated versions of the query image, may not be retrieved using this technique.

Wang *et al.* [122] have proposed a technique which is similar to that of Jacobs *et al* [48]. Let the four lowest resolution subimages be denoted by S_L (lowpass), S_H (horizontal band),

S_V (vertical band), and S_D (diagonal band). Image matching is then performed using a three-step procedure. In the first stage, 20% of the images are retrieved based on the variance of S_L band. In the second stage, a fewer number of images will be selected based on the difference of S_L coefficients of query and target images. Finally, the images will be retrieved based on the difference of S_L , S_H , S_V and S_D coefficients of query and target images.

A texture discrimination technique has been proposed by Smith *et al.* [103] based on wavelet coefficients. Here, the energy ϵ_k , of wavelet coefficient from each high pass band is calculated first. Then, ϵ_k is upsampled to the full size by inserting zeros. The missing points are then filled in using block filters $B_{i,j}$ (a simple pixel replication filter has been used in [103]) to obtain a texture channel. The whole process is illustrated in Fig. 3.10. Here, nine texture channels are generated from the DWT coefficients. A texture point is then defined as a 9-D vector by considering texture channel values from the same location of all nine bands. Thus for an $N \times N$ image, there will be N^2 9-D vectors. The authors have proposed to threshold each element of the 9-D vectors to two levels – high (1) and low (0). A wavelet histogram (with 512 bins) of all texture points, with 9-D thresholded vectors, is thus created. The wavelet histogram of the query image is compared to the corresponding histogram of the candidate images for retrieval.

Chang *et al.* [16] have proposed a texture analysis scheme using irregular tree decomposition where the middle resolution subband coefficients are used for texture matching. In this scheme, a J dimensional feature vector is generated consisting of the energy of J most important subbands. Indexing is done by matching the feature vector of the query image with those of the target images in a database. For texture classification,

superior performance can be obtained by training the algorithm. Here, for each class of textures, the most important subbands and their average energies are found by the training process. A query image can then be categorized in one of the texture classes, by matching the feature vector with those of the representative classes.

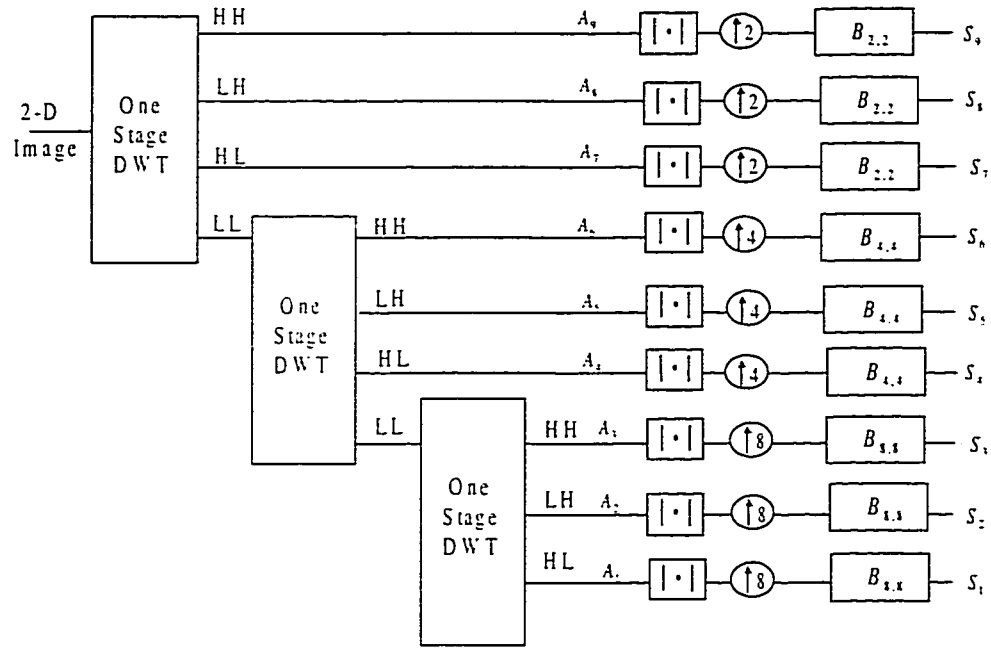


Figure 3.10. Wavelet histogram generation [103]

Qi *et al.* [87] have proposed a complex wavelet transform where the magnitude of the DWT coefficients is invariant under rotation. The mother wavelet is defined in the polar coordinates. An experiment on a set of English character images shows that the proposed technique performs better than complex Zernike moments (whose magnitudes are also rotation invariant). Rashkovskiy *et al.* [92] have proposed a class of nonlinear wavelet transforms which are invariant under scale, rotation and shift (SRS) transformations. This wavelet transform adjusts the mother wavelet for every input signal to provide SRS invariance. The wavelet parameters or the wavelet shape are iteratively computed to

minimize an energy function for a specific application. Although, these techniques have not been employed for indexing, they have potential to provide superior performance.

3.3 Illumination Invariant Indexing

Direct comparison of features, such as histograms, moments, *etc.*, provides a good indexing performance when the illumination level of the acquired images is at the same level. However, the illumination level depends on many factors such as changes in ambient lighting condition and camera flash. Hence, it is crucial to design indexing techniques that are invariant to varying illumination conditions.

Recently, Funt *et al.* [25] have proposed an illumination-invariant indexing technique based on the coefficient model (see section 2.1.1). For three color channels $\{R, G, B\}$, the image function of Eq. 2.1 can be written as:

$$i_k(x, y) = l_k(x, y)r_k(x, y) \quad \text{where } i_1 \equiv R, i_2 \equiv G, i_3 \equiv B \quad (3.13)$$

The ratio of sensor responses from two locations (x_1, y_1) and (x_2, y_2) under the same illumination conditions yields the ratio of surface reflectance, which is given below:

$$\frac{i_k(x_1, y_1)}{i_k(x_2, y_2)} = \frac{l_k(x_1, y_1)r_k(x_1, y_1)}{l_k(x_2, y_2)r_k(x_2, y_2)} \approx \frac{r_k(x_1, y_1)}{r_k(x_2, y_2)} = \vartheta_k(x_1, y_1, x_2, y_2) \quad (3.14)$$

By computing the logarithms of both sides of Eq. 3.14, the ratio is converted to differences:

$$\ln(i_k(x_1, y_1)) - \ln(i_k(x_2, y_2)) = \ln(r_k(x_1, y_1)) - \ln(r_k(x_2, y_2)) \quad (3.15)$$

Since the right hand side of Eq. 3.15 is independent of illumination, the left-hand side can be used for illumination independent indexing. The ratio-histogram technique (henceforth referred as RHT technique) is implemented as follows: i) logarithms of $\{R, G, B\}$ channels are computed, ii) a convolution operator (with functions such as *Laplacian*) is then applied to the logarithm values, iii) a 3-D histogram is computed from the convolution output. This 3-D histogram is then used as an index.

The main disadvantage of this technique is that it can be applied only on $\{R, G, B\}$ images that are generally not used for compression. For example, JPEG standard employs $\{Y, Cb, Cr\}$ color space to obtain a superior coding performance. Since, the ratio histogram technique is not readily extendible to $\{Y, Cb, Cr\}$, the images have to be first converted to $\{R, G, B\}$ which increases the complexity. In addition, this technique involves logarithmic operations followed by a two-dimensional convolution. Hence, the overall complexity is high.

3.4. Video Indexing in Pixel Domain

A video is a sequence of image frames ordered in time. Therefore, it is natural to apply image indexing techniques described in section 3.1 to each frame individually. However, we note that the neighboring frames are generally highly correlated. Hence the video sequence is segmented in a series of *shots* for higher efficiency. A *shot* is defined as a sequence of frames generated during a continuous operation, and which represents a continuous action in time and space. A frame in each shot is declared as a representative frame. Retrieval is accomplished by comparing the query image with representative frames from each shot. We

note that comparison of query and candidate images are executed based on image indexing techniques discussed in sections 3.1-3.3. Hence, we now present a review of video segmentation techniques. The pixel domain techniques are presented below. The compressed domain techniques will be presented in section 3.5.

3.4.1 Video Segmentation in Pixel Domain

We note that there are two ways by which consecutive shots can be joined - i) abrupt transition, and ii) gradual transition. In the former, the scene change is abrupt and the frames from two consecutive shots have little correlation. In the latter, the frame contents change gradually from one scene to another. This is generally observed when two scenes merge, fade in, fade out or dissolve. An efficient video segmentation technique should be able to detect shots with both types of transition. The video segmentation techniques in the pixel domain can be divided into four categories - i) pixel intensity matching, ii) histogram comparison, iii) block-based techniques, and iv) twin-comparison method. We present a brief review of each of these techniques.

In the *pixel intensity matching* technique [53], pixel intensities of the two neighboring frames are compared. For example, to detect a scene change between m -th and $(m+1)$ -th frame, the distance between the two frames is calculated with L^k metric. If the distance exceeds a predetermined threshold, a scene change is declared at the m -th frame.

In the *histogram comparison* technique [79, 134], two consecutive frames are compared based on their histograms. There are two variations of this technique. The DOH (difference of histogram) technique measures the difference of histograms of the two frames while the

HODF (histogram of difference frame) technique is based on the histogram of the pixel to pixel difference frame and measures the change between two frames f_m and f_n . The change between f_m and f_n is large if there are more pixels distributed away from the origin.

In the *block-based* technique [59], each frame is partitioned into a set of k blocks. The similarity of the consecutive frames is estimated by comparing the corresponding blocks. In *Block Histogram Difference* (BHD) technique, the blocks are compared with respect to histogram of individual blocks, whereas in *Block Variance Difference* (BVD) technique the blocks are compared with respect to their variance.

The previous segmentation techniques are based on thresholding. With a single threshold, it is difficult to detect the two types of scene changes, namely *abrupt* and *gradual*. If the threshold is small, the cuts will be over-detected. On the other hand, gradual cuts will be undetected if the threshold is large. A two-pass dual threshold algorithm, known as twin comparison algorithm (see Fig. 3.11), has been proposed in [134] to address this problem. In the first pass, a high threshold (T_h) is employed to detect abrupt cuts. In the second pass, a lower threshold (T_l) is used, and any frame that has the difference more than this threshold is declared as a potential start of the transition. Once the start frame is identified, it is compared with the subsequent frames based on the cumulative difference. When this value increases to the level of the higher threshold (T_h), a camera break is declared at that frame. If the value falls between the consecutive frames, then the potential frame is dropped and the search for another transition starts afresh.

We note that a video sequence can also be segmented using associated audio information. Huang *et al.* [37] have proposed a joint audio and visual technique for video segmentation.

The proposed technique is computationally inexpensive and seems to be promising. Nakano [80] has proposed an indexing technique based on object motion. Recently, Irani and Anandan [43] have proposed a video segmentation technique using mosaic representation. Here, a scene is represented using a panoramic mosaic image. It has been reported that this representation provides a good coding as well as indexing performance.

3.5. Video Indexing in Compressed Domain

Video indexing techniques in the compressed domain have been proposed in the literature employing VQ [40], DCT [7], DWT [59], *etc.* Given that such techniques require robust segmentation, we present a brief review of video segmentation techniques in DCT and DWT domains. We note that motion vector, which is not available for image indexing, is an important feature for video segmentation. Hence, a review of motion vector-based video segmentation is also presented.

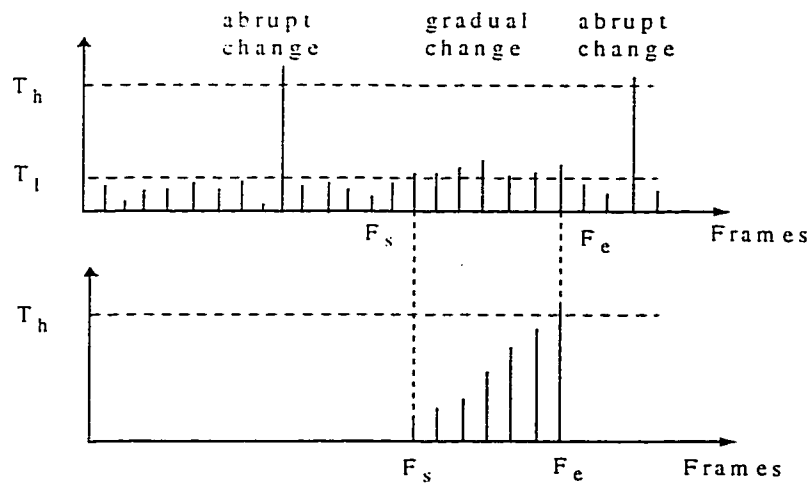


Figure 3.11. Shot detection by twin comparison technique [134].

3.5.1 Video Segmentation using DCT Coefficients

The video indexing techniques using DCT coefficients have generally been proposed in MPEG framework. Zhang *et al.* [133] have presented a pair-wise comparison technique for the I-frame, where the corresponding DCT coefficients in the two frames f_m and f_n are matched. This is similar to the pixel intensity matching technique (see section 3.4), but executed in the frequency domain. Here, the pairwise normalized absolute difference $D(f_m, f_n, l)$ (see Eq. 3.21) of the l block in two frames f_m and f_n are first calculated. If the difference $D(f_m, f_n, l)$ is larger than a threshold, the block l is considered to have changed. If the number of changed blocks exceeds a certain threshold, a scene change is declared in the video sequence from frame f_m to frame f_n .

Arman *et al.* [7] have proposed a technique based on the correlation of corresponding DCT coefficients of two neighboring frames. For each compressed frame f_m , B blocks are first chosen *a priori* from R connected regions in f_m . A set of randomly distributed coefficients $\{c_x, c_y, c_z, \dots\}$ is selected from each block where c_x is the x th coefficient. A vector $Vf_m = \{c_1, c_2, c_3, \dots\}$ is formed by concatenating the sets of coefficients selected from the individual blocks in R . The vector Vf_m represents f_m in the transform domain. The normalized inner product (see Eq. 3.22), of the feature vectors of the query and candidate image is then calculated to find the similarity.

3.5.2. Video Segmentation using Subband Coefficients

Video segmentation can be performed efficiently [59, 81] exploiting the multiresolution properties of wavelets/subbands. Lee *et al.* [59] have proposed a hierarchical video

segmentation technique in the subband domain. Here, the histogram-based video segmentation technique is first applied on the coarsest resolution (say Level-m) lowpass subimage. The segmentation result is refined by applying the segmentation technique recursively on higher resolution lowpass subimages. We note that most of the decisions on scene change and similarity can be taken by comparing one or two coarsest resolution lowpass subimages. However, when this is not possible, the results can be refined using the next higher resolution subimages.

This hierarchical approach results in a reduction in computational complexity. We note that the complexity of a spatial domain technique is generally proportional to the number of pixels of the image (say, $N \times N$). In a multiresolution approach with K -level pyramid, the lowpass coarsest resolution subimage has only $N^2/2^{2K}$ pixels. Therefore, a substantial reduction in complexity is achieved for $K \geq 2$.

3.5.3. Video Segmentation using Motion Vectors

Motion analysis is an important step in video processing [5]. A video stream is composed of video elements constrained by the spatio-temporal piecewise continuity of visual cues. The normally coherent visual motion becomes suddenly discontinuous in the event of scene changes or new activities. Hence, motion discontinuities may be used to mark the change of a scene, the occurrence of occlusion, or the inception of a new activity.

Shahraray *et al.* [97] have proposed a technique based on motion-controlled temporal filtering of the disparity between consecutive frames to detect abrupt and gradual scene changes. A block matching process is performed for each block in the first image to find the

best fitting region in the second image. A nonlinear statistical filter is then used to generate a global match value. Gradual transition is detected by identification of sustained low-level increases in matched values.

In MPEG, B- and P-frames contain the DCT coefficients of the error signal and motion vectors. Liu *et al.* [63] have presented a technique based on the error signal and the number of motion vectors. A scene cut between a current P- frame f_n^P and the corresponding past reference frame f_n^R increases the error energy. Hence, the error energy is employed to find the similarity between f_n^P and the motion compensated frame f_n^R . For the detection of scene changes based on B-frames, the difference between the number of forward-predicted macroblocks F_P and backward predicted B_P is used. A scene change between a B-frame and its past reference B-frame will decrease F_P and increase B_P . A scene change is declared if the difference between F_P and B_P changes from positive to negative.

Zhang *et al.* [134] have proposed a technique for scene cut detection using motion vectors in MPEG. This approach is based on the number of motion vectors M . In P-frames, M is the number of motion vectors. In B-frame, M is the smaller of the counts of the forward and backward non-zero motion. Then $M < T$ will be an effective indicator of a camera boundary before or after the B- and P-frame, where T is a threshold value close to zero. However, this method yields false detection when there is no motion. This is improved by applying the normalized inner product metric to the two I-frames on the sides of the B-frame where a break has been detected.

3.6 Feature Similarity

A review of the current indexing techniques has been presented in sections 3.1-3.5. The feature vectors are generally compared using a distance metric. In this section, we present a review of selected distance metrics employed in indexing applications.

L^p Metric

The L^p distance between K -dimensional feature vectors q (for query image) and c (for target image) is given by:

$$d_{L^p}(q, c) = \sum_{i=1}^K |q(i) - c(i)|^p \quad (3.16)$$

The images corresponding to the vectors q and c are considered to be similar if the distance $d_{L^p}(q, c)$ is less than a predetermined threshold. Typically, absolute error (L^1), or square error (L^2) metrics (*i.e.* $p=1$ or 2) are used in indexing applications.

The complexity of computing L^1 and L^2 distances, in a straightforward manner, is $2K$ and $3K$, respectively. However, the complexity of calculating L^2 distance can be reduced as follows. Eq. 3.16 can be written as:

$$d_{L^2}(q, c) = \sum_i [q(i)]^2 - 2 \sum_i q(i)c(i) + \sum_i [c(i)]^2 \quad (3.16a)$$

We note that the first term in the right side of Eq. 3.16a is a constant (since the query image is fixed) present in each distance value, and hence can be ignored. The third term can be calculated offline and pre-stored, and thus need not be calculated. The second term, however, has to be calculated and needs $2K$ operations. Hence, the overall complexity is of the order of $2K$ operations.

Mahalanabis Distance

Mahalanabis distance is statistically the optimum distance between two feature vectors. It projects the error vector in an orthogonal space and calculates the magnitude of the vector. It is calculated as follows: Let us assume that the feature vectors are K -dimensional. Each feature vector can then be treated as a random vector of $\tilde{X} = [x_1, x_2, \dots, x_K]$. The expectation of $\tilde{X}$ can be expressed as

$$\mu_{\tilde{X}} = E\{\tilde{X}\} = [E\{x_1\}, E\{x_2\}, \dots, E\{x_K\}] \quad (3.17)$$

The covariance matrix of $\tilde{X}$ is defined as

$$\Lambda_{\tilde{X}} = E\left\{\left[\tilde{X} - \mu_{\tilde{X}}\right]^T \left[\tilde{X} - \mu_{\tilde{X}}\right]\right\} = [\lambda_{ij}] (i, j \in [1, K]) \quad (3.18)$$

where λ_{ij} is the covariance of x_i and x_j . The Mahalanabis distance between two feature vectors X_q and X_c can be expressed as:

$$D_{MD}(X_q, X_c) = (X_q - X_c) \Lambda^{-1} (X_q - X_c)^T \quad (3.19)$$

where Λ is the covariance matrix corresponding to the feature vectors of all images present in the database. We note that Λ is a diagonal matrix when x_i 's are statistically independent. In this case, the simplified Mahalanabis distance can be expressed as:

$$D_{SMD}(X_q, X_c) = \sum_{k=1}^K \left(\frac{x_{kq} - x_{kc}}{\sigma_k} \right)^2 \quad (3.20)$$

where, x_{kp} is the k -th element of the feature vector X_p and σ_k is the standard deviation of k -th element of $\tilde{X}$.

Pairwise normalized difference

The *pairwise normalized difference* $D_{PN}(q, c)$ of two K -dimensional feature vectors q and c is calculated by

$$D_{PN}(q, c) = \frac{1}{K} \sum_{k=1}^K \frac{|q(k) - c(k)|}{\max(q(k), c(k))} \quad (3.21)$$

The pairwise normalized difference has been employed by Zhang *et al.* [133] to find the similarity of DCT coefficients of neighboring frames.

Normalized Inner Product

The inner product reflects the cross correlation among two vectors. The *normalized inner product* of two feature vectors q and c is calculated by:

$$D_{NIP}(q, c) = 1 - \frac{q \bullet c}{\|q\| \|c\|} \quad (3.22)$$

The two vectors are considered similar if the difference is smaller than a threshold. The normalized inner product has been employed by Arman *et al.* [7] to find the similarity of DCT coefficients of neighboring frames for video segmentation.

Histogram Intersection

The common similarity metrics employed for evaluating color similarity are histogram intersection [108], and weighted distance between color histograms [31]. The intersection of two histograms q (query image) and c (candidate image) is calculated as follows:

$$D_{HI}(q, c) = 1 - \frac{\sum_i \min(q(i), c(i))}{\sum_i q(i)} \quad (3.23)$$

The match value ranges between $[0, 1]$. The two histograms are considered similar if $D_{HI}(q, c)$ is smaller than a certain threshold. We note that the histogram intersection technique reduces the influence of large background. However, when the two images have the same number of pixels, the histogram intersection technique becomes the equivalent of histogram difference technique using L^1 Metric.

3.7. Evaluation Criteria for Image Indexing Techniques

A variety of criteria have been employed in the literature for evaluating the performance of indexing techniques. In all cases, the entire database is manually indexed before the evaluation criteria are applied. Several query images are selected randomly from the database and the indexing/retrieval technique is applied in each case. The average performance is then used for comparison purposes. Here, we discuss the three most popular criteria.

In the first evaluation criterion (EC-1), a large number of images is retrieved for each query image. A matrix is formed using the ranks of the retrieved images. We note that the rank of the image is the order of the correct match among the retrieved images. A typical rank matrix might look like $[85\%, 10\%, 5\%, 0\%]$. This indicates that the target image appears at the first, second, and third place of the retrieved list, respectively, 85%, 10%, and 5% of the test cases.

In the second evaluation criterion (EC-2), the following parameters are calculated from the retrieved image list:

a = No. of similar images which are retrieved

b = No. of similar images which are not retrieved

c = No. of retrieved images which are not similar

d = No. of the remaining images = $N-(a+b+c)$, N is the total number of images.

Three performance measures are then calculated from the above parameters:

$$\begin{aligned} recall &= \frac{a}{a+b} \\ precision &= \frac{a}{a+c} \\ fallout &= \frac{c}{c+d} \end{aligned} \quad (3.24)$$

The parameter *recall* specifies the ratio of the similar images retrieved, the *precision* specifies the ratio of the retrieved images which are similar to the query image, and the *fallout* specifies the ratio of the retrieved non-similar images.

The third evaluation criterion (EC-3) [76] is as follows. For each image i , in a database of S images, we manually list the similar images found in the database. Let, N_i , $1 \leq i \leq S$, be the number of such images. The indexing technique is applied for a query image i . By comparison with all images (except the query image itself), we retrieve the first $(N_i + T)$ images. Here, T is a positive integer and is used as a tolerance. If n_i is the number of successfully retrieved images, the *retrieval efficiency* is defined as:

$$\eta_R = \frac{\sum_{i=1}^S n_i}{\sum_{i=1}^S N_i} \quad (3.25)$$

In EC-1 method, two indexing techniques are compared based on their rank matrix. However, comparing two matrices may not always be straightforward. In EC-2 method, three parameters have to be compared which may not provide unique result. The EC-3

method provides a unique retrieval efficiency for a given tolerance which can be easily interpreted. Hence, in this thesis we will employ EC-3 to compare the various indexing techniques.

3.8 Integrated Coding and Indexing

Compression and indexing are the two most important requirements in any image and video database. Compression techniques reduce the storage requirements of a database, whereas, indexing techniques facilitate fast retrieval of a desired image or video from a large database. The review of compression and indexing techniques, in Chapters 2 and 3, respectively, show that the two branches have progressed almost independently. Most of the compression techniques are based on waveform coding and generally employ transform domain techniques. On the other hand, most of the indexing techniques are based on object detection and generally employ spatial domain techniques. Therefore, in most cases, the coded data must be decompressed before applying the spatial domain indexing techniques. This increases the overhead and hence the complexity of the algorithm. In addition, because of the large size of the decompressed data, spatial domain techniques are relatively slow. Thus, it is desirable to integrate both coding and indexing techniques to achieve a superior performance. This issue is now receiving considerable attention from international standards groups [94].

We recall from section 2.3.2 that ISO is presently finalizing MPEG-4 standard. Although the main objective of MPEG-4 standard is to achieve very low bit-rate coding, it also emphasizes object-based indexing and manipulation of the coded video. The main features of the MPEG-4 standard, which may be useful in indexing applications are [42, 55]:

- Content-based manipulation and bit-stream editing
- Content-based multimedia data access tools
- Content-based scalability of textures, images and video
- Spatial, temporal and quality scalability

Finally, MPEG-7 group [47] has recently been formed to develop "Multimedia Content Description Interface" by the end of 2001 A.D. The main features of the MPEG-7 standard are [47]:

- Specification of a standard set to describe various types of multimedia information
- Content description to allow fast and efficient searching for audiovisual material
- Indexing of still pictures, audio, video, graphics, and 3-D models
- Information about how the above elements are combined in a multimedia presentation.

3.9. Summary

In this chapter, we have presented several image and video indexing techniques. These include pixel domain as well as compressed domain techniques. We have also presented illumination-invariant indexing techniques. A review of distance metrics and evaluation criteria has also been presented. Finally, the importance of an integrated coding and indexing scheme was pointed out with a discussion on the upcoming standards MPEG-4 and MPEG-7.

We have presented a detailed review of compression and indexing techniques in chapters 2 and 3, respectively. We now propose several novel compression and indexing techniques in chapters 4-7. We hope that the proposed techniques will have positive impact on the future research in this area.

Chapter 4

Video Compression using Wavelets

A brief review of wavelet-based video coding techniques was presented in section 2.4.3. We recall that motion estimation (ME) is crucial to achieve a superior video coding performance. However, it also increases the computational complexity of the coder significantly. It was noted that some of the ME techniques [18] have a high computational complexity, while others [135] provide a poor coding performance. In this chapter, we propose several efficient motion estimation techniques for a wavelet based video coder. The organization of this chapter is as follows. In section 4.1, we briefly outline the baseline video coder [135] employed in this thesis. The limitation of motion estimation in wavelet domain is discussed in section 4.2. An adaptive thresholding technique is presented in section 4.3 while multiresolution bi-directional motion estimation technique is proposed in section 4.4. Three adaptive motion estimation techniques are then proposed in section 4.5. The adjustable resolution selection (ARS) and adaptive bit-allocation (ABA) techniques are presented in sections 4.5.1 and 4.5.2, respectively. Bi-directional ARS/ABA technique is presented in section 4.5.3. We conclude the chapter with a summary.

4.1 Baseline Video coder

In this section, we briefly present the baseline video coder employed in this thesis. Schematic of the video encoder and decoder are shown in Figs. 4.1 and 4.2, respectively. The main modules of the video codec are i) DWT and inverse DWT, ii) motion estimation, iii) quantization and de-quantization, and iv) entropy coding and decoding.

The theory and implementation of DWT have been discussed in detail in section 2.3.5. Here, we implement the DWT using periodic extension at the borders. The wavelet filters used are all orthogonal and mostly limited to Daubechies wavelets as this family provides a good coding performance [65]. The video frames are generally decomposed to 3 stages. The multiresolution technique of Zhang *et al.* [135], discussed in section 2.4.3 is employed for motion estimation. The motion vectors, which have two components - horizontal and vertical, are then DPCM coded. We now present the details of bit-allocation and quantization scheme employed in the video coder.

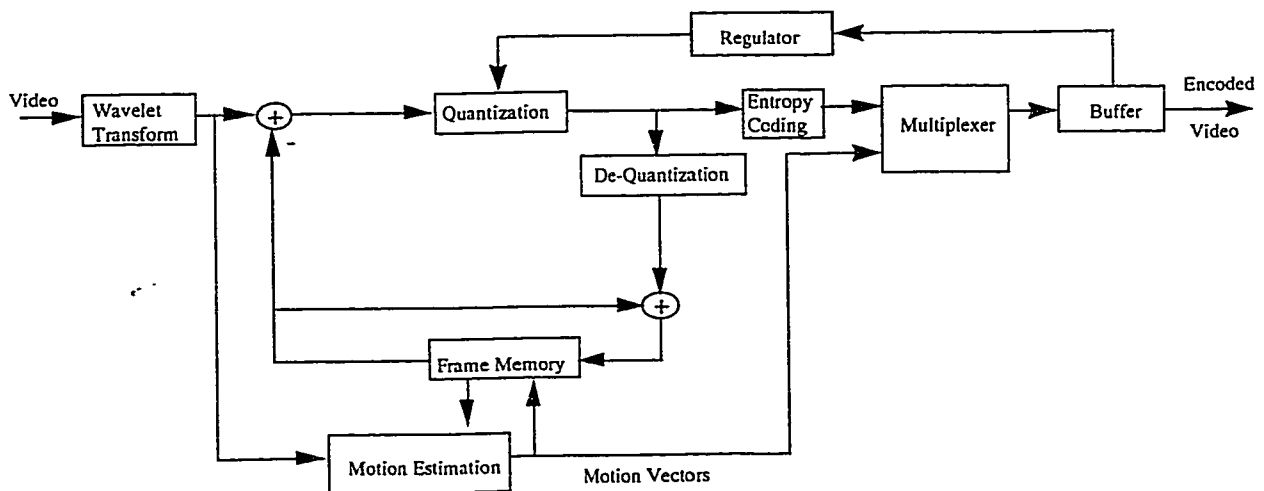


Figure 4.1: A typical wavelet-based video encoder (reproduced from Fig. 2.11)

4.1.1 Bit-allocation and Quantization

An optimal bit-allocation algorithm was discussed in section 2.4.2. It was observed that for a constant bit-rate coder employing orthogonal transforms (including orthogonal wavelets), optimal bit allocation results in uniform distribution of quantization noise over all transform coefficients [52]. However, it is well known that a variable bit-rate coder generally outperforms a constant bit-rate coder in terms of coding efficiency. Hence, a coding scheme similar to that shown in Fig. 2.10, employing uniform quantization followed by entropy coding, is used in this thesis for encoding the I-frame and the motion compensated error frames. For simplicity, one quantization step size is employed for each wavelet band. Let the quantization step-size of wavelet band k to be δ_k . Given a DWT coefficient z from band k , the corresponding quantized coefficient $\bar{z}$ is produced using the following relationship:

$$-\Pi + (\bar{z} - 1)\delta_k < z < -\Pi + \bar{z}\delta_k \quad \text{if } z < -\Pi \quad (4.1a)$$

$$\Pi + (\bar{z} - 1)\delta_k < z < \Pi + \bar{z}\delta_k \quad \text{if } z > \Pi \quad (4.1b)$$

$$\bar{z} = 0 \quad \text{if } -\Pi \leq z \leq \Pi \quad (4.1c)$$

We note that the dynamic range $[-\Pi, \Pi]$ is known as *dead zone* in the subband coding literature. The dead zone is generally larger than δ_k so that more DWT coefficients fall in this zone and results in a reduction of entropy of the quantized coefficients as well as reduction in noise present in an image.

The choice of quantization step size δ_k for the wavelet band k is crucial in order to obtain a good coding performance. In the following, we evaluate three quantization schemes - schemes 1-3, each producing different values of δ_k . The scheme providing the best

performance will then be employed in the coder. In each scheme, the lowpass wavelet band is DPCM coded for superior performance.

4.1.1.1 Scheme-1

In this scheme, the ratio of quantization step-sizes of the successive levels is chosen [65] to be 2:1, the step-sizes increase from lower resolution to higher resolution bands. In other words, the quantization step-sizes corresponding to various bands for three stage decomposition are related as follows (see Fig. 2.8 for the naming convention):

$$\delta_9 = \delta_8 = \delta_7 = 2\delta_6 = 2\delta_5 = 2\delta_4 = 4\delta_3 = 4\delta_2 = 4\delta_1 = 4\delta_0 \quad (4.2)$$

We note that the step-sizes of the bands belonging to the same scale are identical since the statistics are expected to be similar. This scheme [20, 65] is popular because of its simplicity and good coding performance. Different rate-distortion points are obtained using various absolute values of δ 's while satisfying Eq. 4.2.

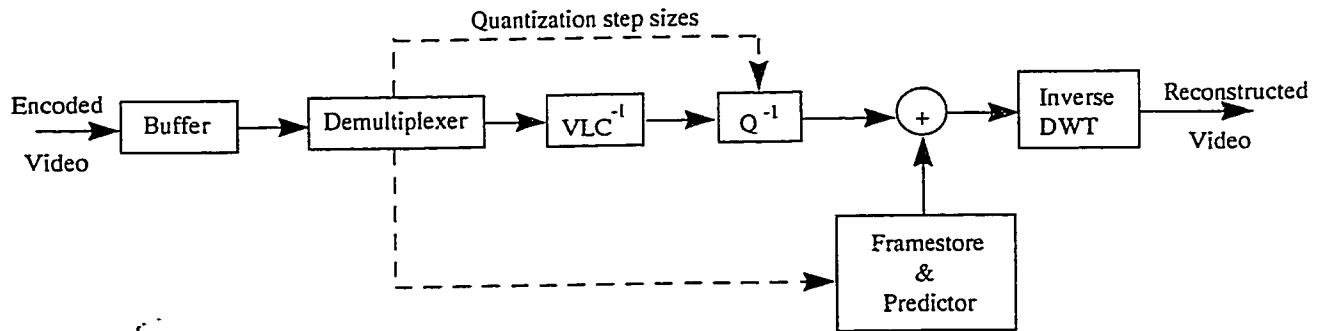


Figure 4.2: Block diagram of a typical wavelet-based video decoder. VLC^{-1} is the variable length decoder and Q^{-1} is the de-quantization module.

4.1.1.2 Scheme-2

It is assumed in Scheme-1 that the wavelet bands belonging to a scale have similar statistics and as a result same step-sizes are assigned to each of them. However, in practice, the statistics may be significantly different. In other words, each wavelet band has its own rate distortion plane depicting the distortion at a given rate. It is known that [17, 89] when wavelet bands are coded independently, the overall coding performance is optimum when the rate-distortion point of the individual bands has the same slope in their respective rate-distortion planes. Hence, in scheme-2, we employ a technique which ensures that the slopes in various rate distortion planes corresponding to the different wavelet bands are similar. This is achieved by employing a Lagrange cost function, $J = D + L_M R$ where D is the distortion, R is the rate, and L_M is the slope. For a given slope (L_M), the quantization step-size for each band is calculated from the corresponding DWT coefficients using a search technique. DWT coefficients are then quantized uniformly using these step-sizes. The overall bit-rate of a video frame is controlled by an appropriate choice of the slope (L_M).

4.1.1.3 Scheme-3

We note that scheme-2 provides a superior coding performance compared to scheme-1 exploiting the knowledge of statistics of DWT coefficients in the individual bands. However, computing the quantization step-sizes for a given slope increases the algorithm complexity significantly. In scheme-3, we reduce the complexity by modeling the probability distribution function (*pdf*) of the DWT coefficients in each wavelet band. The quantization step-size is then determined from the model parameters using a look up table for a given Lagrange cost function, L_M .

We note that the *pdfs* of highpass wavelet subimages can be modeled accurately [12] using generalized Gaussian density (GGD) function. A GGD *pdf* can be expressed as:

$$p(x) = a \exp\{-|b(x - \mu)|^\gamma\} \quad (4.3)$$

where μ , σ^2 and γ are mean, variance and shape parameter of the *pdf*, respectively. The positive constants a and b can be expressed as:

$$a = \frac{b\gamma}{2\Gamma(1/\gamma)} \quad (4.4a)$$

$$b = \frac{1}{\sigma} \sqrt{\frac{\Gamma(3/\gamma)}{\Gamma(1/\gamma)}} \quad (4.4b)$$

where $\Gamma(\cdot)$ is the standard gamma function defined by

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt \quad x > 0 \quad (4.5)$$

We note that the mean values of the highpass wavelet subimages are very close to zero, because of the lowpass nature of most natural images. Hence, the mean can be ignored while representing the DWT coefficients using GGD function. Thus, the *pdf* of a highpass subimage can essentially be described using two parameters; σ - the *standard deviation* of the coefficients, and γ - the *shape* parameter. Fig. 4.3 shows several GGD functions with different values of σ and γ .

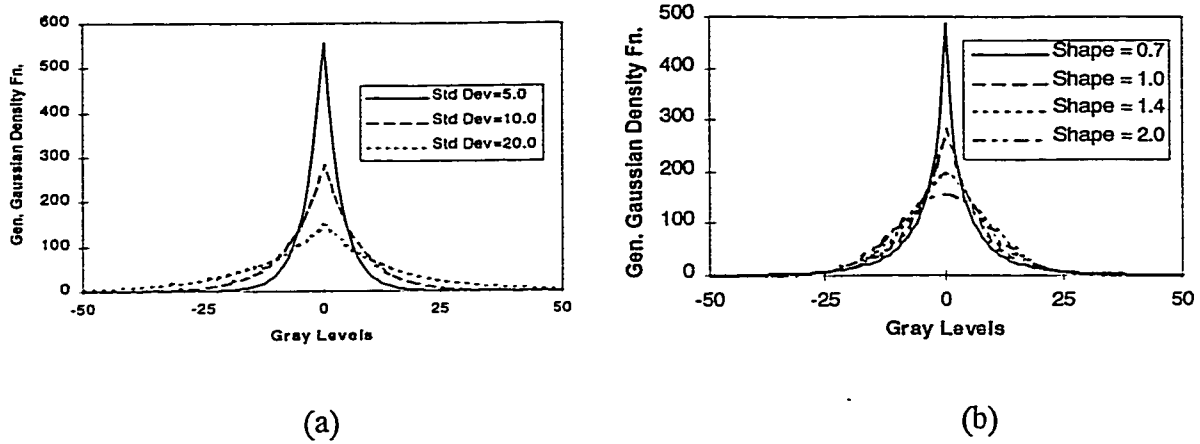


Figure 4.3. Generalized Gaussian density functions with different parameters. a) $\sigma = 5, 10$ and 20 with $\gamma = 1.0$, b) $\gamma = 0.7, 1.0, 1.4$ and 2.0 with $\sigma = 10.0$. Note that $\gamma = 1.0$ and 2.0 refer to Laplace and Gaussian density, respectively.

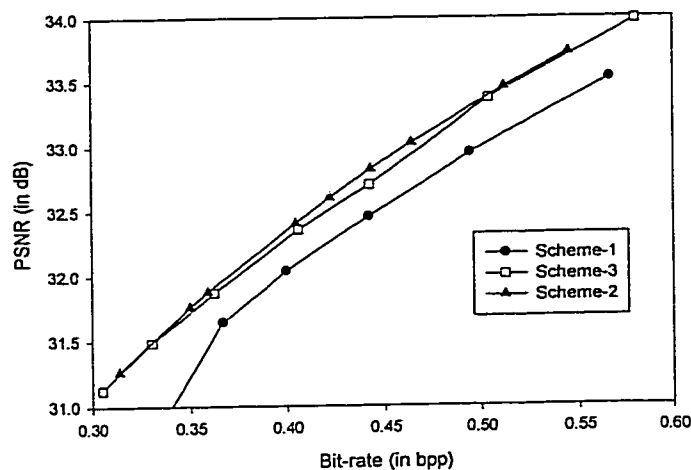
Thus, in scheme-3, the optimal quantization step-size for a given density function (*i.e.*, σ and γ) and L_M is pre-computed, and stored in a look-up table (LUT). At run-time, the GGD model parameters are first estimated for each wavelet band. The desired quantization step-size for each band, corresponding to the σ , γ and L_M , is then obtained from the LUT.

4.1.1.4 Performance Evaluation of Schemes 1-3

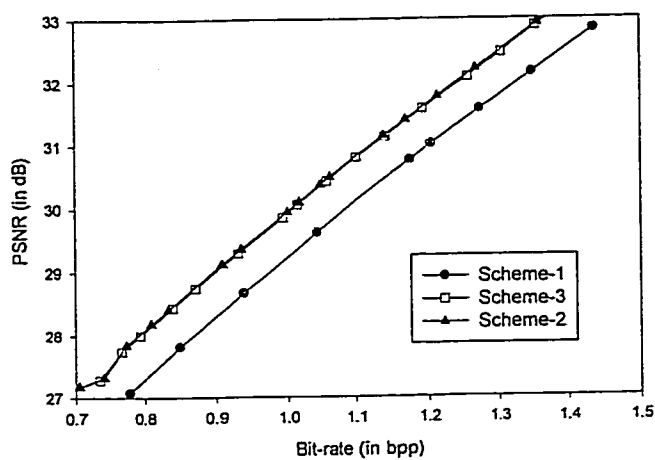
In this section, we evaluate the performance of the three schemes presented above. The images are first wavelet decomposed to three stages. The DWT coefficients of each band are quantized and encoded separately. We note that the DWT coefficients are scanned horizontally, and encoded using adaptive arithmetic coder [124].

Fig. 4.4 shows the relative coding performance of scheme 1-3 for various images, including, Lena, and one frame each from pingpong and salesman sequences. It is observed that scheme-2 and scheme-3 provide similar performance for all images. This proves that the modeling of wavelet-band *pdf* with GGD function is reasonably accurate. It is also seen that

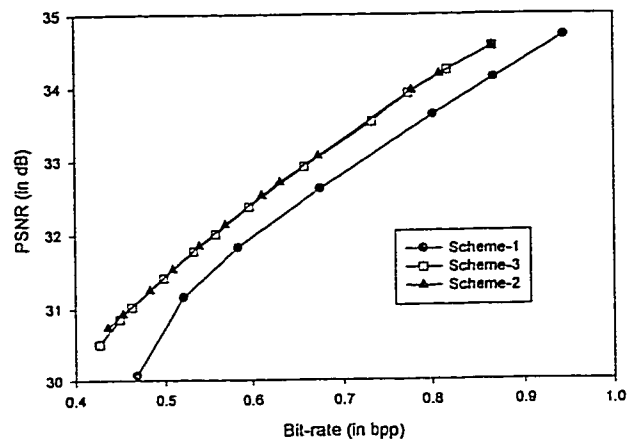
scheme-3 provides an improvement (approximate 0.5-1 dB) over scheme-1 which is expected. Hence, in this thesis, we employ scheme-3 in all subsequent analysis.



(a)



(b) Pingpong



(c) Salesman

Figure 4.4. Coding performance of three quantization schemes for various images. a) Lena, b) first frame of Pingpong sequence, c) first frame of Salesman sequence.

4.2 Translation Variance of DWT Coefficients

We recall from section 2.4.1 that block-based methods provide good motion compensation performance in spatial domain. This is mainly because the underlying assumption of translatory motion of objects or camera is valid in most cases. In an ideal situation when the objects follow a translatory motion model, the blocks in a video frame can be predicted accurately from the previous frame using the corresponding block and a motion vector (which has horizontal and vertical components). However, when the frames are decomposed using wavelet/subbands, the block-based motion compensation does not provide a good motion compensation. This is mainly because the DWT coefficients are not translation invariant, *i.e.*, if a signal is shifted by a few spatial units, the DWT coefficients are not shifted by the corresponding amount in the wavelet domain.

In order to demonstrate the translation variance of DWT, we consider a signal $x[n]$ passing through a filter $\tilde{H}(z)$ and subsequently decimated (as shown in Fig. 2.6). Then the decimated output, $v_0[n]$ can be expressed as:

$$\begin{aligned} V_0(z) \Big|_{x[n]} &= \frac{1}{2} [X(z^{1/2}) + X(-z^{1/2})] \\ &= \frac{1}{2} [X(z^{1/2})\tilde{H}(z^{1/2}) + X(-z^{1/2})\tilde{H}(-z^{1/2})] \end{aligned} \quad (4.6)$$

Now, if we assume that a signal $p[n] = x[n - k]$ is passed through the same filter. The corresponding decimated output will be

$$\begin{aligned} V_0(z) \Big|_{p[n]} &= \frac{1}{2} [P(z^{1/2})\tilde{H}(z^{1/2}) + P(-z^{1/2})\tilde{H}(-z^{1/2})] \Big|_{P(z) = z^{-k}X(z)} \\ &= \frac{1}{2} [z^{-k/2}X(z^{1/2})\tilde{H}(z^{1/2}) + (-1)^k z^{-k/2}X(-z^{1/2})\tilde{H}(-z^{1/2})] \end{aligned} \quad (4.7)$$

The z -transform of translated (we have used a translation of $k/2$ because of 2:1 decimation) filter output can be expressed as :

$$\begin{aligned} Z\{v_0[n-k/2]_{x[n]}\} &= z^{-k/2}V(z)|_{x[n]} \\ &= \frac{1}{2}\left[z^{-k/2}X(z^{1/2})\tilde{H}(z^{1/2}) + z^{-k/2}X(-z^{1/2})\tilde{H}(-z^{1/2})\right] \end{aligned} \quad (4.8)$$

Eq. 4.7 shows that the decimated filtered output has two components. The first component is the result of the filtering, while the second component (which is an aliased form of the first component) is due to decimation. Comparison of Eqs. 4.7 and 4.8 reveals that the DWT coefficients (with one stage decomposition) will be translation invariant if and only if $(-1)^k = 1$, i.e., k is even. For odd k , the aliased component will not be translation invariant. Generally, for a m -stage DWT decomposition, the DWT coefficients will be translation invariant only when k is a multiple of 2^m . This is clearly a very restrictive assumption. As a result, the motion of an object in the spatial domain generally does not correspond to the translation of coefficients in the wavelet domain.

The effect of aliasing on motion estimation is shown below using an example. Given two sequences $x(n)$ and $p(n) = x(n-k)$, we would like to predict $p(n)$ from $x(n)$ in the wavelet domain. It is to be noted that for even values of k , the prediction is perfect whereas for any other values of k (odd or fractional values), there will be aliasing and hence a prediction error. A test sequence $x(n)$ of 32 samples (selected arbitrarily from *Lena* image) is shown in Fig. 4.5a. This is a typical sequence from images where neighboring pixels are highly correlated. We would like to predict $p(n) = x(n-1)$ (i.e., k has been chosen to be 1) from $x(n)$ in the wavelet domain. The lowpass and highpass filter outputs using Daubechies 8 tap

wavelet, $x_0(n)$ and $x_1(n)$ (see Fig. 2.6) are shown in Fig. 4.5b. The prediction error in the lowpass subband can be found as follows:

1. Generate $v_0(n)$ by decimating $x_0(n)$, *i.e.* $v_0(n) = x_0(2n)$.
2. Generate $v_0(n-0.5)$ by interpolating $v_0(n)$.
3. Generate $w_0(n)$ by choosing the odd samples of $x_0[n]$, *i.e.* $w_0(n) = x_0(2n-1)$.
4. The difference between $w_0(n)$ and $v_0(n-0.5)$ is the prediction error (assuming the prediction coefficient is 1.0).

Plots of $v_0(n)$, $v_0(n-0.5)$ and $w_0(n)$ are shown in Fig. 4.5c. It is observed that $v_0(n-0.5)$ and $w_0(n)$ are reasonably similar. The prediction gain which is defined as the ratio of signal energy and the motion predicted error energy (see Eq. 2.9), was found to be 1250. Thus, we observe that motion compensation works well in the lowpass subband.

The prediction error in the highpass subband can be calculated in a similar manner by comparing the outputs of the highpass filter, *i.e.*, by comparing $w_1(n)$ and $v_1(n-0.5)$. The plots of $v_1(n)$, $v_1(n-0.5)$ and $w_1(n)$ are shown in Fig. 4.5d. It is observed that $w_1(n)$ is difficult to predict from $v_1(n-0.5)$. The prediction gain in this case was found to be 0.86. We note that when the prediction gain is less than 1, it is beneficial to encode the coefficients, rather than the prediction error. Hence, in this particular case, there is no gain in predicting the highpass coefficients. However, in general, we expect some gain (greater than unity) from the prediction.

From the above analysis, it is observed that the estimation in the lowpass band provides a good performance when the object motion is translatory in nature. However, the motion compensation does not work well in the case of high frequency coefficients. This is expected

because $x_0(2n-1)$ can be well approximated by interpolating $x_0(2n)$ and $x_0(2n-2)$ due to the lowpass nature of $x_0(n)$. However, since $x_1(n)$ contains substantial high frequency energy, $x_1(2n-1)$ cannot be well approximated by interpolating $x_1(2n)$ and $x_1(2n-2)$.

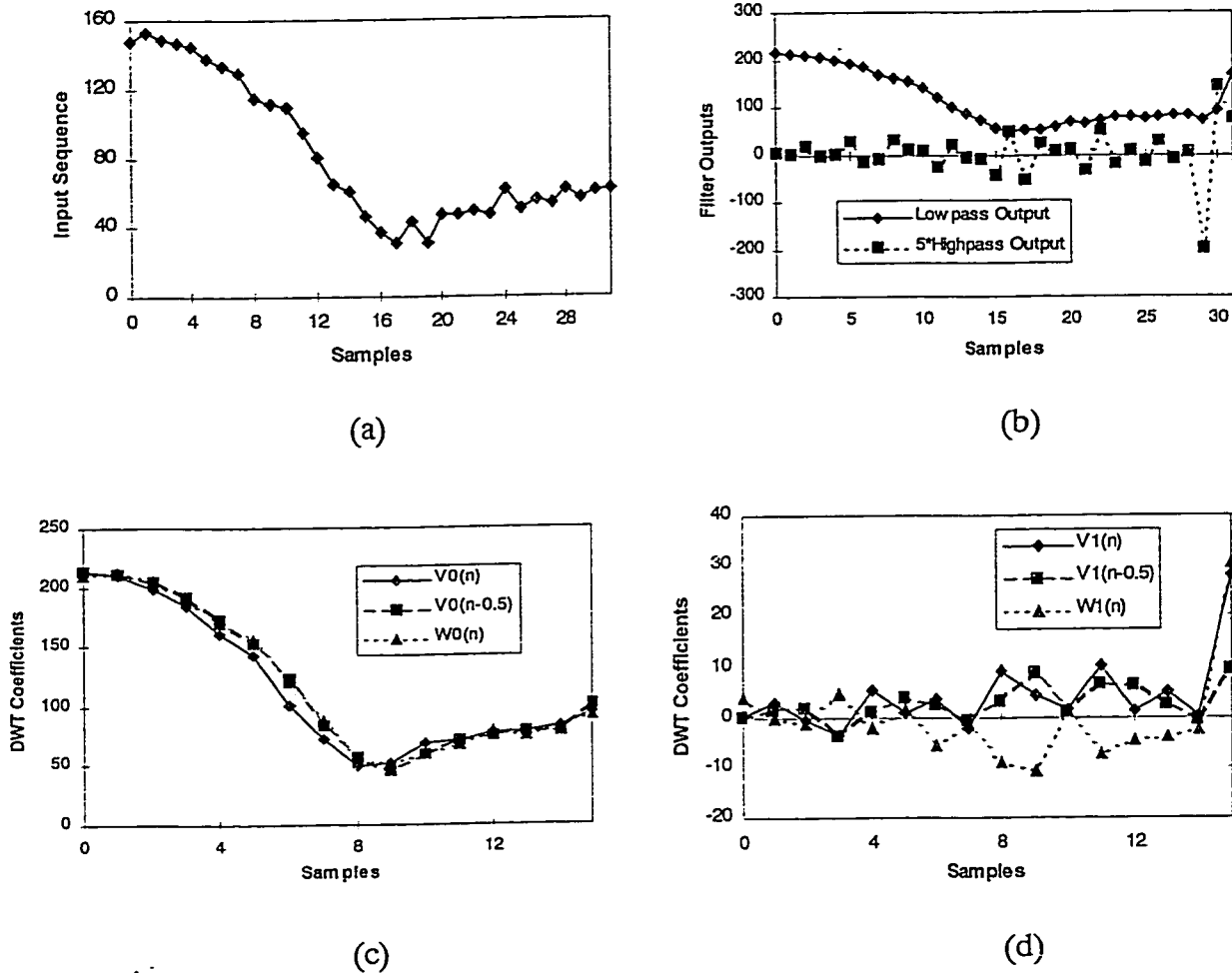


Figure 4.5: Effect of aliasing on motion estimation in the lowpass and highpass wavelet bands. (a) An arbitrary input sequence $x(n)$, (b) The undecimated output of lowpass and highpass filter, (c) Lowpass DWT Coeffs. $v_0(n)$: coeffs. available, $v_0(n-0.5)$: predicted coeffs., $w_0(n)$: interpolated coeffs, (d) Highpass DWT Coeffs. - $v_1(n)$: coeffs. available, $v_1(n-0.5)$: predicted coeffs, $w_1(n)$: interpolated coeffs.

Thus, we observe that the motion information, which is local in the spatial domain, is spread (due to aliasing) across all the subbands in the wavelet domain. Therefore, to achieve better motion compensation, the subbands cannot be treated independent of one another. Cafforio *et al.* [13] proposed a technique for estimating motion by jointly considering all the subbands. To ensure scalability, the motion vectors for a subband are estimated by using the same or lower resolution subbands. Although this technique provides better motion estimation, the complexity is very high compared to the MRME-Z [135] or MRME-U [117] techniques described in section 2.4.3.

We now propose two motion estimation techniques within the framework of multiresolution motion estimation. These techniques treat each subband independently, and provide a superior performance compared to the MRME-Z technique [135] at a similar level of complexity.

4.3 Proposed Adaptive Thresholding (AMRME) Technique

In the MRME-Z technique [135], the motion vectors corresponding to all the subimages are calculated. However, we recall from section 4.2 that the high frequency subimages of consecutive frames are not well correlated even though the consecutive frames are very similar in the spatial domain. The aliasing problem due to decimation increases with the number of decomposition levels, and the performance of ME algorithm is expected to degrade at lower resolution subbands. Although, the highpass subimages cannot be predicted well, the MRME-Z technique performs well in practice as most of the information (or energy) is contained in the lowpass subimages which can be motion compensated well. For

highpass subimages, it would be helpful to determine on a block-by block basis if motion estimation is beneficial for predicting the highpass subband coefficients.

We propose to use an adaptive threshold (AMRME) to decide if the motion vector should be transmitted to the receiver [67]. Here, each reference block is matched to a particular block (which provides the best match) in the previous frame. If the dissimilarity between the reference block and the best-match block is greater than a threshold, the block is discarded and the motion vector corresponding to that block is not coded. A zero block is initialized in the corresponding location at the decoder. The proposed thresholding improves the coding performance in two ways. First, the number of motion vectors to be coded will be less, resulting in a reduced bit-rate. Secondly, the objective quality of the reconstructed image will improve since the mismatched blocks are discarded.

In summary, the motion vectors are predicted and refined in a hierarchical manner using the following equations (also see section 2.4.3).

$$V_4^o(x, y) = 2V_8^o(x, y) + \Delta_4^o(x, y) \quad (4.9)$$

$$V_2^o(x, y) = V_4^o(x, y) + 2V_8^o(x, y) + \Delta_2^o(x, y) \quad (4.10)$$

where $V_i^o(x, y)$ represents the motion vector of the reference block for o -orientation ($o \in \{H, V, D\}$) subband at i th scale. However, the DFD energy of a block will determine if the corresponding (coarse/incremental) motion vectors $V_8^o(\cdot)$, $\Delta_4^o(\cdot)$ and $\Delta_2^o(\cdot)$ will be encoded or discarded.

The choice of threshold is crucial in achieving a good performance. Since, we have assumed *PSNR* as the measure of quality, the adaptive threshold should be related to the energy of the reference block. If the threshold is made equal to the energy of the reference

block, the image frames will be reconstructed with lowest distortion in L^2 metric. However, we note that the rate-distortion function may not be optimal since the entropy of the motion vectors may be high. Generally, a threshold that is smaller than the energy of the block, provides a better performance. In AMRME, we employ an adaptive energy threshold T_B , for a block B , according to the following relation:

$$T_B = \tau * E_B \quad (4.11)$$

where E_B is the energy of the reference block and τ is a constant (henceforth referred to as threshold factor) having a dynamic range [0,1]. If the energy of the motion compensated block is less than T_B , motion compensation is considered to be successful. An appropriate τ is chosen for a given sequence to obtain a superior coding performance. It should be noted that the complexity of AMRME is marginally higher compared to MRME-Z [135].

4.3.1 Coding Performance

We now evaluate the performance of the AMRME technique using three test sequences - i) Pingpong, ii) Salesman, and iii) Miss America (see Appendix B for details of these sequences). The basic video coder as shown in Fig. 4.1 has been employed in the simulations. The full search algorithm with *mean square error* (MSE) as the matching criterion has been used to obtain the motion vectors. The block sizes of the level-1, level-2 and level-3 wavelet subimages have been chosen to be 3×3 , 6×6 and 12×12 , respectively. The maximum allowed displacement (MAXAD) for level-3 (S_8 and W_8 's) subimages is 4 pixels and the maximum allowed refinements (MAXAR) in level-2 (W_4 's) and level-1 (W_2 's) subimages are 2 and 1 pixel, respectively. The *group of pictures* (see Fig. 2.4) has been

taken as IPPPPPI (*i.e.*, five P-frames between consecutive I- frames). The bit-rate and PSNR have been calculated by averaging the bit-rate and MSE for all (I- and P-) frames.

The video frames are decomposed with Daubechies-8 tap wavelet. The DWT coefficients as well as the motion compensated error coefficients have been quantized using scheme-3 described in section 4.1.1.3. Various rate-distortion points have been obtained with different Lagrange cost function L_M 's.

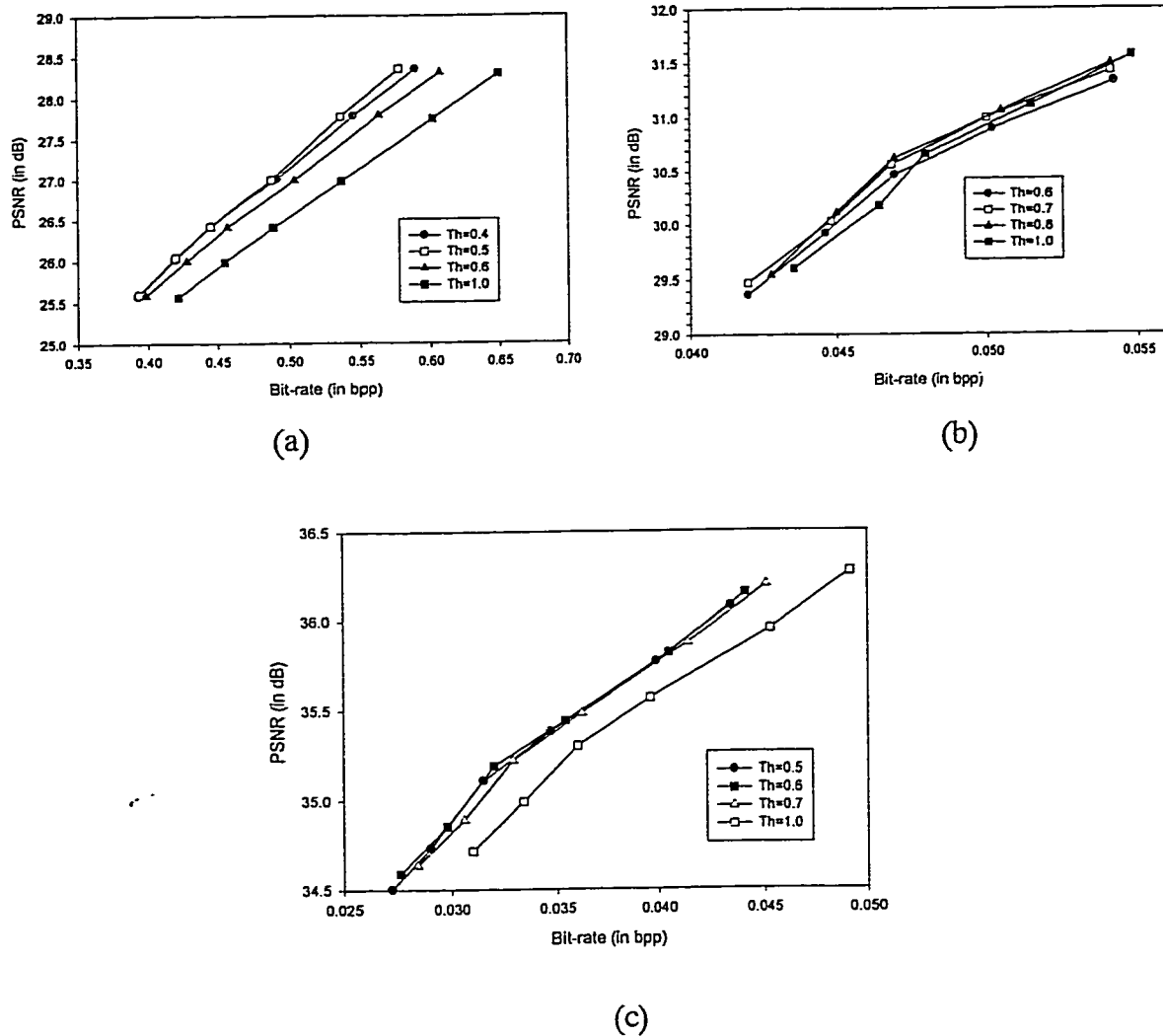


Figure 4.6: Coding performance of AMRME techniques for different thresholds (Th). (a) Pingpong, (b) Salesman, and (c) Miss America sequence. "Th" refers to the threshold factors τ .

The coding performance of AMRME depends on the choice of the threshold, and hence also on the choice of τ 's. Fig. 4.6 compares the coding performance for various τ 's. It is observed that the best τ is in the range 0.5-0.8 depending on the sequence. For the pingpong sequence, τ of 0.5 provides an improvement of about 1 dB over τ of 1.0. For salesman sequence, although a τ of 0.7 provides the best performance, the influence of τ is not as significant. For a Miss America sequence, a τ of 0.6 provides the best coding performance. Miss America and Salesman, being low entropy sequences, require fewer bits for encoding. Therefore, bit-saving due to thresholding is able to improve the overall coding performance. The improvement of coding performance for Pingpong sequence with a lower threshold can be attributed to two reasons. Firstly, a lower τ reduces the entropy of the motion vectors. Secondly, a lower τ provides more robust motion estimation at higher levels of the wavelet pyramid. Since Pingpong is a high motion sequence, robust motion at the top level of the pyramid is important and results in an improved performance.

The coding performance of AMRME and MRME-Z are compared in Fig. 4.7. It is observed that AMRME technique provides a significant improvement over MRME-Z. The improvement is more significant for the Pingpong sequence, since the highpass subimages of the consecutive frames are less correlated due to its fast motion.

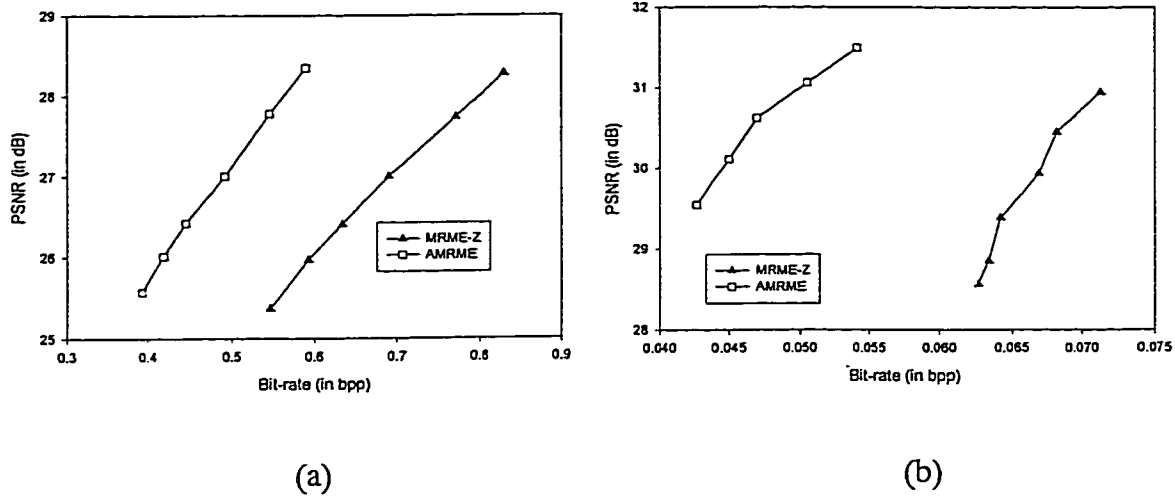


Figure 4.7: Performance comparison of MRME-Z and AMRME Techniques. (a) Pingpong (with $\tau=0.5$), (b) Salesman (with $\tau=0.7$).

4.4 Bi-directional Motion Estimation (BMRME) Techniques

We recall from section 2.3.2 that MPEG coder [27, 45] divides video frames into three categories: I-, P-, and B- frames, to achieve a good coding performance. I- frames are coded independent of others. The P- frames are motion predicted from the previous reference (which may be I- or P-) frame and the associated motion vectors, along with the prediction error frame, are sent to the decoder for reconstruction. B- frames are compared with both previous, and next reference frames, and generally three types of predictions are used: forward prediction, backward prediction and bi-directional prediction [27]. After motion estimation is performed, the motion vectors corresponding to the B- frames are sent to the decoder.

Although, bi-directional motion estimation improves the coding performance, it also increases the complexity of the coder. First, extra memory is required in the decoder to store the future reference frame. Secondly, a delay of a few frames is inevitable since B- frame

coding can be performed only after the future reference frame is available. Finally, the computational complexity of ME is significantly higher than (approximately twice) that of unidirectional ME.

In this section, we present two bi-directional motion estimation techniques in the wavelet domain. The first technique is a straightforward adaptation of the bi-directional ME as performed in MPEG. The second technique exploits the multiresolution property of wavelet transform, and provides a superior performance. We now present the details of these two techniques.

4.4.1 Technique-1 (BMRME-1)

We recall from the previous section that the motion vectors, in AMRME technique, corresponding to any level depend on the motion vectors corresponding to all higher levels (see Eq. 4.9 and 4.10). The AMRME technique employs two wavelet pyramids, one corresponds to the previous reference frame, while the other corresponds to the current frame. We note that there are three wavelet pyramids in bi-directional ME corresponding to the two reference frames and the current frame.

In the proposed bi-directional motion estimation technique (BMRME-1), a temporal flag (TFLAG) having three states: 0, 1 and 2, is employed to denote whether the best match is from the previous (state 0) or from the future reference frame (state 1), or by interpolating the two frames (state 2). The details of the proposed technique for three level decomposition is summarized below.

1. For each block position (x, y) , the motion vectors are calculated for the subimages (*i.e.* s_8 , w_8^H , w_8^V and w_8^D) of the highest level pyramid with respect to both the

previous and the next frame. Let the motion vectors corresponding previous and next reference frames be denoted by ${}^pV_8^o(.)$, and ${}^nV_8^o(.)$ (where $o \in \{H, V, D\}$), respectively. Depending on the DFD energy of a motion block, a TFLAG is set to 0 (forward prediction), 1 (backward prediction), or 2 (bi-directional prediction).

2. For each block position (x, y) , the motion vectors are calculated for subimages corresponding to the other levels of wavelet pyramid. The motion vectors corresponding to subimages corresponding to a specific direction (horizontal, vertical, or diagonal) are estimated using the following equation.

$$\begin{aligned}
 {}^pV_4^o(x, y) &= 2 \cdot {}^pV_8^o(x, y) + {}^p\Delta_4^o(x, y) \\
 {}^pV_2^o(x, y) &= 2 \cdot {}^pV_8^o(x, y) + {}^pV_4^o(x, y) + {}^p\Delta_2^o(x, y) \\
 {}^nV_4^o(x, y) &= 2 \cdot {}^nV_8^o(x, y) + {}^n\Delta_4^o(x, y) \\
 {}^nV_2^o(x, y) &= 2 \cdot {}^nV_8^o(x, y) + {}^nV_4^o(x, y) + {}^n\Delta_2^o(x, y)
 \end{aligned} \tag{4.12}$$

Again, TFLAG corresponding to a block is set to 0, 1, or 2 depending on the DFD energy of the block.

A schematic of the proposed technique is shown in Fig. 4.8. We note that in the above scheme, all coarse and incremental motion vectors (*i.e.*, ${}^pV_8^o, {}^nV_8^o, {}^p\Delta_4^o, {}^n\Delta_4^o$) corresponding to subimages of level-3 and level-2 are required for decoding. This is due to the fact that motion vectors at level-1 depend on the values of these higher-level motion vectors. However, for level-1 subimages, either ${}^p\Delta_2^o$, or ${}^n\Delta_2^o$, or both are required depending on the TFLAG of a block. The complexity and coding performance of BMRME-1 technique will be discussed in section 4.4.3.

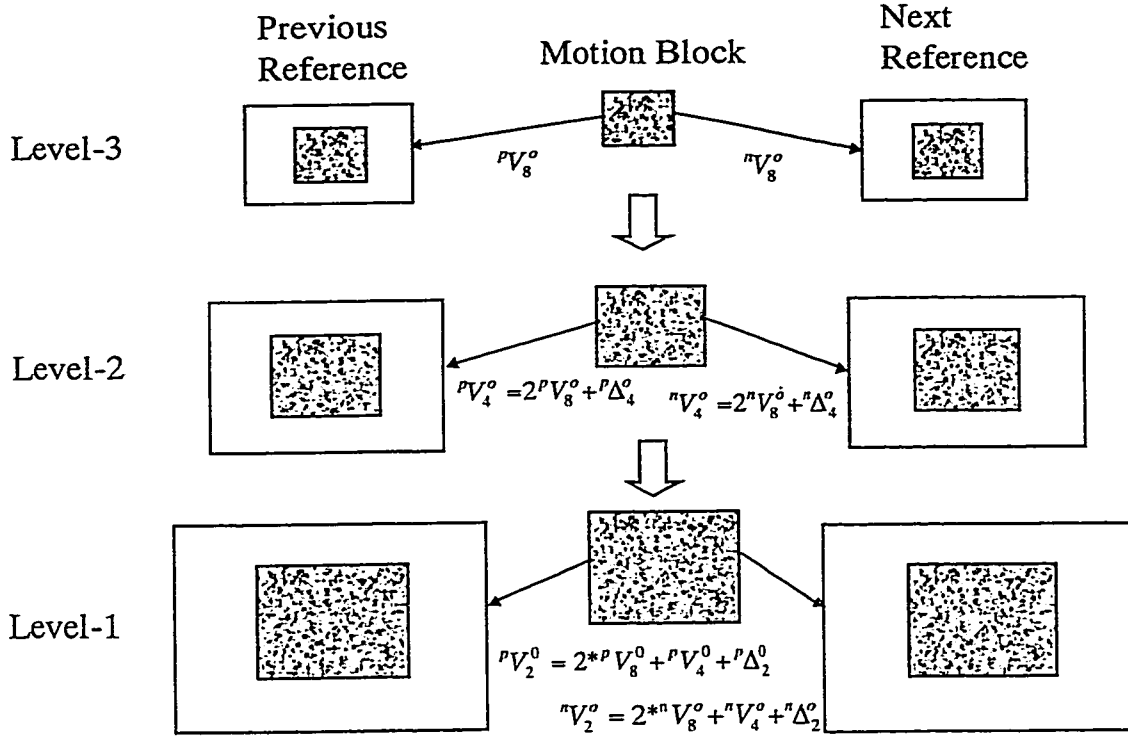


Figure 4.8. Schematic of BMRME-1 technique. $P V_8$, $P V_4$, $P V_2$ are motion vectors corresponding to previous reference frame; $N V_8$, $N V_4$, $N V_2$ are motion vectors corresponding to next reference frame. In each level, $P V$ or $N V$, or both are employed for motion compensation.

4.4.2 Technique-2 (BMRME-2)

We recall from section 2.3.3 that motion blocks in MPEG coding scheme do not overlap and hence each block may vary differently over time. As a result, bi-directional motion estimation in MPEG coder is performed for all blocks in a B- frame. The BMRME-1 technique presented in section 4.4.1 estimates bi-directional motion vectors for all blocks in a wavelet-decomposed frame. In addition, it encodes most of the coarse and fine motion vectors corresponding to previous and next reference frames. Although, the complexity is significantly smaller than MPEG bi-directional ME, it does not exploit the full potential of wavelet pyramid.

We note that an image-block (in pixel domain) is represented by DWT-blocks at different levels of pyramid. Consider that an image-block b has a TFLAG of k ($k \in \{0,1,2\}$) depending on the bi-directional ME. It is highly probable that all DWT blocks corresponding to b will have the same TFLAG k . Therefore the computational complexity can be reduced by executing the bi-directional ME only at the highest level of the pyramid. The motion estimation at all other levels can essentially be uni-directional using TFLAGs of blocks at the highest level. In addition, it has been observed that the interpolation of highpass DWT coefficients from previous and next reference frames does not provide good motion compensation. Hence, the interpolation mode is ignored in BMRME-2 technique. In addition to the reduction of computational complexity, BMRME-2 technique needs to encode fewer motion vectors, resulting in a reduction in bit-rate. The procedure can be summarized as follows:

1. For each block position (x, y) , the motion vectors are calculated for the subimages (*i.e.* s_8 , w_8^H , w_8^V and w_8^D) of the highest level pyramid with respect to both the previous and the next frame. Let the motion vectors corresponding previous and next reference frames be denoted by ${}^pV_8^o(\cdot)$, and ${}^nV_8^o(\cdot)$. Depending on the DFD energy, the corresponding TFLAG is set to 0 (forward prediction), or 1 (backward prediction). Set $V_8^o(x, y) = {}^pV_8^o$ if TFLAG=0, otherwise $V_8^o(x, y) = {}^nV_8^o$. If neither forward nor backward prediction performs well, the block is discarded (as in AMRME).
2. For subimages corresponding to lower levels of the pyramid, TFLAG is not calculated. The motion vectors corresponding to subimages and corresponding to a

specific direction (horizontal, vertical, or diagonal) are estimated using the following equation.

If $TFLAG_8^o(x, y) = 0$

$$\begin{aligned} V_4^o(x, y) &= 2^{*p} V_8^o(x, y) + {}^p\Delta_4^o(x, y) \\ V_2^o(x, y) &= 2^{*p} V_8^o(x, y) + {}^pV_4^o(x, y) + {}^p\Delta_2^o(x, y) \end{aligned} \quad (4.13)$$

If $TFLAG_8^o(x, y) = 1$

$$\begin{aligned} V_4^o(x, y) &= 2^{*n} V_8^o(x, y) + {}^n\Delta_4^o(x, y) \\ V_2^o(x, y) &= 2^{*n} V_8^o(x, y) + {}^nV_4^o(x, y) + {}^n\Delta_2^o(x, y) \end{aligned} \quad (4.14)$$

A schematic of BMRME-2 technique is shown in Fig. 4.9.

4.4.3 Computational Complexity of BMRME Techniques

Here, we estimate the computational complexity of BMRME-1 and BMRME-2 techniques. We note that the complexity of the bi-directional motion estimation is twice that of uni-directional ME (as in case of AMRME) in case of BMRME-1 techniques. In BMRME-2 technique, the bi-directional ME is executed only in case of first four subimages. In other words, the complexity of ME in first four subimages is twice that corresponding to the complexity of AMRME. For all other subimages, the complexity is identical to that of AMRME. The overall complexity of AMRME, BMRME-1, and BMRME-2 techniques are shown in Table-4.1. The complexity is given in operations/pixel (an *operation* includes one subtraction, one multiplication and one addition for MSE criteria). It is observed that the complexity of all three techniques is significantly smaller than full search ME in pixel

domain. Among MRME techniques, the complexity of BMRME-2 is marginally higher than that of AMRME, but substantially lower than BMRME-1.

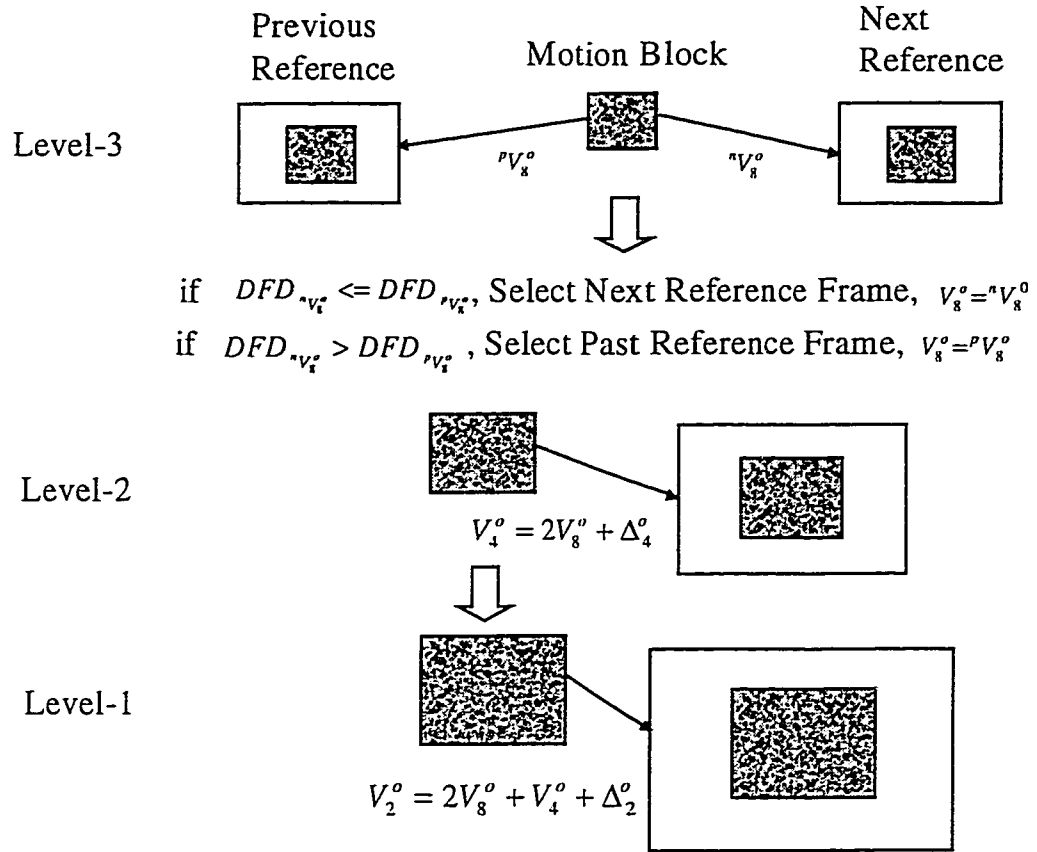


Figure 4.9. Schematic of BMRME-2 technique. ${}^pV_8$ and ${}^nV_8$ are motion vectors corresponding to highest pyramid level of previous and next reference frame. In Level-2 and Level-1, either previous or next reference frame is employed depending on their motion compensation performance at Level-3.

To compare the bit-rate of MRME and BMRME, we calculate the number of blocks in each subimage as

$$N_{BIS} = \left(\frac{N_R}{2^L m} \right) \left(\frac{N_C}{2^L n} \right) \quad (4.15)$$

where N_R and N_C are, respectively, the number of rows and columns of an image frame and $m \times n$ is the block-size at the highest level of the pyramid. The number of motion vectors

(each vector having two components: horizontal and vertical) and TFLAGs required to be transmitted are shown in Table 4.2. The dynamic range of the motion vectors (each component) and TFLAGs are shown in column 2. We observe that the number of motion vectors in MRME and BMRME-2 is identical. However, BMRME-2 has a small overhead for transmitting the TFLAGs (for the first four subimages). On the other hand, the number of motion vectors in BMRME-1 is almost twice that of AMRME/BMRME-2 techniques. In addition, the number of TFLAGs is also higher than BMRME-2 technique.

Table 4.1

Computational complexity (in operations/pixel) of motion estimation algorithms. p is the equivalent search area in level-0.

	Full Search		Multiresolution ME		
	ME	B-ME	MRME	BMRME-1	BMRME-2
Complexity	$\approx 4p^2$	$\approx 8p^2$	$\approx 0.6p^2 + 1.5p$	$\approx 1.2p^2 + 3.0p$	$\approx 0.9p^2 + 1.5p$
Typical Value	1024 ⁺	2048 ⁺	15.6 ⁺⁺	31.2 ⁺⁺	20.4 ⁺⁺

⁺ $p=16$, ⁺⁺ $p=4$

Table 4.2

Number of motion vectors and TFLAGs in AMRME/BMRME algorithms (N_{BIS} = number of blocks in a subimage)

Level	Dynamic range of vectors	AMRME	BMRME-1	BMRME-2
3	[-4, 4]	$4 * N_{BIS}$	$8 * N_{BIS}$	$4 * N_{BIS}$
2	[-2, 2]	$3 * N_{BIS}$	$6 * N_{BIS}$	$3 * N_{BIS}$
1	[-1, 1]	$3 * N_{BIS}$	$3 * N_{BIS}$	$3 * N_{BIS}$
Time	BMRME-1: [0-2] BMRME-2: [0-1]	NA	$10 * N_{BIS}$	$4 * N_{BIS}$

4.4.4 Performance of BMRME Techniques

In this section, we evaluate the coding performance of BMRME techniques. We note that the coding performance of a video coder significantly depends on the composition of I-, P-, and B- frames. In order to make a fair comparison, we compare the coding performance of bi-directional ME with that of uni-directional ME. BMRME employs a GOP (group of pictures) as IBBBBPBBBBPBBBBI while the AMRME technique employs a GOP as IUUUUPUUUUPUUUI. The motion estimation of U- frames is done with respect to the past reference frame (*i.e.*, I- or P- frames). A threshold factor (τ) of 0.6 is used for all sequences.

We note that for a fast moving sequence, the motion vectors corresponding to only the B- frames may not always provide a good subjective quality of the reconstructed video frames. Hence, we have implemented an adaptive wavelet-based video coder which sends the error frames corresponding to B- frames when it is necessary, *i.e.* when the difference between the reconstructed I and B frames exceeds a certain threshold.

Fig. 4.10 compares the coding performance of AMRME and BMRME techniques. The coding performance has been obtained by averaging the performance of P-, B-, and U- frames, *i.e.*, I-frames have not been considered since they provide identical coding performance in both cases. We observe that BMRME-1 and BMRME-2 techniques provide a similar performance. It has been found that the gain in BMRME-1 due to better motion estimation is offset by the increase in the bit-rate due to large number motion vectors, resulting in similar performance. However, both BMRME-1 and BMRME-2 techniques provide a superior coding performance (an improvement of over 0.5 dB) compared to

AMRME technique. For the Pingpong sequence, a performance improvement of 0.5 dB has been observed.

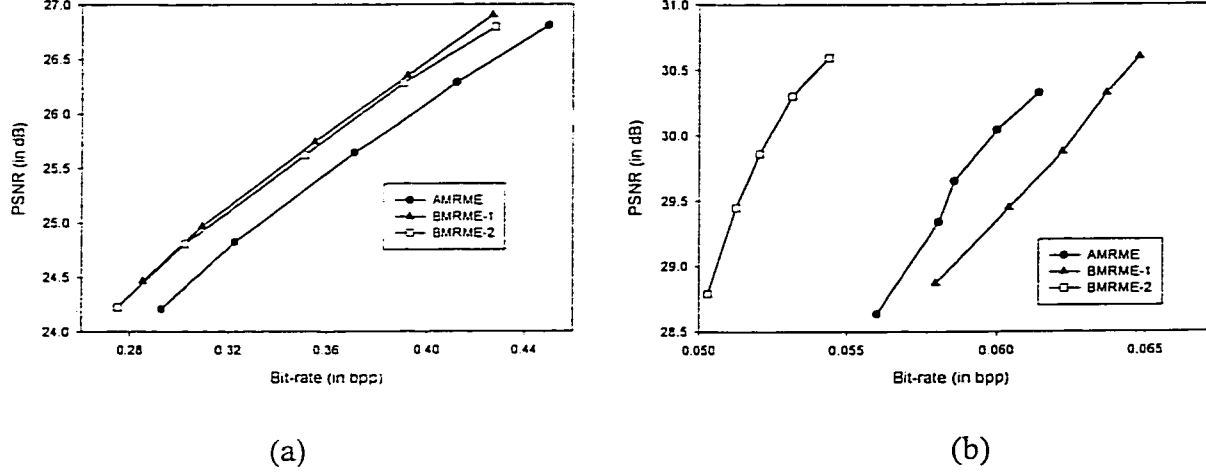


Figure 4.10: Performance comparison of AMRME and BMRME techniques. (a) Pingpong sequence. (b) Salesman sequence. A threshold factor (τ) of 0.6 has been used in each case.

It can be seen from Fig. 4.10b that BMRME-2 provides a superior performance compared to AMRME technique. We note that the entropy of the *displaced frame difference* (DFD) is very small in the case of the salesman sequence. Hence, even a small improvement in motion compensation may have a large effect on the overall coding efficiency. In addition, BMRME-2 addresses the uncovered areas better, since an uncovered area can be predicted only from the future reference frame. Hence, BMRME-2 technique has the potential to provide a superior performance when there is a scene change in a video sequence.

However, the BMRME-1 technique has failed to provide a good coding performance in the case of the salesman sequence. We note that BMRME-1 technique employs a large number of motion vectors and TFLAGs, which consume a large portion of the bit-budget. Since salesman is a slow moving sequence, it is not beneficial to employ a large number of motion

vectors. BMRME-2 technique employs a smaller number of motion vectors and hence results in a superior performance.

It can be concluded from the above experimental results that BMRME-2 technique provides a superior coding performance. In addition, it also has a lower computational complexity compared to BMRME-1. Hence, the BMRME-2 (henceforth referred to as BMRME) technique will be employed in video coders developed in later sections.

4.5 Adaptive Motion Estimation Techniques

The multiresolution motion estimation techniques (AMRME, BMRME) presented in sections 4.3, and 4.4 are based on hierarchical coarse-to-fine approach. In these techniques, motion vectors are first estimated at the highest level of the pyramid. The motion vectors at lower levels are predicted from these coarse motion vectors, and subsequently refined using a coarse-to-fine approach. In MRME framework, the number of motion blocks in all subimages is identical. As a result, there is a one-to-one correspondence between the motion blocks at various levels of the wavelet pyramid. Hence, during motion estimation, the block sizes are quadrupled and the search areas are reduced in each successive higher resolution subbands. Although, these techniques have lower complexity, motion estimation may be inaccurate due to three main factors. First, the small block-size employed for ME at a higher level does not provide robust motion estimation. Increasing the block-size may, on the other hand, increase the residual error energy resulting in a degradation of coding performance. Secondly, the amount of aliasing energy increases with the number of decomposition levels resulting in poor motion compensation (see section 4.2). Finally, when ME is performed

using a reconstructed reference frame, the quantization noise present in the reconstructed frame degrades the quality of the ME process.

In order to reduce the degradation caused by the above factors, Conklin *et al.* [18] have proposed MRME techniques (MRME-C) based on a fine-to-coarse approach. Here, the initial coarse motion estimation is executed in the pixel domain. The motion vectors of the subbands are predicted from these coarse motion vectors by appropriate scaling. These motion vectors are then refined using a small search window. Although, MRME-C provides a superior estimation, the complexity is very high since the initial motion estimation is executed in the pixel domain.

In this section, we propose two adaptive techniques to improve the motion estimation process. First, we propose an *adjustable resolution selection* (ARS) technique where the coarse motion estimation is performed at a pyramid level determined by the complexity constraint of the encoder. The fine motion estimation is done in a manner similar to AMRME technique. Secondly, an *adaptive bit allocation* (ABA) technique is proposed to distribute the bits optimally between the motion vectors and the displaced frame difference (DFD) information. These two techniques are now presented.

4.5.1 Adjustable Resolution Selection (ARS) Technique

We note that coarse-to-fine ME has a low complexity because of its hierarchical nature. However, as noted earlier, it does not provide accurate motion estimation. One of the major reasons for the failure of the coarse-to-fine ME is the small block-size used in the higher levels of the pyramid. For example, a 2×2 block in level-3 corresponds to a block-size of 16×16 in the pixel domain (as is done in MPEG-1). Hence, motion estimation using 16×16

blocks in pixel domain is equivalent to ME using 2x2 blocks in level-3. However, we note that the 2x2 block-size may be too small for robust estimation of motion vectors.

We propose an adjustable resolution selection (ARS) technique to estimate the coarse motion vectors at an appropriate level. Each video frame is wavelet-decomposed for m levels. The lowpass subband at level- m is then segmented into blocks of size $p2^{-m} \times q2^{-m}$, where $p \times q$ is the block size for ME in the pixel domain. We note that p and q are chosen, appropriately, depending on m and the frame-size. The specific level m to be used is chosen according to the complexity and accuracy requirement.

Once, the appropriate resolution is chosen, coarse motion estimation is performed by comparing the original DWT coefficients corresponding to the lowpass subband of the current and past reference frame. The schematic of the proposed scheme is shown in the first part of Fig. 4.11. A band memory (BM) stores the original coefficients of level- m lowpass subband coefficients of the previous reference frame. Coarse motion vectors are then estimated from these original coefficients using full search block matching techniques.

In ARS (L- m) technique, the coarse ME is performed at level- m , which is followed by a fine ME. Let the coarse motion vector at location (x, y) be denoted by $v_c(x, y)$. The fine ME at various scales can then be estimated using the following equations.

ARS (L-1)

$$V_4^o(x, y) = v_c(x, y) / 2 + \Delta_4^o(x, y) \quad (4.16)$$

$$V_8^o(x, y) = v_c(x, y) / 4 + \Delta_8^o(x, y) \quad (4.17)$$

ARS (L-2)

$$V_z''(x, y) = 2v_c(x, y) + \Delta_z''(x, y) \quad (4.18)$$

$$V_s''(x, y) = v_c(x, y)/2 + \Delta_s''(x, y) \quad (4.19)$$

ARS (L-3)

$$V_4''(x, y) = 2v_c(x, y) + \Delta_4''(x, y) \quad (4.20)$$

$$V_2''(x, y) = 4v_c(x, y) + 2\Delta_4''(x, y) + \Delta_2''(x, y) \quad (4.21)$$

After both coarse and fine ME is executed, the motion compensated error coefficients are quantized and encoded using scheme-3 of section 4.1.1. The complete process is illustrated in Fig. 4.11.

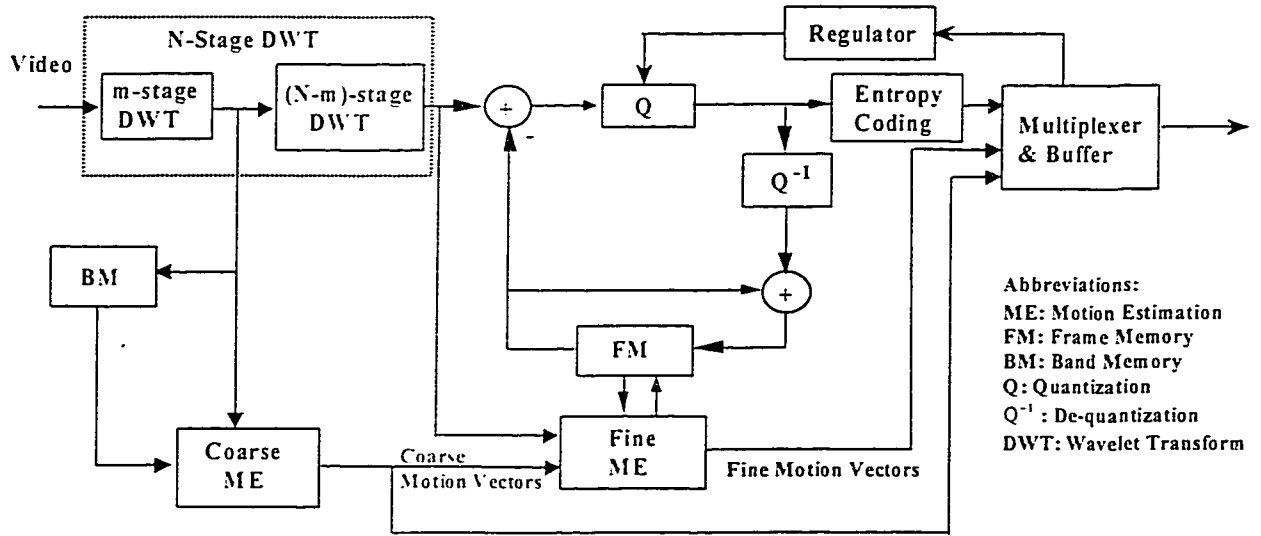


Figure 4.11: Video encoder employing Adjustable Resolution Selection (ARS) technique followed by AMRME technique.

We now compare the storage overhead of ARS and MRME-C techniques. Let the buffer space required to store an entire frame be B_F . The overheads due to storage of previous reference frame for MRME-C and ARS (L- m) technique are B_F and $B_F/4^m$, respectively.

Thus, ARS technique has a lower storage overhead. The computational complexity of various techniques (AMRME, MRME-C, ARS) is shown in Table 4.3. We note that the complexity of the ARS technique (at all levels) is substantially lower compared to the MRME-C technique while the complexities of ARS (L-2) and ARS (L-3) are comparable to that of AMRME. Thus, ARS techniques at different levels offer various degrees of complexity and estimation accuracy that can be exploited by the user.

4.5.1.1 Motion Estimation Performance

In this section, we evaluate the motion estimation performance of the ARS technique. We note that the MRME-C technique is a special case of ARS technique where the coarse ME is executed in level-0, and fine ME is calculated employing original DWT coefficients. A threshold factor (τ) of 0.9 has been employed in each case to discard the motion vectors that do not provide appreciable compensation. The GOP structure used is IPPPPPI. Each frame has been decomposed using the *Daubechies-8* tap wavelet.

Table 4.3

Computational complexity of various motion estimation techniques (in operations/pixel). r is the equivalent motion search area in pixel domain. Δ_z is the coarse motion search area of AMRME at level-3. Δ_m is the fine-motion search area in level- m . Typical values of overall complexity has been obtained assuming ($r=16$, $\Delta_z=4$, $\Delta_3=2$, $\Delta_2=2$, $\Delta_1=1$).

	Coarse ME (approx.)	Fine ME (approx.)	Typical Value (Overall)
Full Search	$4r^2$	-	1024
MRME-C	$4r^2$	$\Delta = 0.25\Delta_z^2 + 0.75\Delta_2^2 + 3\Delta_1^2$	1024+7
ARS (L-m)	$\frac{4r^2}{4^{2m}}$	Δ	L-1 $\rightarrow$ 64+7 L-2 $\rightarrow$ 4+7 L-3 $\rightarrow$ 0.25+7
AMRME	$0.25\Delta_z^2$	$0.75\Delta_2^2 + 3\Delta_1^2$	10

Effect of Block-size

We note that the accuracy of coarse motion vectors is crucial since the fine ME is performed based on coarse motion vectors. The effect of block-sizes on motion estimation performance is now presented. Fig. 4.13 shows the coarse motion vectors corresponding to frame-2 (shown in Fig. 4.12) of *pingpong* sequence. The motion vectors are estimated by AMRME technique at lowpass level-3 subband (S_8) using two different block-sizes, namely 2x2, and 3x3. We observe that the motion vectors estimated using 2x2 block-size are more noisy compared to those estimated using 3x3 block-size. In addition, 3x3 block-size also reduces the number of motion vectors and therefore results in a lower bit-rate required to encode the motion vectors. Hence, the remaining of the simulation has been performed using a block-size equivalent to 3x3 in level-3.



Figure 4.12: Frames 1 and 2 of Pingpong sequence. a) Frame-1, b) Frame-2. We note that frame-1 has a general shift towards left. In addition, there is significant movement in arm-region of the player.

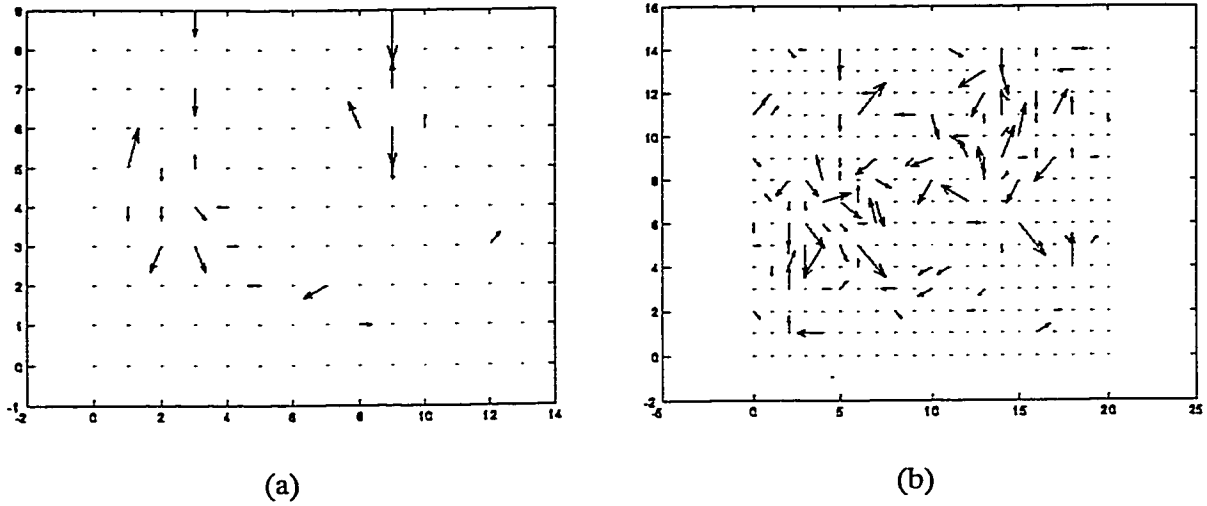


Figure 4.13: Motion vectors corresponding to frame-2 (with respect to frame-1) of pingpong sequence, estimated by AMRME technique. a) ME using 3x3 block-size at level-3, b) ME using 2x2 block-size at level-3

ME using original DWT coefficients

The performance of motion estimation employing original and reconstructed DWT coefficients is presented in Table 4.4. The table entries show the energy-ratio (SNR) of original and motion-compensated residuals of different subbands. "Reconst" refers to the ME employing reconstructed DWT coefficients, while "Original" refers to the ME employing original DWT coefficients. It is observed that SNR is significantly higher when the compensation is performed employing the original DWT coefficients. This proves that the motion vectors will have a better accuracy if ME is executed using the original DWT coefficients.

Coarse ME at different levels

Fig. 4.14 shows the coarse motion vectors corresponding to *pingpong* frame-2, estimated using $3 \times 2^{3-m} \times 3 \times 2^{3-m}$ block-size at level- m . We note that the estimated motion vectors corresponding to level-1 are very similar to (except for a scale factor of 2) those of level-0.

However, the motion vectors at higher levels (2 and 3) have deviated marginally from the motion vectors corresponding to level-0. A comparison of Fig. 4.13a and Fig. 4.14d (both of them have been obtained using ME at S_8) reveals that Fig. 4.13a contains a few noisy motion vectors at blocks (5,1), (9,9), (1,8) which are not present in Fig. 4.14d. Hence, the ARS technique provides better coarse motion estimation.

Table 4.4

SNR of motion compensation, between frame 1 and 2 of Salesman sequence, at various subbands. Search Area $p:q:r$ corresponds to the search area of p , q , and r in level-3, 2, and 1, respectively.

Quantization	Search Area	SNR at various Bands									
		0	1	2	3	4	5	6	7	8	9
Reconst	4:2:1	16787	103	73	28	18	11	3.1	2.8	2.1	1.00
Original	4:2:1	24014	335	127	110	128	46	29	14	5.4	1.05
Original	4:4:4	24014	335	127	110	128	46	29	14	5.6	1.58
Original	4:8:8	24014	335	127	110	128	46	29	14	5.8	1.63

4.5.1.2 Choice of Threshold Factors

It was observed in section 4.3.1 that choice of τ is crucial in obtaining a good coding performance. For AMRME technique, it was noted that the best choice of τ ranges from 0.5 to 0.7, depending on the video sequence. Since the proposed ARS technique is significantly different from the AMRME technique discussed in section 4.3, we reconsider the choice of τ for ARS technique.

We note that there are two types of ME in the ARS technique - coarse ME using the original lowpass DWT coefficients of a given scale, and the fine ME using the reconstructed DWT coefficients. Hence, we need two τ 's - one each for coarse and fine ME. Here, we have performed two experiments using ARS (L-1) technique. In the first experiment, we initialize the τ for coarse ME (τ_c) with a value of 0.5 (and keep it unchanged) and vary the

τ for fine ME (τ_F) from 0.6-0.9. The coding performance of ARS coder with different τ_F is shown in Fig. 4.15. It is observed that the overall coding performance is less sensitive to the choice of τ_F compared to the case of AMRME technique. In the second experiment, we compute the coding performance by keeping the τ_F constant (say 0.6) and varying the τ_C from 0.5 to 0.8. It has been observed that the coding performance is marginally sensitive to the choice of τ_C , and hence has not been shown graphically.

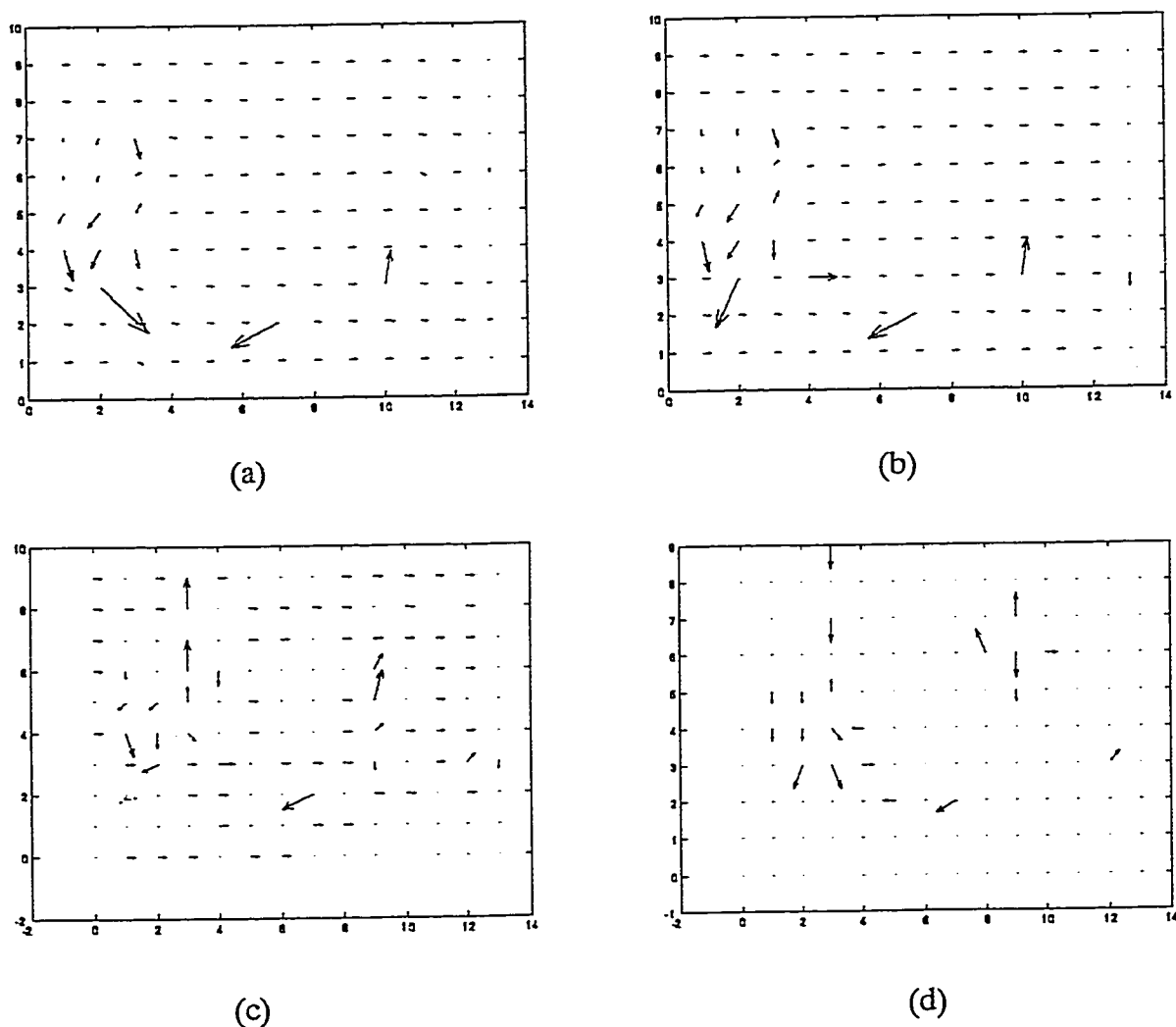
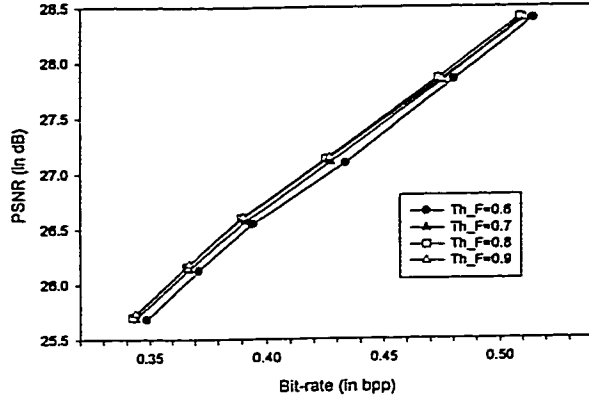
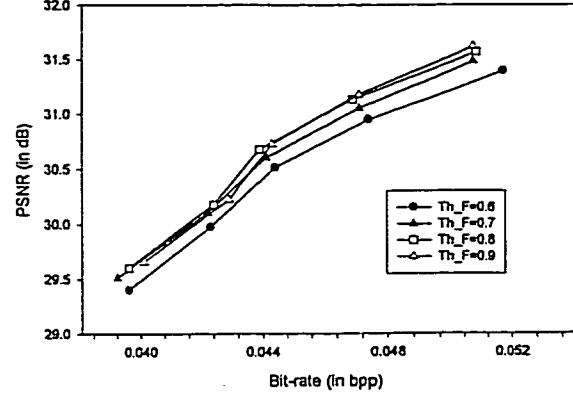


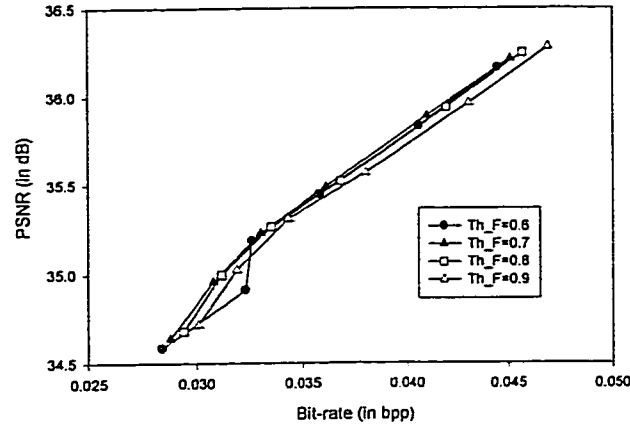
Figure 4.14: Motion vectors corresponding to frame-2 (with respect to frame-1) of pingpong sequence, estimated by ARS technique. a) Motion estimation (ME) in pixel domain (*i.e.* in level-0), b) ME in level-1, c) ME in level-2, d) ME in level-3.



(a)



(b)



(c)

Figure 4.15: Choice of threshold factors for fine ME in ARS (L-1) technique. (a) Pingpong, (b) Salesman, and (c) Miss America sequence. τ_F is shown as Th_F in the graphs. A coarse threshold factor (τ_C) of 0.5 has been employed for coarse ME in all cases.

A possible reason for the insensitivity of the coding efficiency to τ_C is that the coarse motion estimation using the original lowpass DWT coefficients are robust. We also note that the coding performance is marginally sensitive to the choice of τ_F since most of the prediction results from coarse ME, while τ_F is employed to refine the motion vectors. Ideally, coarse motion estimation should be as accurate as possible since these vectors will be

used for further prediction. But, the fine ME need not be very accurate since the main goal is to reduce the energy of the DFD blocks. Hence, τ_F should ideally be close to 1. Hence, in our subsequent analysis, we employ $\tau_C=0.5$ and $\tau_F=0.9$.

4.5.1.3 Coding Performance

We now evaluate the coding performance of ARS video coder shown in Fig. 4.11. The initial motion estimation for ARS (L- m) technique is performed at level- m using block-sizes of $(24*2^{-m}) \times (24*2^{-m})$. The maximum allowed displacement (MAXAD) for the ARS (L- m) technique is $16*2^{-m}$ pixels while the maximum allowed refinements (MAXAR) for level-3, level-2 and level-1 subimages are 2, 2 and 1 pixels, respectively. We have employed a τ_C of 0.5 and a τ_F of 0.9. The bit-rate and PSNR values have been obtained by averaging over 25 P-frames.

We note that the ARS technique performs coarse motion estimation with original DWT coefficients. However, it estimates the fine (incremental) motion vectors using reconstructed DWT coefficients of the previous reference frame. Fig. 4.16 shows the improvement (0.3-0.5 dB) in coding performance due to the use of reconstructed DWT coefficients. This is expected since the main objective here is to reduce the DFD energy rather than to obtain the correct motion vectors. However, accuracy of coarse motion vectors is very important since they are used for fine motion estimation.

The performance of the ARS technique with coarse motion estimation at various levels is shown in Fig. 4.17. We note that for the pingpong sequence, ARS (L-1) provides comparable performance with ARS (L-0), while a marginal degradation in performance is observed when the motion is estimated in level-2 (L-2) and level-3 (L-3) of the wavelet

pyramid. For the salesman sequence, however, ARS techniques of all levels provide similar performance. This is because the motion vectors are small in magnitude in the case of salesman sequence and estimating coarse motion vectors at lowpass band is therefore sufficient.

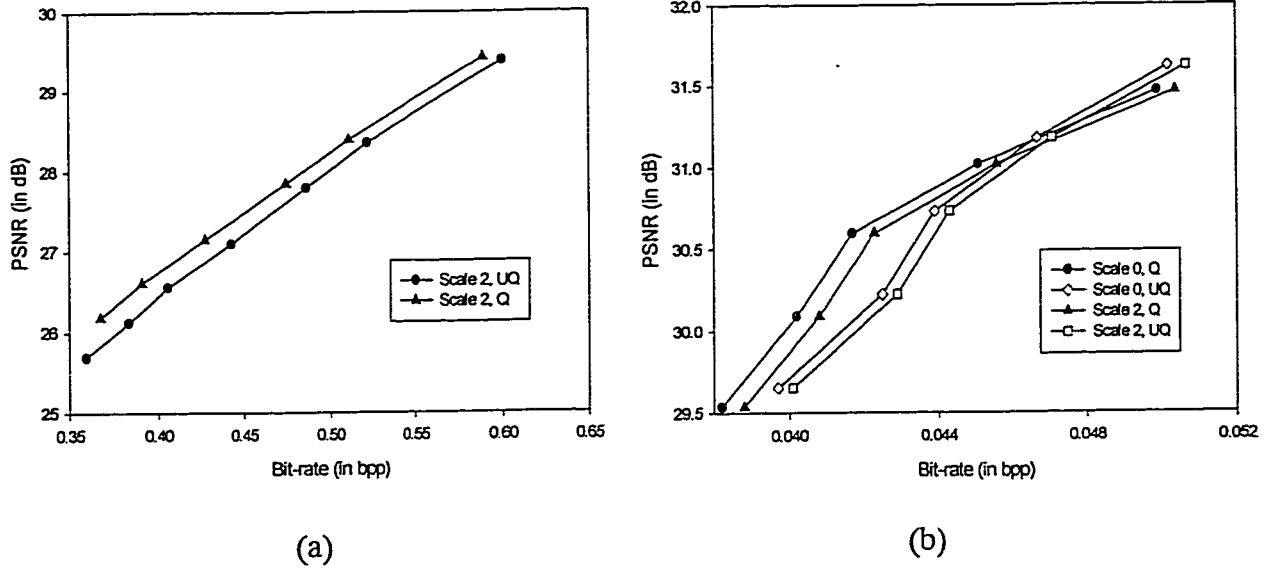
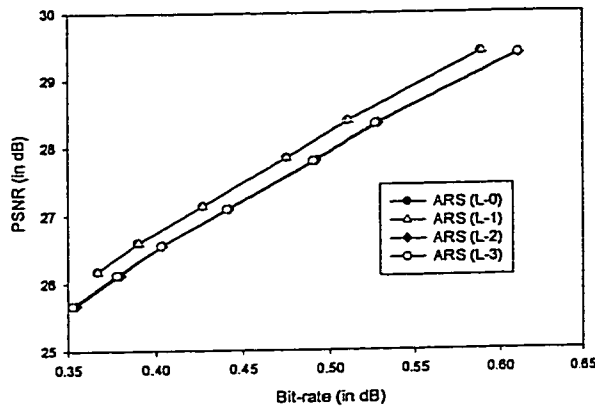
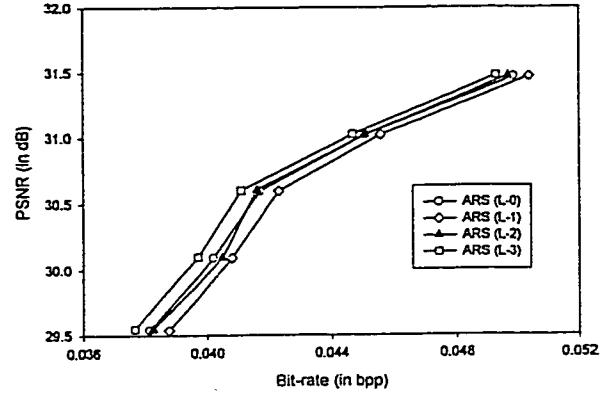


Figure 4.16: Performance of fine motion estimation using original and reconstructed DWT coefficients. a) Pingpong, b) Salesman sequence.

We now compare the performance of ARS technique with the performance of AMRME and MRME-C techniques. The initial motion estimation for MRME-C and AMRME techniques are performed at level-0, level-3 using block-sizes of 24x24 and 3x3, respectively. For AMRME technique, the MAXAD for level-3 subbands is 4 pixels and the MAXAR for level-2 and level-1 subbands are 2 and 1 pixels, respectively. The MAXAD for the MRME-C technique is 16 pixels while the MAXAR for level-3, level-2 and level-1 subimages are 2, 2 and 1 pixels, respectively.



(a)



(b)

Figure 4.17: Performance of ARS technique with motion estimation at various levels of wavelet pyramid. a) Pingpong, b) Salesman sequence.

The performance of the ARS technique is compared with other techniques in Fig. 4.18. We note that the bit-rate and PSNR values have been obtained by averaging over 25 frames. Since the I-frames have been coded using the same technique in all four cases, the bit-rate/PSNR of only P-frames have been considered. It is observed in Fig. 4.18 that the ARS technique provides an improvement of about one dB over AMRME technique. It (ARS) also provides a marginal improvement over the MRME-C technique since it refines the motion vectors using the reconstructed DWT coefficients in contrast to the MRME-C technique which refines the motion vector using the original DWT coefficients.

4.5.2 Adaptive Bit Allocation Technique

The motion estimation in AMRME and ARS video coder is performed employing blocks of $p2^{-n} \times q2^{-n}$ coefficients for n -th level subbands. There are two major limitations associated with this fixed block-size approach. First, when objects present in a scene are smaller than the block-size employed for ME, and the objects move in different directions, the motion

estimation does not perform well. In this case, a smaller block-size will provide better motion estimation of individual objects. Similarly, when a large object moves with a constant velocity in a scene, a single motion vector associated with a larger block-size encompassing the entire object may be more efficient since it reduces the number of motion vectors. Secondly, it is difficult to control the overall bit-rate of the coder with motion estimation techniques employing fixed block-size. In this section we propose an adaptive bit-allocation (ABA) technique which addresses these two issues.

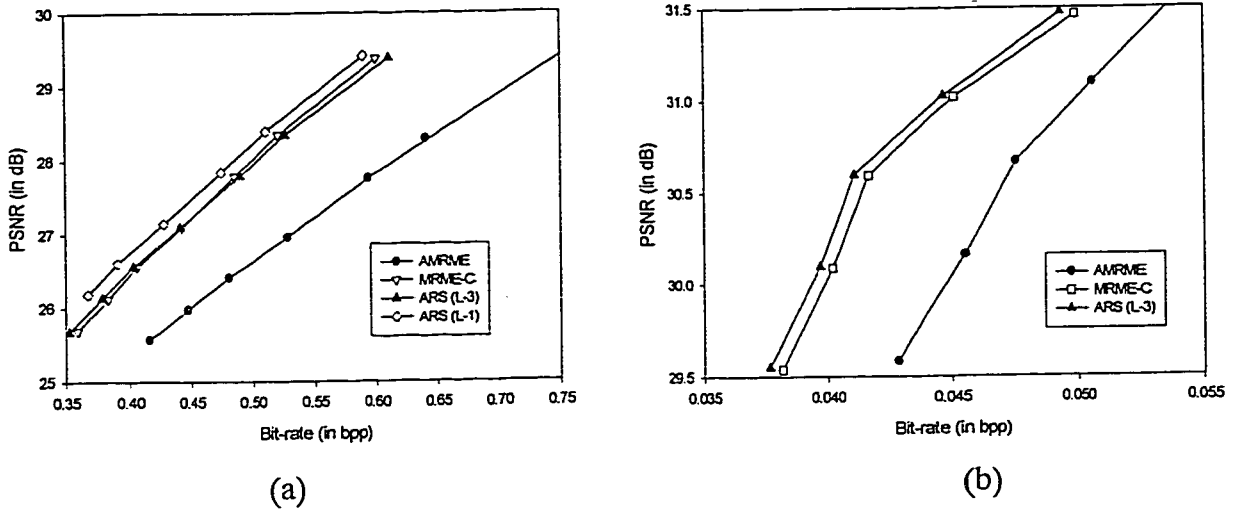


Figure 4.18: Performance comparison of AMRME, MRME-C, and ARS techniques. a) Pingpong , b) Salesman sequence. In MRME-C and ARS technique, adaptive threshold has been implicitly assumed.

We note that the total bit-rate for video coding can be expressed as:

$$R = R_{Motion} + R_{DFD} \quad (4.22)$$

In order to obtain a superior coding performance, the available bit-budget should be distributed optimally between the motion vectors and the DFD. We note that the distribution of bits according to Eq. 4.22 is influenced by the choice of block-size. Generally, a larger

block size produces fewer motion vectors and thus results in a smaller R_{Motion} , but it also produces a larger R_{DFD} . Therefore, a judicious choice of block-size is crucial in obtaining the best coding performance. Since AMRME and MRME-C techniques employ fixed block size, only a sub-optimal coding performance can be achieved using these techniques.

We note that Eq. 4.22 is a constrained problem of allocating the bits subject to minimum distortion and is difficult to solve. In the proposed ABA technique, the constrained problem of Eq. 4.22 is transformed into an unconstrained optimization problem using the Lagrange multiplier method. We define the associated cost function as:

$$J = D + L_M R \quad (4.23)$$

where D is the distortion, R is the rate, and L_M is the Lagrange multiplier penalty function. Since the wavelet transform is orthogonal, the distortion will be additive if we employ the motion compensated error energy as the distortion. In this case, both D and R are additive (rate is always additive) and hence the cost function J is also additive. It has been argued in [17, 89] that for additive cost functions, the overall performance is an optimum point in the rate-distortion plane if all the subbands are encoded by minimizing the cost function for a given L_M . Here, we employ the same principle to determine the optimal bit-allocation between the motion vectors and DFD.

The principle of ABA technique is as follows. Assume that the average entropy of a motion vector is R_{MV} . Each block in the wavelet domain is then considered for either splitting or merging. A given block (termed as the *father*) is split into four equally sized smaller blocks (denoted by *children*) if

$$D_{Father} + L_M R_{MV} > \sum_{i=1}^4 (D_{child-i} + L_M R_{MV}) \quad (4.24)$$

where $D_{child-i}$ is the residual error energy for *child-i*. Eq. 4.24, effectively, allows a splitting only if the gain in reducing the prediction energy compensates the loss in using additional motion vectors.

On the other hand, four neighboring blocks (termed as the *children*) are merged into a single block (denoted by *father*) if

$$D_{Father} + L_M R_{MV} < \sum_{i=1}^4 (D_{child-i} + L_M R_{MV}) \quad (4.25)$$

Eq. 4.25, effectively, allows a merging only if the gain in using fewer motion vectors compensates the loss in increasing the prediction energy.

The schematic of the proposed technique is shown in Fig. 4.19. The coarse motion estimation is performed using ARS technique with fixed block-size. This is followed by fine motion estimation employing variable block-size. However, we note that R_{MV} should be estimated first in order to use Eqs. 4.24 and 4.25. This can be done in two ways. In the first method, the entropy corresponding to the previous frame can be used for the current frame. Since, the contents of neighboring frames are highly correlated, this approach should provide good estimation. In the second method, the entropy can first be estimated using AMRME technique employing a fixed block-size. The entropy corresponding to various subimages is then employed for motion estimation in ABA technique. Once, R_{MV} is known, the ABA can be performed using Eqs. 4.24 and 4.25. We note that the cost of encoding the decomposition vector should also be included in these equations. Although they have not been shown here

for clarity, the simulation results that will be presented in the following section include these costs.

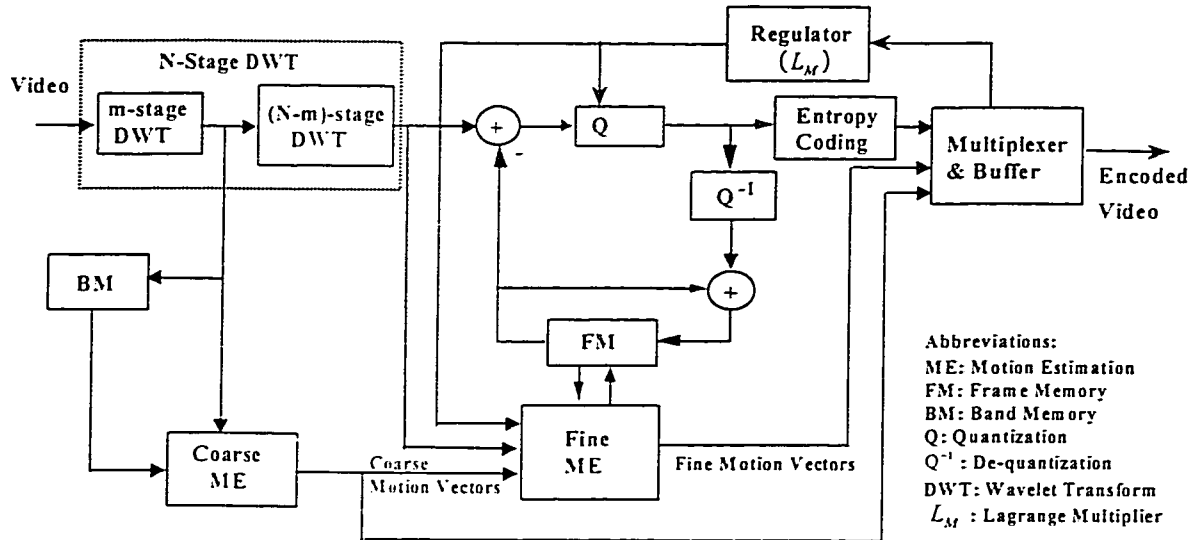


Figure 4.19: Video coder employing ARS followed by ABA Techniques

4.5.2.1 Performance of the ABA Technique

In this section, we evaluate the performance of the ABA technique. The video coder shown in Fig. 4.11 is employed to evaluate the performance of ARS technique (followed by MRME) whereas the ABA+ARS technique is evaluated using Fig. 4.19. The bit-rate and PSNR values reported have been obtained by averaging over 25 P frames. In case of ARS technique, block-sizes of 3x3, 6x6 and 12x12 have been employed for motion compensation in level-3, level-2, and level-1, respectively.

Here, we have implemented a simple version of ABA technique, where the biggest allowed block size (BABS) at level-3, level-2, and level-1 are 6x6 and 12x12, and 24x24, respectively. On the other hand, the smallest allowed block size at level-3, level-2, and level-

1 are 3x3, 3x3, and 6x6, respectively. All motion vectors are estimated using single-pixel accuracy.

Fig. 4.20 shows typical block formation for motion estimation in ARS, and ABA+ARS technique. This particular example corresponds to frame-2 of Pingpong sequence. There are three kinds of blocks shown in Fig. 4.20. The *black* blocks represent areas where motion compensation has failed; the *white* blocks represent areas with regular motion compensation; and the gray (only in ABA+ARS) blocks represent smaller blocks for efficient motion compensation. We observe from Fig. 4.20(a) that the block-size does not vary within a pyramid level.

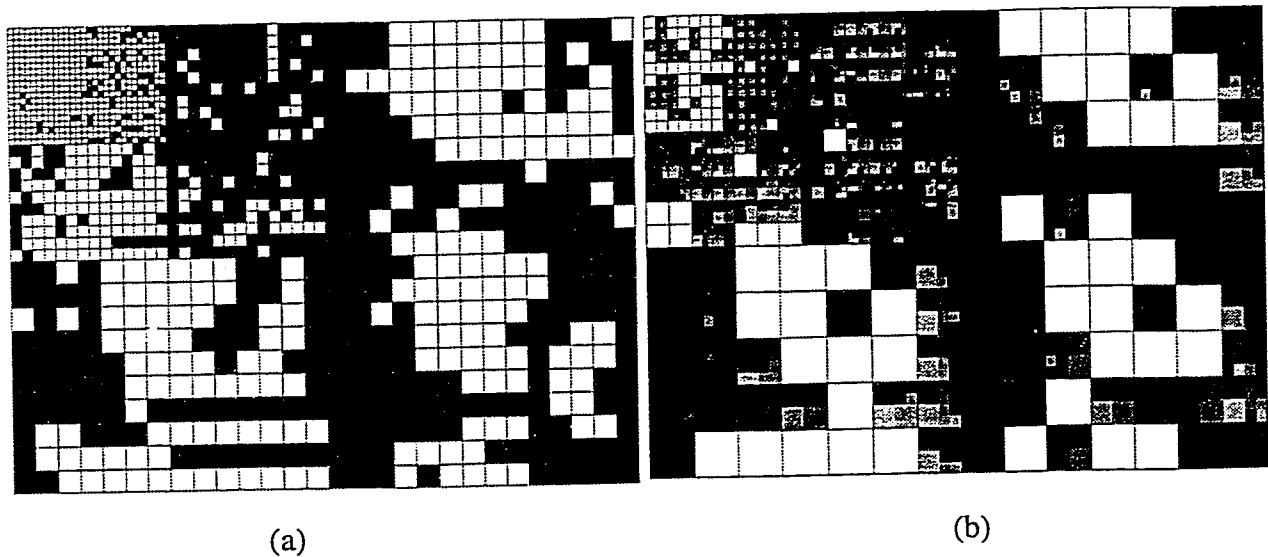


Figure 4.20: Typical motion blocks in a) ARS, and b) ABA+ARS technique. There are three kinds of blocks - black: no motion compensation, white: regular motion compensation, gray (only in ABA+ARS): smaller blocks for efficient motion compensation.

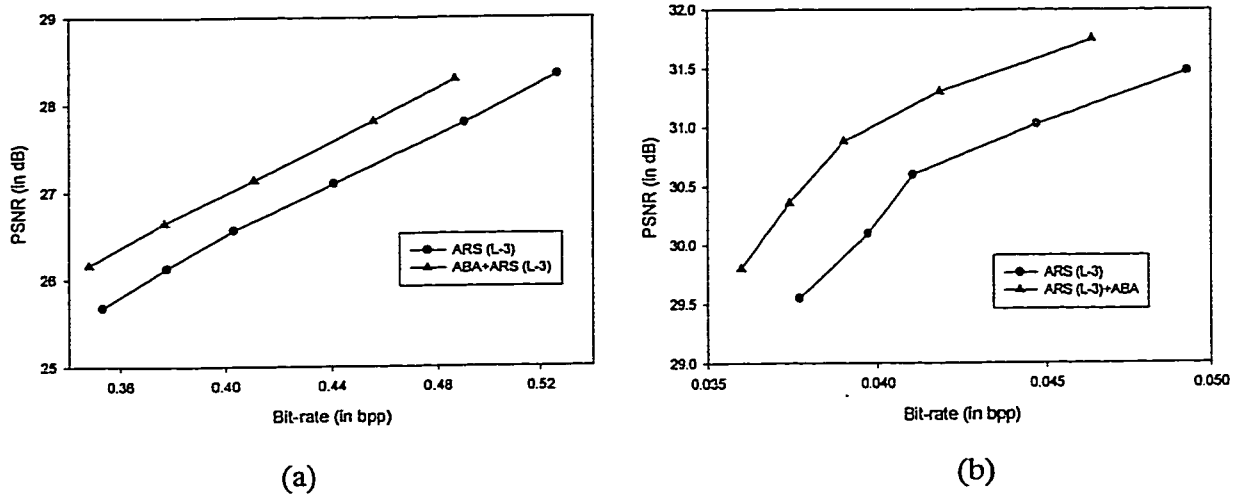


Figure 4.21: Coding performance of ABA+ARS technique. a) Pingpong, b) Salesman sequence.

The performance of the proposed video coder, *i.e.*, combined ARS and ABA technique, is shown in Fig. 4.21. It is observed that the ABA technique provides a further improvement of 0.5-1 dB over ARS technique. The coding performance can be further improved by employing smaller block-sizes and estimating motion vectors using fraction-pixel accuracy.

4.5.3 Bi-directional ARS/ABA Technique

We recall from section 4.4 that bi-directional motion estimation provides a superior coding performance compared to uni-directional motion estimation. The BMRME-2 technique proposed in section 4.4 performs bi-directional motion estimation in the lowest resolution bands and initializes the time flag (TFLAG). For other wavelet bands, motion estimation is essentially uni-directional and the motion vectors are computed either from previous or from next reference frame depending on the TFLAG.

In this section, we propose a bi-directional ARS/ABA technique to further improve the coding efficiency of ARS/ABA. In bi-directional ARS (henceforth referred to as

ARS+BMRME) technique, the coarse motion vectors are calculated using bi-directional motion estimation (BMRME-2) at a selected level (L-m) and a temporal flag (TFLAG) is initialized for each block depending. If the best match block is from the previous frame, TFLAG is set to 0, and if the best match block is from the next reference frame TFLAG is set to 1. During fine ME, the motion vectors are estimated with respect to either previous or next reference frame depending on the corresponding TFLAG which has been calculated during coarse ME.

We note that the bi-directional ABA (henceforth referred to as ABA+BMRME) technique can be implemented in a manner similar to ARS+BMRME with appropriate modifications. The modifications are required since the block-size is not fixed as in the case of ARS technique. There are three steps in ABA+BMRME technique. First, the wavelet bands are divided into a set of special blocks (SB) with size BABS (see section 4.5.2.1 for more details). We note that a SB contains several normal blocks (NB) that are used for motion estimation in ARS technique. The value of the BABS is user-selectable and the values of BABS in different levels are related by a factor of 2x2. Secondly, TFLAGS are computed for each SB using the original lowpass DWT coefficients at the level (L-m) selected for coarse ME. Finally, coarse and fine motion vectors are computed for each NB either from previous or from next reference frame depending on the TFLAG of the corresponding SB. In other words, the estimation of coarse and fine motion vectors in ABA+BMRME is similar to that of ABA+MRME within a given SB except that the former has a choice between the previous and next reference frame to achieve a lower DFD energy.

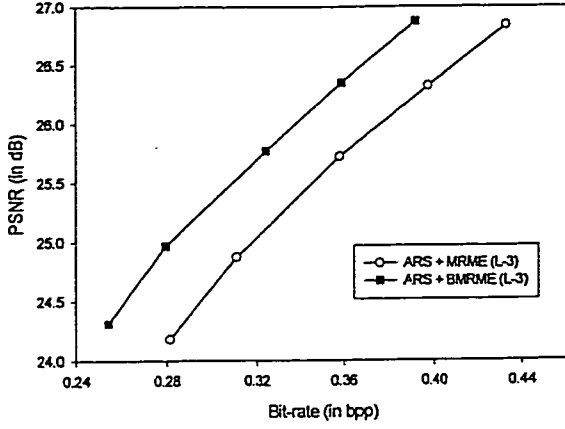
We now evaluate the coding performance of bi-directional ARS (ARS+BMRME) technique. In order to make a fair comparison, we compare the coding performance of bi-

directional ARS technique with that of uni-directional ARS technique (ARS+MRME). The ARS+BMRME employs a GOP (group of pictures) as IBBBBPBBBBBPBBBBI while the ARS+MRME technique employs a GOP as IUUUUPUUUUPUUUI. The U- frames are coded in the same manner as that of B- frame, except that the motion compensation is obtained only from the past reference frame. A τ_c of 0.5 is used for coarse motion estimation while a τ_F of 0.9 is used for fine motion estimation for all sequences.

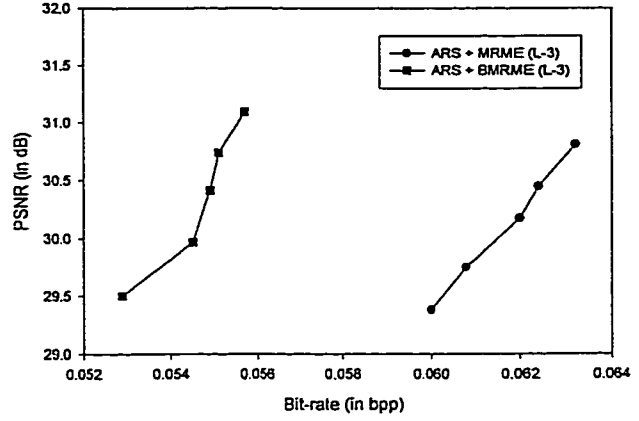
The coding performance of ARS+BMRME technique is shown in Fig. 4.22. It is observed that for pingpong sequence an improvement of more than 0.5 dB is achieved for P- and B- frames. On the other hand, a substantial reduction in bit-rate is achieved for salesman sequence at a similar bit-rate.

4.6 Comparison of the Proposed Techniques

We now evaluate the coding performance of all motion estimation techniques proposed in this chapter. The full search algorithm with *mean square error* (MSE) as the matching criterion has been used to obtain the motion vectors. For MRME-Z, AMRME, ARS, and ARS+BMRME techniques, the block sizes of the level-3, level-2 and level-1 wavelet subimages have been chosen to be 3×3 , 6×6 and 12×12 , respectively. For MRME-Z, and AMRME techniques, the maximum allowed displacement (MAXAD) for level-3 subimages is 4 pixels and the maximum allowed refinements (MAXAR) in level-2 and level-1 subimages are 2 and 1 pixel, respectively. On the other hand, for ARS, and ARS+BMRME techniques, the MAXAD for level-3, and MAXAR for level-2, and level-1 subimages are 2, 2, and 1 pixel, respectively.



(a)

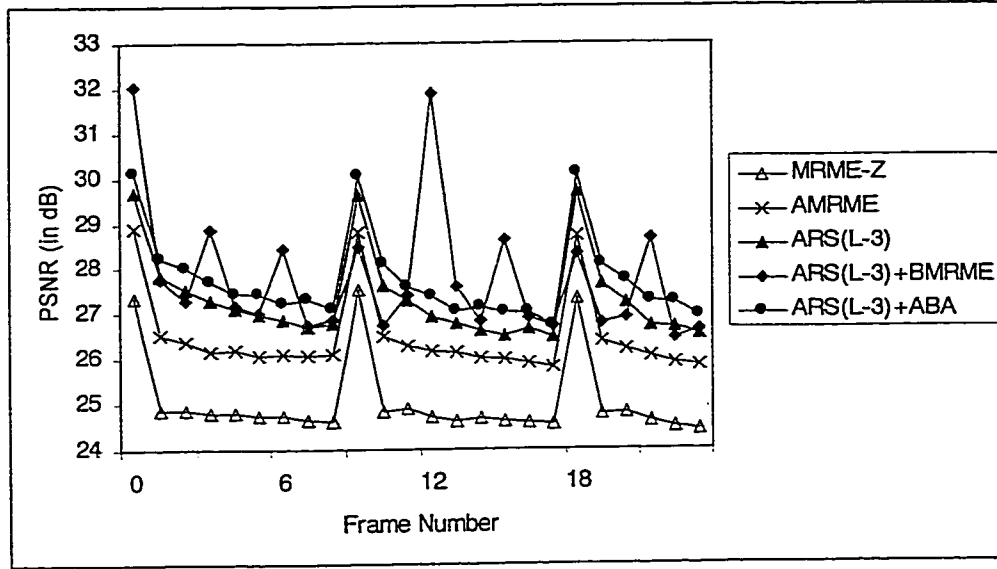


(b)

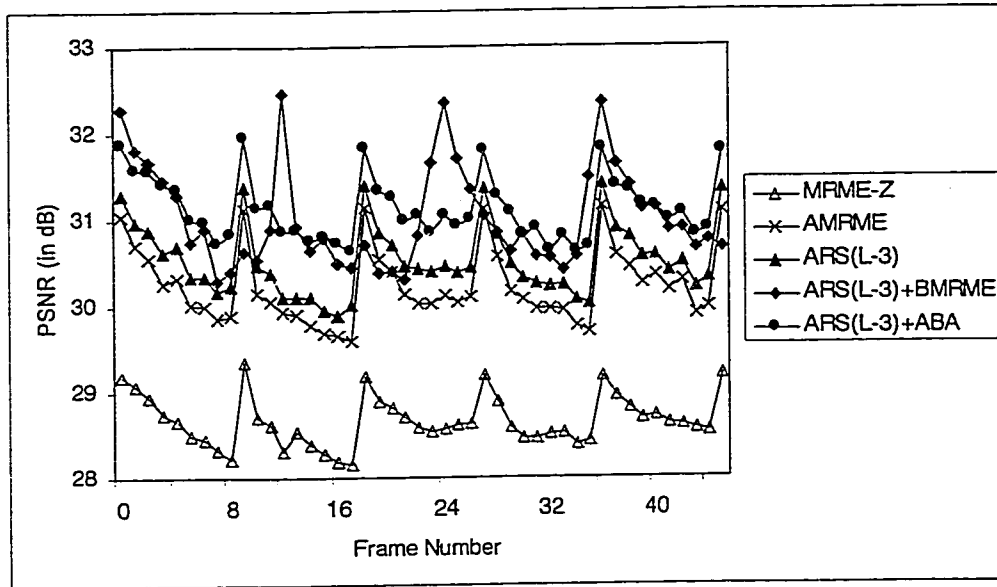
Figure 4.22: Performance comparison of ARS+MRME and ARS+BMRME techniques. a) Pingpong, b) Salesman sequence. Adaptive thresholds have been employed in both cases.

For ARS+ABA technique, the biggest allowed block size (BABS) at level-3, level-2, and level-1 are 6x6 and 12x12, and 24x24, respectively. On the other hand, the smallest allowed block size at level-3, level-2, and level-1 are 3x3, 3x3, and 6x6, respectively. All motion vectors are estimated using single-pixel accuracy.

The *group of pictures* for uni-, and bi- directional motion estimation are, respectively, IPPPPPPPI, and IBBPBBPBBPBI. The overall bit-rate and PSNR have been calculated by averaging the bit-rate and MSE for all (I-, P-, and B-) frames. The video frames are decomposed with Daubechies-8 tap wavelet. The DWT coefficients as well as the motion compensated error coefficients have been quantized using scheme-3 described in section 4.1.1.3.



(a)



(b)

Figure 4.23: Peak signal-to-noise ratio (PSNR) values of reconstructed sequence. (a) Pingpong sequence coded at an average rate of 0.5 *bpp*, (b) Salesman sequence coded at an average rate of 0.08 *bpp*.

The coding performance of the proposed techniques at an average bit-rate of 0.5 *bpp* (for pingpong) and 0.08 *bpp* (for salesman sequence) is shown in Fig. 4.23. It is observed that ARS+BMRME and ARS+ABA techniques provide the best coding performance. The

coding performance at different bit-rates is shown in Fig. 4.24. We note that the bit-rate, and PSNR shown in Fig. 4.24 have been calculated by averaging 30 frames for pingpong sequence. For salesman sequence the bit-rate, and PSNR have been calculated by averaging 50 frames. It is observed that the proposed techniques provide a significant improvement over MRME-Z technique for both sequences. For pingpong sequence, both ABA and BMRME techniques provide an improvement of at least 0.7 dB over ARS technique. For salesman sequence, ABA provides about 1 dB improvement over ARS, while BMRME provides 0.5-0.7 dB improvement. Joint ABA and BMRME is expected to provide further improvements in coding performance.

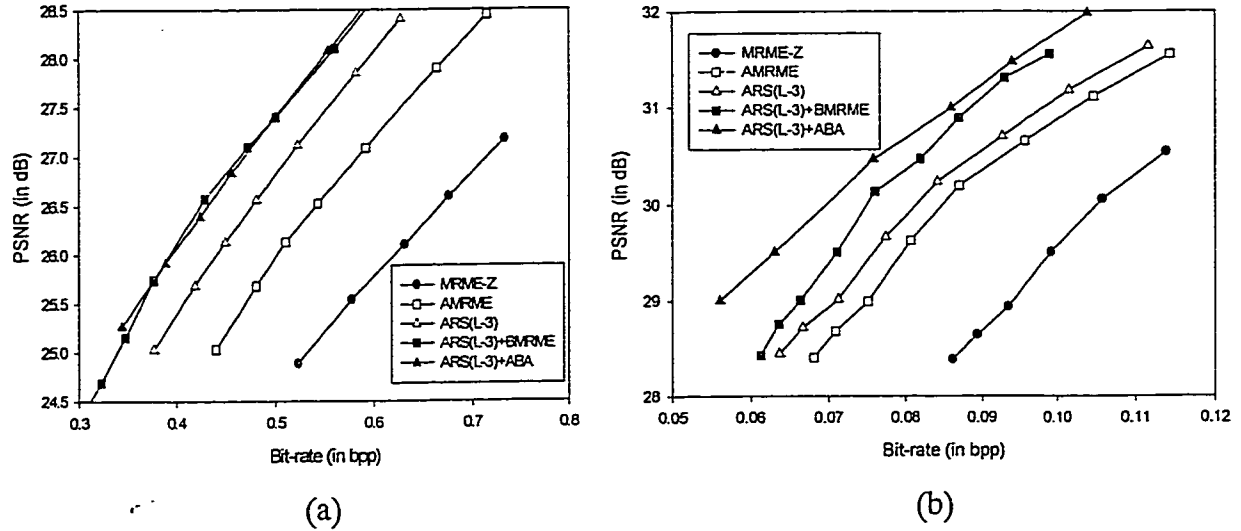


Figure 4.24: Coding performance of the proposed techniques. (a) Pingpong, (b) Salesman sequence. $\tau=0.6$ for AMRME technique while for ARS, and ABA techniques, $\tau_C=0.5$, and $\tau_F=0.9$.

4.7 Summary

In this chapter, we have presented several techniques for motion estimation in the wavelet domain. First, we proposed an AMRME technique where an adaptive threshold is employed for each block to determine if motion compensation is effective. Secondly, we proposed a bi-directional motion estimation technique (BMRME) for efficient coding in the wavelet domain. Third, we proposed an adjustable resolution selection (ARS) technique to perform efficient motion estimation with a reduced complexity. Fourth, an adaptive bit-allocation (ABA) technique has been proposed to estimate the optimal block-size for motion estimation. Finally, a bi-directional ARS/ABA technique has been proposed by combining the BMRME and ARS/ABA techniques. These techniques jointly provide a superior coding performance compared to the existing MRME technique in a wavelet-based video compression system.

Chapter 5

Image Indexing using Histograms, Moments and Wavelets

We recall from chapter 3 that histogram-based techniques are popular in image and video indexing applications because of their low complexity and good performance. In addition, the computational complexity of histogram-based techniques could be further reduced by employing the moments of histogram, rather than the histogram itself. In this chapter, we propose several new techniques for image indexing based on histograms, moments, and wavelet coefficients. The organization of this chapter is as follows. Section 5.1 addresses the issue of image indexing using histograms and moments. Novel image indexing techniques in the wavelet domain are presented in section 5.2. A joint moment and wavelet technique is presented in section 5.2.3. A video indexing technique based on wavelet coefficients and motion vectors is presented in section 5.3. This chapter concludes with a summary.

5.1 Indexing by Histogram/Moments

5.1.1 Evaluation of Histogram/Moment-based Techniques

Histogram comparison has been accepted as an effective tool for image indexing. In this section, we present a detailed performance analysis of histogram and moment-based techniques. We note that EC-3 criterion (section 3.7) has been applied to evaluate the indexing techniques.

5.1.1.1 Difference of Image Histogram

The performance of *difference of image histogram* (DOIH) technique, discussed in section 3.1.2, is evaluated employing an absolute error (L^1) metric on images from database IDB1. A tolerance (T) of 5 (see section 3.7) has been used in the experiment. It is observed that the efficiency of retrieval (η_R) is approximately 92.3% for the database IDB1. This means, in each trial an image similar to the query image was present in the first five retrieved images in 92.3% cases. The 7.7% cases of failure arise due to the fact that histograms of different images can be very similar.

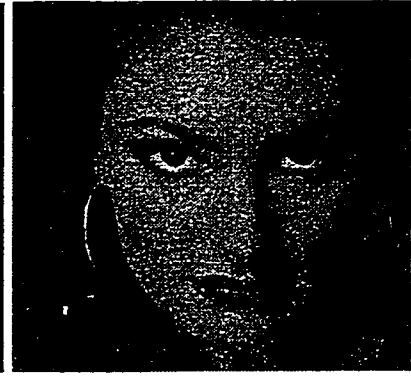
A visual example of histogram-based retrieval is shown in Fig. 5.1 where the upper leftmost image is the query image and the other eight images are retrieved images from the database. Based on the image histograms, the similarity between the retrieved images and the query image decreases from left to right and from top to bottom. The relevant histograms have been plotted in Fig. 5.2. It is observed that the histograms of the images of an owl, shown in Fig. 5.1(e-h) are closer to histogram of the query image in Fig. 5.1(a) compared to Fig. 5.1(i). A close examination reveals that Fig. 5.1(i) has substantial amount of black hair that results in a large peak in the histogram in the bin-range 32-64. As a result of this, the histogram difference is quite high and the technique fails to retrieve the desired image.



(a)



(b) ($E=9492$)



(c) ($E=9546$)



(d) ($E=13088$)



(e) ($E=21418$)



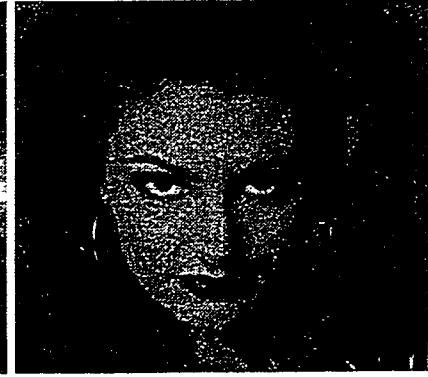
(f) ($E=21858$)



(g) ($E=23028$)



(h) ($E=23244$)



(i) ($E=25758$)

Figure 5.1. Images retrieved using histogram-based technique. The image at the upper left was selected randomly; the remainder of the faces are 8 most-similar images from the database with respect to histogram. Similarity decreases left to right, top to bottom. The L^1 distances are shown in parenthesis.

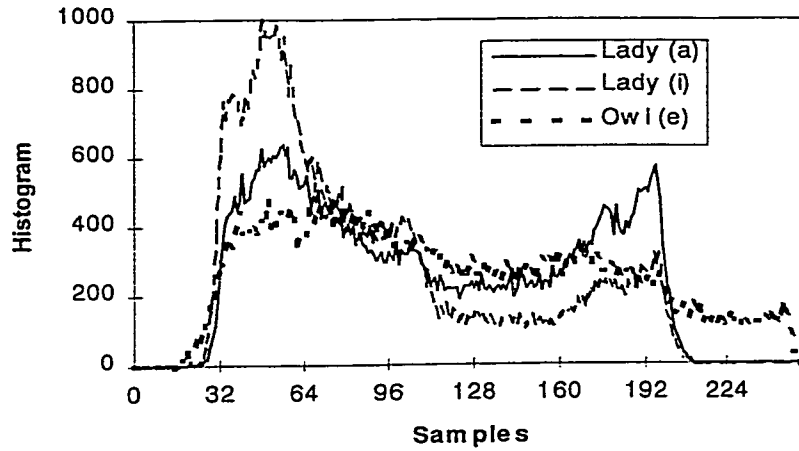


Figure 5.2. Histograms of the images shown in Fig. 5.1a, 5.1e and 5.1i.

The performance in the above experiment was evaluated employing absolute error (L^1) metric. However, as noted in section 3.6, several distance metrics can be used to compare histograms. The square error (L^2) metric is employed in many cases. The Mahalanabis distance (see section 3.6), although statistically optimal, is generally not recommended due to the inherent instability associated with inverting a large covariance matrix. In addition, the computational complexity is also high in this case. We now address the issue of the best choice of p when L^p metric is employed for indexing. The variation in indexing performance over different choices of p with tolerance (T) of 5 is shown in Fig. 5.3. Performances of original histogram as well as moment-reconstructed histograms are shown in the figure. It is observed that the performance of original histogram is sensitive to the choice of the L^p -metric, the best performance being achieved around $p \approx 0.5$. At this value, the histogram technique provides a superior performance of roughly 95%. For higher p , the effect of large error components dominates over the smaller error components, and degrades the performance. A small value of p diminishes the differences between two histograms,

and provides more robust retrieval. However, for very small values of p , the error metric fails to discriminate between histograms, and results in a poor performance. Thus, $p \approx 0.5$ seems to be a near-optimal value. However, we note that the complexity of implementing $L^{0.5}$ -metric is higher compared to L^1 -metric since the former requires a *square root* operation.

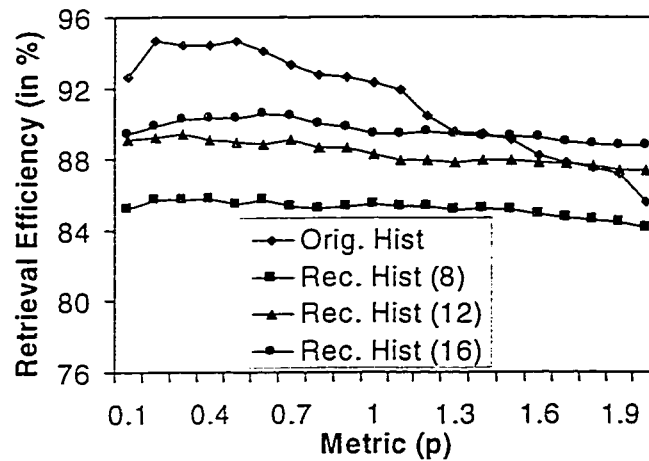


Figure 5.3. Performance of different Metrics for indexing. "Rec. Hist (M)" refers to the histogram reconstructed from M nontrivial moments.

5.1.1.2 Difference of Reconstructed Histograms

Simulations have also been carried out to evaluate the performance of histograms reconstructed from a finite number of lower order regular moments. Here, the histograms are reconstructed using the first 8, 12, and 16 regular moments (excluding the zeroth order moment). We note that reconstruction of a signal from a finite number of regular moments is an ill-posed problem and does not have a unique solution. Here, we have employed the technique described in [49, section 9.8] to reconstruct the histograms. The simulation results are shown in Fig. 5.3. It is observed that the indexing performance of reconstructed

histograms is relatively insensitive to the choice of metric- p in contrast to the original histogram. The most probable reason is that the reconstructed histograms, using a small number of moments, are generally smooth compared to the original histograms, and results in smaller error components. Therefore, a large difference in histograms of similar images is not expected and hence the performance is robust to changes in difference metric. However, the overall performance of reconstructed histograms is lower compared to the original histogram because of this smoothing.

Next, we consider the issue of the optimum number of moments to be used in representing histograms of probability density functions (*pdf*). We note that the minimum number of moments required to represent a *pdf* with reasonable accuracy depends on the characteristics of the *pdf*. The *pdf* shown in Fig. 3.5. can be represented reasonably well using 10-16 moments. However, a *pdf* with sharp peaks cannot be represented well with a few moments. This brings up the issue of the efficiency of moments as a descriptor of a signal (here, the histogram). Here, we evaluate the efficiency of moments by computing the *SNR* of original and reconstructed (using finite number of moments) histograms. We define the *SNR* as follows:

$$SNR = \frac{\int f^2(x)dx}{\int [f(x) - f'(x)]^2 dx} \quad (5.1)$$

where, $f(x)$ and $f'(x)$ are the original and moment-reconstructed histogram, respectively. *SNR* values have been calculated for the histograms of all images in IDB1. Table 5.1 shows the percentage of images having a *SNR* value greater than a certain threshold value using 8, 12 and 16 moments. It can be seen from Table 5.1 that at least 12 moments are required to have a reasonable confidence level.

5.1.1.3 Difference of Moments of Histograms

We now evaluate the performance of indexing techniques employing the difference of moments (DOM). Here, we consider regular as well as central moments. Table 5.2 shows regular (column-2) and central moments (column-4) of the images shown in Fig. 3.3a. These moments have been calculated on integer grids in the interval $[0, 255]$. We note that the magnitudes of the regular and central moments grow exponentially with increasing order, and as a result, appropriate selection of weights for moment comparison is difficult.

Table 5.1

SNR of reconstructed *pdf*'s using finite number of moments

SNR	No. of moments		
	8	12	16 ⁺
>8 dB	40%	58%	72%
>9 dB	38%	50%	63%
>10 dB	35%	46%	57%
>11 dB	34%	42%	53%
>12 dB	23%	38%	48%

⁺ Using 16 moments, 72% of the reconstructed images have SNR > 8 dB

Stricker *et al.* [107] have proposed to employ normalized moments (see Eq. 3.5) since the magnitudes are generally of the same order because of the normalization (see column 3 and 5 in Table 5.2). Hence, while comparing the moments, the normalized moments are likely to provide a better performance. Alternatively, the exponential growth of the regular moments can be reduced by employing a grid with smaller dynamic range for moment computation. Table 5.3 shows various moments of the same image calculated on a grid $[-1, -1+x, -1+2x, \dots, 1-x, 1]$ where $x=2/255$ for a 256 bin histogram. Here, the histogram bins are equally spaced and the histogram waveform remains unchanged. However, the dynamic

range of moments of different order is now on the same level. Hence, we will employ this grid, *i.e.*, $[-1,1]$ to evaluate the performance of various moment techniques.

Table 5.2

Various moments (on grid $[0;1;255]$) of histogram of image shown in Fig. 3.3a

M	Regular	Regular Normalized	Central	Central Normalized
1	144.59	144.59	0	0
2	26195.0	161.85	5289.42	72.73
3	5.30e+06	174.38	-13874.3	-24.03
4	1.13e+09	183.40	3.89e+07	78.99
5	2.48e+11	190.06	-3.17e+08	-50.13
6	5.52e+13	195.14	3.18e+11	82.61
7	1.24e+16	199.15	-4.80e+12	-64.80
8	2.82e+18	202.39	2.76e+15	85.14

Table 5.3

Various moments (on grid $[-1;2/255;1]$) of histogram of image shown in Fig. 3.3a

M	Regular	Regular Normalized	Central	Central Normalized
1	0.134	0.134	0	0
2	0.343	0.585	0.325	0.570
3	0.126	0.502	-0.006	-0.188
4	0.179	0.650	0.147	0.619
5	0.095	0.625	-0.009	-0.393
6	0.107	0.689	0.074	0.647
7	0.069	0.683	-0.008	-0.508
8	0.069	0.716	0.039	0.667

The performance of normalized regular moments (RGNM) and normalized central moments (CNNM) on database IDB1, using L^1 and L^2 metrics, is shown in Fig. 5.4. The

weights for moments of different order have been chosen to be 1.0 (other weights were also tried, but the η_R 's remain very close). It is observed from Fig. 5.4 that a retrieval efficiency of 55-65% is achieved using these moments. The performance of RGNM is marginally better than the corresponding CNNM. We note that the indexing performance is not highly sensitive to the choice of metric- p . Hence, unless specified, we employ L^2 metric in subsequent performance evaluations.

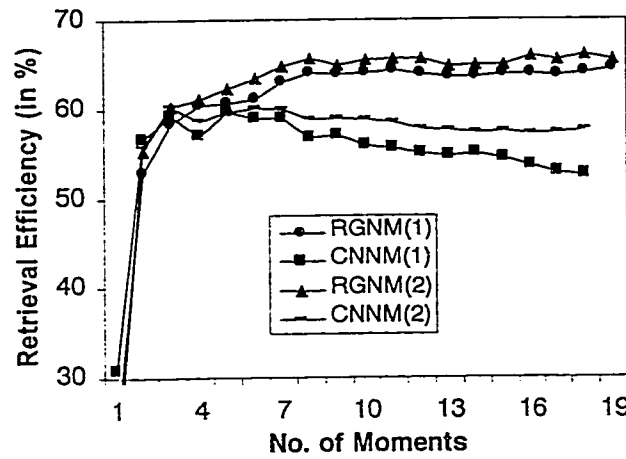


Figure 5.4. Indexing performance of various moments with L^1/L^2 metric. RGNM: regular normalized moments, CNNM: central normalized moments. RGNM(k) employs L^k metric for comparison.

Fig. 5.5 shows the performance of regular moments (RGM), central moments (CNM), RGNM, and CNNM on images from database IDB1 using L^2 metric. It is observed that the performance of RGM and CNM is marginally superior compared to the RGNM and CNNM.

It is noted from Figs. 5.4 and 5.5 that the performance of regular, central and normalized moments is substantially poorer compared to the performance of the moment-reconstructed histogram (MRH) shown in Fig. 5.3. This is because the regular moments are projections of the histogram onto monomials $\{x^k\}$ that are not orthogonal. Hence, the resulting moments

have substantial redundancy in them and the error vectors are not orthogonal. Performance is expected to improve by employing Mahalanabis distance that orthogonalizes the error space. The performance of L^2 metric is compared with that of Mahalanabis distance in Fig. 5.6. It is observed that Mahalanabis distance provides a superior performance compared to the regular moments. We also note that it provides a similar performance as that of moment-reconstructed histogram.

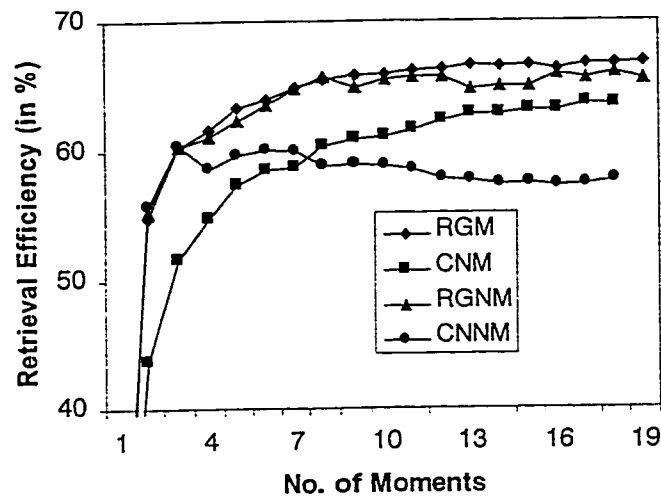


Figure 5.5. Indexing performance of various moments with L^2 metric on IDB1. RGM: regular moments, CNM: central moments, RGNM: regular normalized moments, CNNM: central normalized moments.

5.1.1.4 Difference of 2-D Moments of Image

We recall from section 3.1.2 that 2D moments (of the image, and not of the image histogram) are effective tools in pattern recognition. Here, we evaluate the performance of various 2-D moments in image retrieval from the database IDB1. It is observed from Fig. 5.7 that the performance of 2D moments of an image is substantially poorer compared to 1-D moments of the histogram. The best performance (approximately 50%) is achieved using orthogonal Zernike moments.

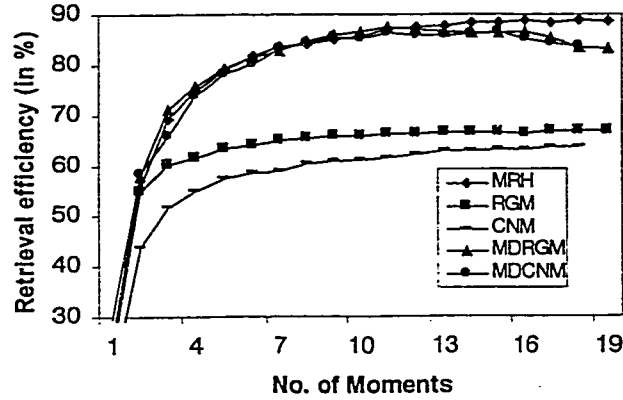


Figure 5.6. Indexing performance of various moments. MRH: moment-reconstructed histogram, RGM: regular moments, CNM: central moments, MDRGM: Mahalanabis distance of regular moments, MDCNM: Mahalanabis distance of central moments.

A superior performance was expected from 2-D moments, since they retain the structural property of the image. However, it turns out that a small number of 2-D moments results in poor representation of an image. To verify this, we compute the moments of the image shown in Fig. 3.3(a), and reconstruct the image using first 20, and 100 moments. The reconstructed images are shown in Fig. 5.8. Note that the global structure of the image is maintained in both cases. Fig. 5.8a shows that the middle region consists of high intensity pixels (*i.e.*, the face) and the border region consists of low intensity pixels (*i.e.*, the hair). No more detail is visible in Fig. 5.8(a) that has been reconstructed using 20 moments. Fig 5.8b, which was reconstructed using 100 moments, however shows more details. The low intensity regions of eyes, mouth and hair can be detected, although the quality is poor. We may note that the SNR's in the two cases are 11.4 and 14.2 dB, respectively. Overall, it can be inferred that while 10-20 1-D moments may be sufficient to represent the image histogram, a larger number of 2-D moments is required to represent the entire image. This increases the complexity of the indexing technique as well as the storage space requirement.

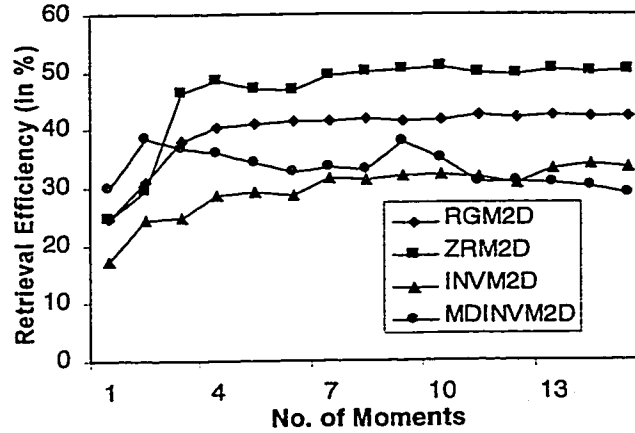


Figure 5.7. Indexing performance of various 2D moments. RGM2D: 2D regular moments, ZRM2D: 2D Zernike moments, INVM2D: 2D Hu's invariant moments, MDINVM2D: Mahalanabis distance of Hu's invariant moments.

5.1.1.5 Indexing of Texture Images

The indexing performance of histogram-based techniques presented so far, has been applied to a general image database IDB1 (see appendix B). We now evaluate the performance of histogram techniques when retrieving texture images. The set of images employed is given by the database IDB2 and is shown in Fig. B.2 (appendix B). We note that these texture images are different from the images in IDB1 in that they do not contain large uniform objects. In addition, these images show strong directional properties.

Performance of DOIH (difference of image histogram) employing L^1 metric, and MDRGM technique (Mahalanabis distance of regular moments) is shown in Fig. 5.9. It is observed that the DOIH technique provides a retrieval efficiency of approximately 72%, while MDRGM provides a performance up to 65% depending on the number of moments employed.

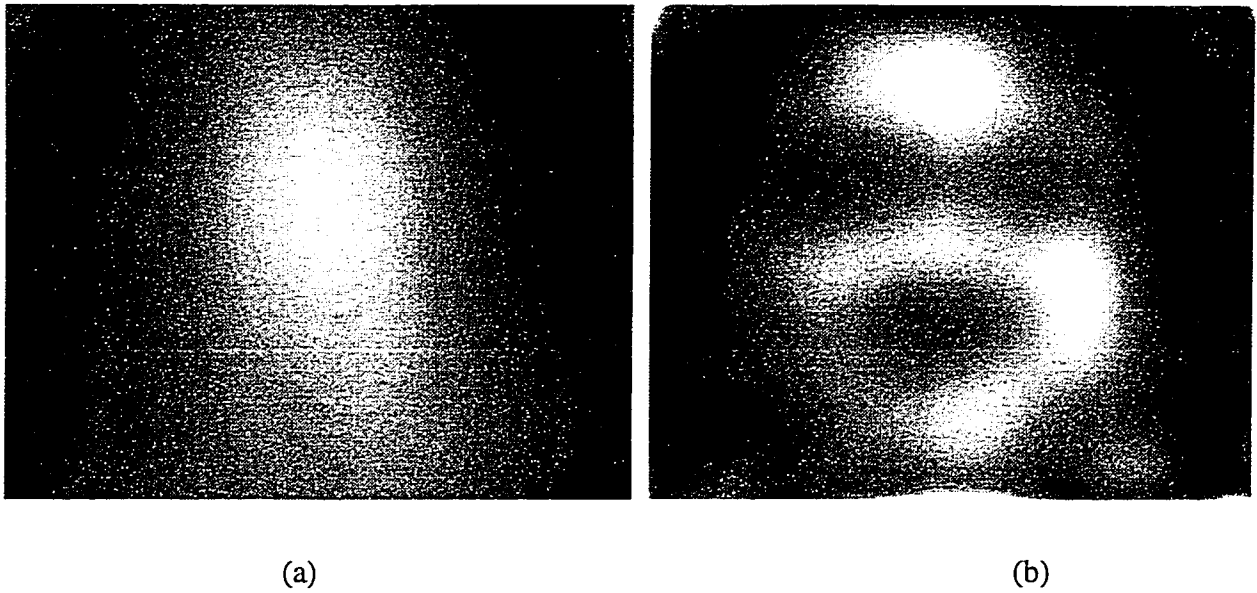


Figure 5.8. Reconstructed images of Fig. 3.3a using a) 20 and b) 100 2-D regular moments.

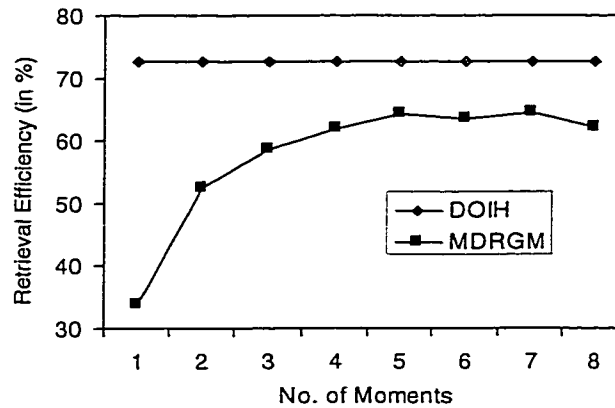


Figure 5.9. Performance of histogram techniques for indexing texture images. DOIH: Difference of image histogram, MDRGM: Mahalanabis distance of regular moments,

Although the DOIH technique provides a good performance ($\eta_R=92\%$) for IDB1-database, it does not perform well ($\eta_R=72\%$) for a texture (IDB2) database. A probable reason is that the images in IDB2-database do not have large objects. Hence, the histograms

are generally not smooth and similar textures might have large differences in their histograms. This results in a degradation of retrieval efficiency.

Two visual examples of texture image retrieval are shown in Figs. 5.10 and 5.11. In the first example, a brick texture is shown in Fig. 5.10(a) that is used as a query image. The database contains seven other similar brick textures (three of them are from the same scale, and four of them are from a lower scale). The nine retrieved images (Fig. 5.10b-j) contain one brick texture of same scale (Fig. 5.10d) and two brick textures of lower scale (Fig. 5.10b and Fig. 5.10g). In the second example, a tile texture is shown in Fig. 5.11a. The database contains three similar tile texture images. However, the histogram technique could not retrieve any of them with a retrieved list of nine images.

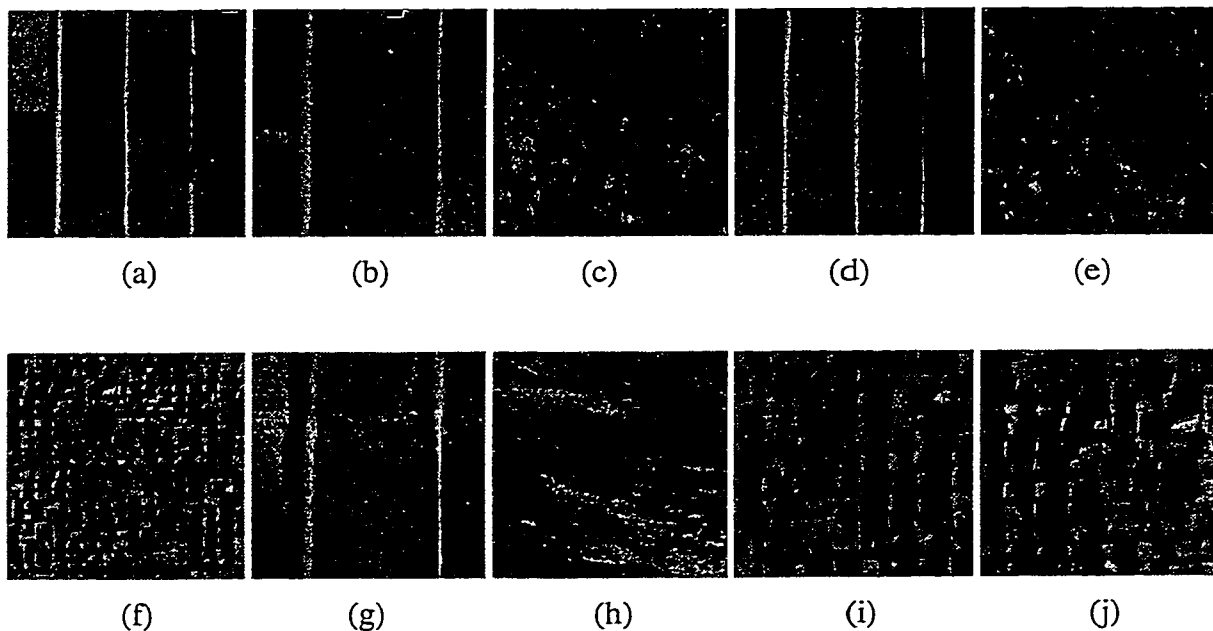


Figure 5.10. Images retrieved using Histogram technique - first query. a) query image, (b)-(j) retrieved image in monotonically increasing order of distance.

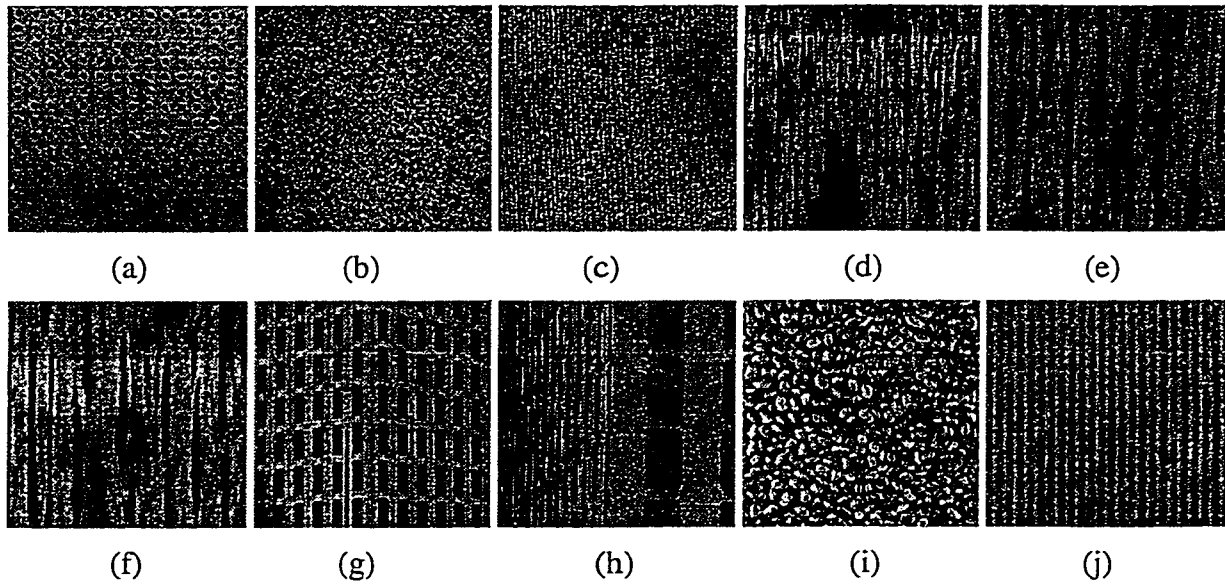


Figure 5.11. Images retrieved using histogram technique - second query. a) query image, (b)-(j) retrieved image in monotonically increasing order of distance.

5.1.2 Histogram Representation by Modified Legendre Moments

It was observed in section 5.1.1 that histogram techniques provide a good performance for indexing natural images. It was also observed that moment representation of histograms reduces the complexity of the indexing technique. However, regular moments do not provide a good indexing performance because of the nonorthogonal error space. It was further noted that indexing performance could be improved significantly by employing Mahalanabis distance of the feature vector. However, this technique is adaptive in nature and considerably increases the complexity of the overall process.

In this section, we propose to employ orthogonal Legendre moments for histogram representation [68]. We note that Legendre moments are based on orthogonal Legendre polynomials [57]. A k -th order Legendre polynomial is defined as follows.

$$P_k(x) = \frac{1}{k!2^k} \frac{d^k}{dx^k} (x^2 - 1)^k = \sum_{j=0}^k a_{kj} x^j \quad (5.2)$$

The values of a_{kj} can be expressed as [57]:

$$a_{kj} = \sum_{\substack{j=0 \\ j,k \text{ odd} \\ j,k \text{ even}}}^k (-1)^{(k-j)/2} \frac{(k+j)!}{2^k ((k-j)/2)! ((k+j)/2)! j!} \quad (5.3)$$

We note that a_{kj} is nonzero only when both k and j are either odd or even. For all other values of k and j , a_{kj} is zero. The Legendre polynomials up to 15th order are shown in Fig. 5.12. $P_k(x)$ can be expressed as a k th order polynomial as shown in Eq. 5.2. We note that the Legendre polynomials $\{P_k(x)\}$ form an orthogonal basis set in the interval $[-1,1]$.

$$\int_{-1}^1 P_m(x) P_n(x) dx = \frac{2}{2m+1} \delta_{mn} \quad (5.4)$$

The k -th Legendre moment of a function $f(x)$ is defined as

$$\Lambda_k = \frac{2k+1}{2} \int_{-\infty}^{\infty} P_k(x) f(x) dx \quad k \geq 0, k \in \mathbb{Z} \quad (5.5)$$

Using Eq. 5.2, Λ_k can be expressed as:

$$\Lambda_k = \frac{2k+1}{2} \sum_{j=0}^k a_{kj} M_j \quad (5.6)$$

where M_j is the j th order regular moment. It is observed that the Legendre moment of any order, depends only on the regular moments of the same order and lower, and can be calculated easily using Eq. (5.6).

Generally, the image gray levels range from 0 to 255. Since, the Legendre polynomials are orthogonal only in the interval $[-1,1]$, the dynamic range of the *pdf* has to be mapped onto this interval. The *pdf* $f(x)$ can then be written as an infinite series expansion in terms of the Legendre polynomials.

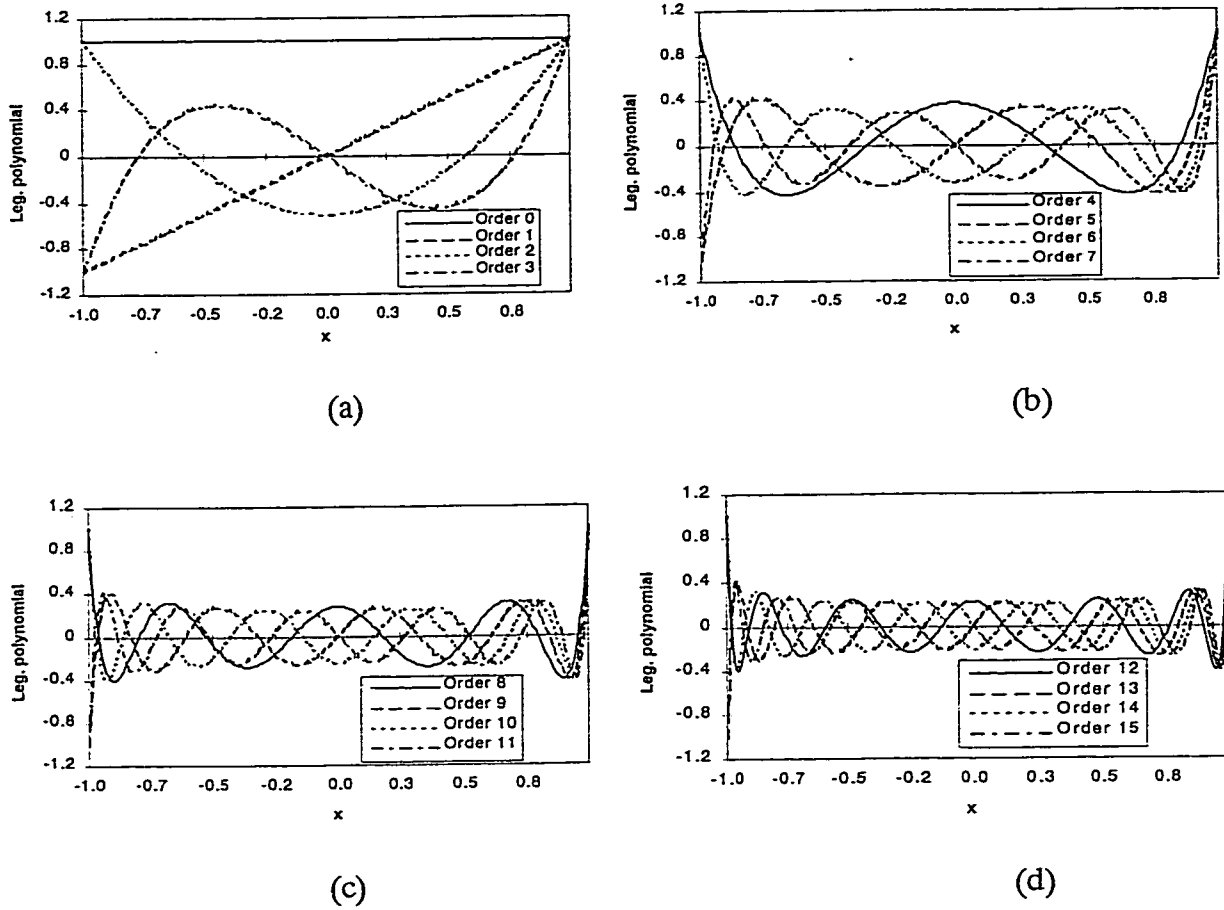


Figure 5.12. Legendre polynomials of order 0-15. a) order 0-3, b) order 4-7, c) order 8-11, d) order 12-15.

$$f(x) = \sum_{k=0}^{\infty} \Lambda_k P_k(x) \quad (5.7a)$$

The reconstructed *pdf* using the first $N + 1$ moments can be expressed as follows:

$$f'(x) = \sum_{k=0}^N \Lambda_k P_k(x) \quad (5.7b)$$

Distance Function

Consider two *pdf*'s $f(x)$ and $g(x)$ and their Legendre moments $\{\Lambda_{kf}\}$ and $\{\Lambda_{kg}\}$. The reconstructed *pdf*'s using the first $N+1$ moments are as follows:

$$f'(x) = \sum_{k=0}^N \Lambda_{kf} P_k(x)$$

$$g'(x) = \sum_{k=0}^N \Lambda_{kg} P_k(x)$$

The L^2 distance between two reconstructed *pdf*s can be expressed as

$$E = \int_{-1}^1 [f'(x) - g'(x)]^2 dx$$

$$= \sum_{k=0}^N \frac{2(\Lambda_{kf} - \Lambda_{kg})^2}{2k+1}$$

$$= \sum_{k=0}^N (\lambda_{kf} - \lambda_{kg})^2 \quad (5.8)$$

where we define the modified Legendre moments as:

$$\lambda_k = \sqrt{(2k+1)/2} \sum_{j=0}^k a_{kj} M_j \quad (5.9)$$

Thus, the L^2 distance between two reconstructed *pdf*s can be calculated from the L^2 distance of the modified Legendre moments. In the proposed technique (henceforth referred to as LGM technique), the modified Legendre moments of the histogram of the query image are calculated first. These are then compared to the corresponding moments of the images stored in the database. The images whose modified Legendre moments have the least L^2 -

distance are retrieved from the database. We note that we have chosen to use modified Legendre moments over Legendre moments to reduce the computational complexity (the performance remains the same).

Performance of Legendre Moments

The retrieval efficiency of regular and Legendre moments is compared in Fig. 5.13. It is observed that the L^2 distance of Legendre moments provide a substantial improvement over L^2 distance of regular moments. We also note that the performance of Legendre moments is very close to that obtained by employing Mahalanabis distance of regular moments. However, the former technique involves a lower computational complexity.

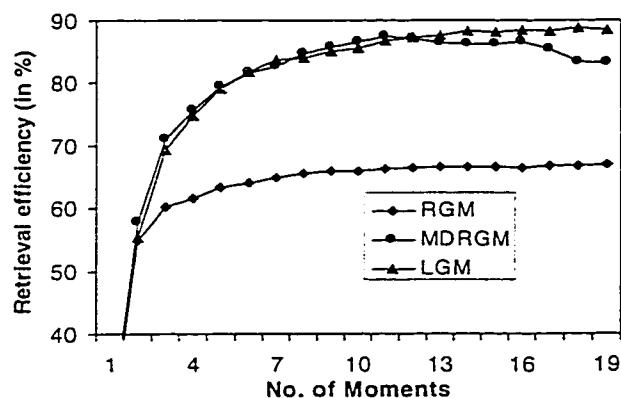


Figure 5.13. Performance comparison of regular and Legendre moments for indexing (based on IDB1 image database). RGM: regular moments, MDRGM: Mahalanabis distance of regular moments, LGM: modified Legendre moments.

5.1.3 Comparison of Computational Complexity

In this section, we provide an estimate of the computational complexity of the various histogram- and moment-based techniques. There are two types of complexities involved - i) complexity of feature vector generation (ξ_G), and ii) complexity of feature vector

comparison (ξ_C). We note that the feature vector is generated only once, and stored along with each image. On the other hand, the complexity of feature comparison is involved each time a retrieval is performed. We note that only the arithmetic operations (namely addition, subtraction, multiplication, and division) have been considered while computing the complexity. The followings notations have been assumed - $N_R \times N_C$: dimension of an image, N_G : No. of histogram gray levels, N_M : No. of moments, S : No. of images in the database.

Table 5.4

Computational complexity of feature generation (ξ_G) for various indexing techniques. The values are given in operations/image.

Operations	Complexity	Typical Value ⁺
Histogram	$N_R N_C$	6.5e+4
Histogram + Regular Moments	$N_R N_C + 2N_M N_G$	7.1e+4
Histogram + Central Moments	$N_R N_C + 3N_M N_G$	7.1e+4
Histogram + Legendre Moments	$N_R N_C + 2N_M N_G + N_M^2 / 2$	7.1e+4
Covariance Matrix	$\approx S N_M^2$	7.2e+4
Histogram + Regular Moments + Covariance Matrix	$N_R N_C + 2N_M N_G + S N_M^2$	1.4e+5

⁺for $N_R=256$, $N_C=256$, $N_G=256$, $N_M=12$, $S=500$

The computational complexity of feature generation is tabulated in Table 5.4. Entries in the first row provide the complexity of computing histograms of an image. Note that only the complexity of histogram bin-count increments has been considered. Entries in row 2-3

show that the overhead of computing regular and central moments (from histograms) is $2N_M N_G$ and $3N_M N_G$, respectively. Legendre moments are computed by multiplying a triangular matrix a_{kj} of dimension $N_M \times N_M$ with the regular moments (see Eq. 5.6). However, approximately half of the matrix coefficients of the triangle (a_{kj}) are equal to zero. Hence, the overhead of computing Legendre moments from the regular moments is approximately $N_M^2/2$. The complexity of computing covariance matrix from S , N_M -dimensional feature vectors is approximately SN_M^2 . The last row provides the total complexity of generating histogram, moments and their covariance matrix, which is the complexity of the MDRGM technique. It is observed from Table 5.4 that the overhead for generating moments from histogram is very small, and can be ignored. However, computing the covariance matrix in order to improve the performance increases the complexity by almost 100%.

Table 5.5

Computational complexity of feature comparison (ξ_C) for various indexing techniques. The values are given in operations/image.

Operations	Complexity	Typical Value ⁺
Histogram	$2N_G$	512
Regular Moments	$2N_M$	24
Mahalanabis Distance of Regular Moments	$2N_M^2 + 3N_M$	324
Legendre Moments	$2N_M$	24

⁺for $N_G=256$, $N_M=12$

The complexity of feature comparison is shown in Table 5.5. Entries in row 2 and 4 show that the run-time complexity of computing L^2 distance (see section 3.6) of regular and Legendre moments, respectively. It is observed that the complexity is significantly lower than histogram techniques. The complexity of computing Mahalanabis distance is seen to be substantially higher (due to the multiplication of covariance matrix) than other techniques. It was shown in section 5.1.2 that L^2 distance of Legendre moments (LGM) provides a performance similar to that of Mahalanabis distance of the regular moments. Hence, LGM technique is more suitable for real time implementation due to its lower complexity.

5.1.4 Section Summary

In summary, we considered histogram-based techniques for image indexing in this section. A detailed performance analysis of histograms and moments in indexing applications is presented in section 5.1.1. In section 5.1.2, we have proposed Legendre moments of histogram (LGM) as a feature in indexing. It was shown that the performance LGM is almost identical to that of Mahalanabis distance of regular moments (MDRGM) which is optimal in the statistical sense. The complexity analysis in section 5.1.3 reveals that LGM has a lower complexity compared to MDRGM and is hence more suitable for real-time implementation.

5.2 Image Indexing in Wavelet Domain

In section 5.1, we have considered histogram-based techniques for image indexing. We note that a given histogram provides global information about its image in the spatial domain. Employing transform domain features in indexing is expected to enhance the

performance. In this section, we address the issue of image indexing in the wavelet domain. In section 5.2.1, we critically evaluate a few selected techniques in the wavelet domain, and propose suitable modifications to improve the performance. In section 5.2.2, we propose an indexing technique based on the distribution of wavelet coefficients. In section 5.2.3, we propose a joint moment and wavelet technique.

5.2.1 Performance of Indexing Techniques based on DWT Coefficients

Several wavelet-based indexing techniques [16, 48, 87, 92, 103, 122] have been discussed in section 3.2.2. These techniques may be divided into two categories. The techniques in the first category [16, 48, 103, 122] extract indexing features from the DWT coefficients obtained by a dyadic decomposition of images. The techniques in the second category [87, 92] employ different kinds of wavelet transform (*e.g.*, complex wavelets) specially suited for extracting better indexing features. We note that the techniques in the second category generally have a higher complexity. In addition, the coding efficiency of these wavelet transforms is yet to be evaluated. In this section, we consider selected techniques from the first category [48, 103] for further analysis where the techniques chosen have feature vectors of small dimension, and thus have lower complexity.

It was observed in section 3.2.2 that the technique proposed by Jacobs *et al.* [48] compares the location of high magnitude DWT coefficients while the technique proposed by Wang *et al.* [122] compares low-resolution DWT coefficients. The first technique can be viewed as a simpler version of the second technique, which requires a lower dimension feature vector. Hence, we have chosen Jacobs *et al.*'s FMIQT for performance evaluation. Smith *et al.* [103] have employed a different technique where multi-band wavelet image is first generated by upsampling each highpass wavelet-band, appropriately (Fig. 3.10). A multi-dimensional

histogram is calculated which is then employed as a feature vector for indexing. As a variant of wavelet-based technique, we also choose Smith's wavelet histogram technique (WHT) for further investigation.

5.2.1.1 Fast Multiresolution Image Querying Technique (FMIQT)

FMIQT technique has been evaluated on both IDB1 and IDB2 databases. The sign (positive and negative) and indices of 240 largest magnitude DWT coefficients of each image are calculated. We note that 240 coefficients have been chosen since the image size (256x256) is approximately four times the size (128x128) of the image employed in [48]. These indices are then employed for indexing. The performance of the FMIQT is shown in Table 5.6. It is observed that the retrieval efficiency of FMIQT, on IDB1 and IDB2, is poor compared to the DOIH technique discussed in section 5.1.1.

Table 5.6

Retrieval efficiency of FMIQT [48] and WHT [103] techniques L^1 metric has been employed for distance calculation.

	General Image (IDB1)	Texture Image (IDB2)
FMIQT	26.47%	20.15%
WHT	70.11%	85.20%
DOIH	92.3%	72%

The main reason for the failure of FMIQT technique is that it assumes the location of the high magnitude DWT coefficients to be invariant for similar images. This is not a valid assumption in general and also in the case of database of images employed in the experiment. The location of high magnitude DWT coefficients may be different due to several reasons,

including translation and rotation in pixel domain. In addition, aliasing due to filtering and decimation as a result of wavelet decomposition is also likely to degrade the performance. An example is shown in Fig. 5.14. The locations of high magnitude DWT coefficients of Fig. 3.3a and Fig. 3.3b are shown in Fig. 5.14a, and Fig. 5.14b, respectively. Fig. 5.14c shows how the locations of the DWT coefficients match in Fig. 5.14a and Fig. 5.14b. It is observed that only a small percentage of locations (represented by white region) matches while a large percentage of locations (represented by black region) does not match. Hence, FMIQT is not expected to perform well in the present case. We note that the performance on IDB1 database is superior compared to the IDB2 database. This is expected since the images in IDB1 are more structured and uniform. As a result, the location of coefficient-peaks will be closer, resulting in an improved performance.

5.2.1.2 Wavelet Histogram Technique (WHT)

The performance of wavelet histogram technique (see section 3.2.2 for details) on IDB1 and IDB2 databases is tabulated in Table 5.6. A wavelet histogram (WH), consisting of 512 bins, is generated from nine (for 3 stage decomposition) highpass wavelet bands. The WH of the query image is compared with the WHs of the target images in the database for retrieval. It is observed that the wavelet histogram technique (WHT) provides a performance of 85.20% on the IDB2 database. This performance is superior to that provided by FMIQT. The retrieval efficiency on the IDB2 database is also superior compared to the performance of histogram-based techniques discussed in section 5.1.1. This is because the images in IDB2 have a strong directional property that is captured by the WHT. On the other hand, the histogram-based techniques capture only the global description of an image in the spatial domain and are hence not efficient.

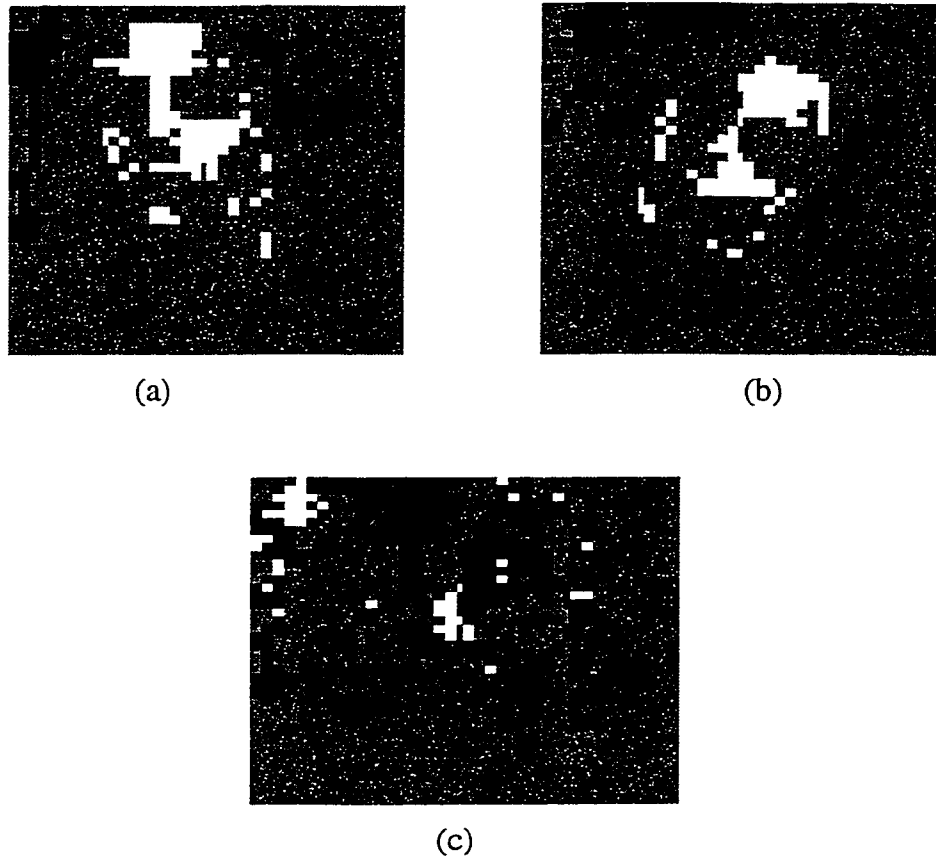


Figure 5.14. Location of high amplitude DWT coefficients. (a) and (b) show the relative amplitudes of Fig. 3.3a and Fig. 3.3b, respectively. White, gray, and black regions correspond to high amplitude positive, low amplitude, and high amplitude negative DWT coefficients, respectively. The high amplitude coefficients are confined to the level-3 regions. Hence, a window of 40x40 (out of 248x256 coefficients) has been shown in these figures. (c) shows the matching of (a) and (b) employing Jacob's algorithm. White and black regions correspond to matched and mismatched regions. Both images have small DWT coefficients in the gray region. Hence, this region is ignored.

For the general database IDB1, the WHT provides a performance of 70.11%. Although, this performance is superior compared to FMIQT, it is inferior to the performance of DOIH techniques, which is on the order of 90-93%. This also seems justifiable, since the general image database contains images with a few large objects. The histogram of the overall image is generally influenced by these objects. Hence, histograms of images containing similar

objects are very similar. On the other hand, directional information is less prominent in these images. Hence, WHT provides a lower retrieval efficiency.

It is clear from the above discussions that WHT is an efficient technique for indexing images with strong directional properties. We now address the issue of the computational complexity, which is shown in Table 5.7. We note that WH generation is a complex process. A comparison of Table 5.7 with Table 5.4 reveals that the complexity of WHT is higher compared to spatial domain histogram techniques. The higher complexity is mainly due to the fact that all highpass bands are to be upsampled, and thresholded. In addition, the multi-channel nature of the entire operation increases the complexity by a factor of N_B , the total number of channels (or bands). In the following, we propose techniques to reduce the complexity while maintaining the indexing performance.

5.2.1.3 Fast Wavelet Histogram Techniques (FWHT)

The WHT [103], as discussed in the previous section, upsamples and filters all highpass wavelet subbands such that each upsampled band contains the same number of coefficients as present in the original image. In other words, the level- k bands (see Fig. 2.9 for the definition of level) are upsampled by a factor of $2^k \times 2^k$. We note that there is a substantial amount of redundancy present in the upsampled and subsequently filtered images. Hence, there is a potential to reduce this redundancy without degrading the performance.

In this section, we propose a fast WHT (FWHT) employing the multiresolution nature of wavelet representation. Here, various highpass wavelet bands are appropriately upsampled or downsampled to have the same number of coefficients as in level- k , $k \in \{1, 2, \dots, m\}$ for m -stage decomposition. In addition, the zeroth band (i.e., lowpass subband at level-3) is also

included in WH generation. The schematic of the FWHT technique is shown in Figs. 5.15-5.17. Fig. 5.15 shows WH generation at level-1. Here, the coefficients from level-1 bands are untouched while the coefficients from level-2 and level-3 bands are upsampled by a factor of 2x2, and 4x4, respectively. These coefficients are then thresholded and appropriate WH is generated. WH generation at level-2 is shown in Fig. 5.16. Here, the coefficients from level-1 bands are downsampled by a factor of 2x2, while the coefficients from level-3 are upsampled by a factor of 2x2. Finally, WH generation at level-3 is shown in Fig. 5.17. Here, the coefficients from level-1 and level-2 bands are downsampled by a factor of 4x4 and 2x2, respectively.

Table 5.7

Complexity of computing wavelet histogram in WHT [103]. N_B : Total number of channels (bands) to be upsampled and filtered, N_P : Number of pixels ($= N_R N_C$) in an image.

Main Modules	Sub-Modules	Complexity	Typical Value ⁺
Thresholding	Absolute Value Calculation	$N_B * N_P$	5.9e+5
	Thresholding	$N_B * N_P$	5.9e+5
Histogram Calculation	Feature Point Generation ⁺⁺	$N_B * N_P$	5.9e+5
	Histogram Calculation	N_P	6.5e+4
Total	Thresholding+ Histogram Calculation	$(3N_B + 1) * N_P$	1.8e+6

⁺ for $N_P=65536$ (256x256), $N_B=9$

⁺⁺ Feature point generation refers to computing of histogram bins from the 9-D feature vector.

We note that one of the modification we have proposed above is to include the lowest frequency band (A_0). However, if we include A_0 as it is, it will make the feature vector

variant to image brightness, which is not desirable. Hence, we make this band zero-mean, before applying the thresholding. This enables us to use the lowpass information without being affected by the brightness variation which may be caused by a camera flash.

The complexity of WH generation at levels 1-3 is provided in Table 5.8-5.10. Entries in Table 5.8 are similar to those of Table 5.7, although with different weights. Table 5.9 contains two types of thresholding operations - for upsampling (thresholding-1) and downsampling (thresholding-2). Here, upsampling and filtering has been implemented by a simple pixel replication scheme whereas downsampling has been implemented as weighted average of 2x2 or 4x4 pixels. We note that the complexity of generating an output pixel by downsampling is greater compared to the corresponding complexity for generation by upsampling. Hence, the former has been given a larger weight (=4) in row 3 of Table 5.9. Comparison of Tables 5.7-5.10 reveals that the complexity of FWHT-1, FWHT-2, FWHT-3 are respectively 0.23, 0.08, and 0.04 times the complexity of WHT technique. This reduction in complexity of WH computation is mainly due to two factors: i) the reduction in the number of effective DWT coefficients for histogram calculation, ii) only 7 bands (out of 10 bands) are employed for upsampling and/or downsampling.

Although, computing the wavelet histograms at levels 1-3 reduces the complexity substantially, the complexity can be further reduced by employing only a selected number of the wavelet bands for WH computation. We note that subbands {1,4,7} provide horizontal information, subbands {2,5,8} provide vertical information, and subbands {3,6,9} provide diagonal information. Since information along a particular direction is provided by at least three subbands (although at different scales), some subbands may be ignored while

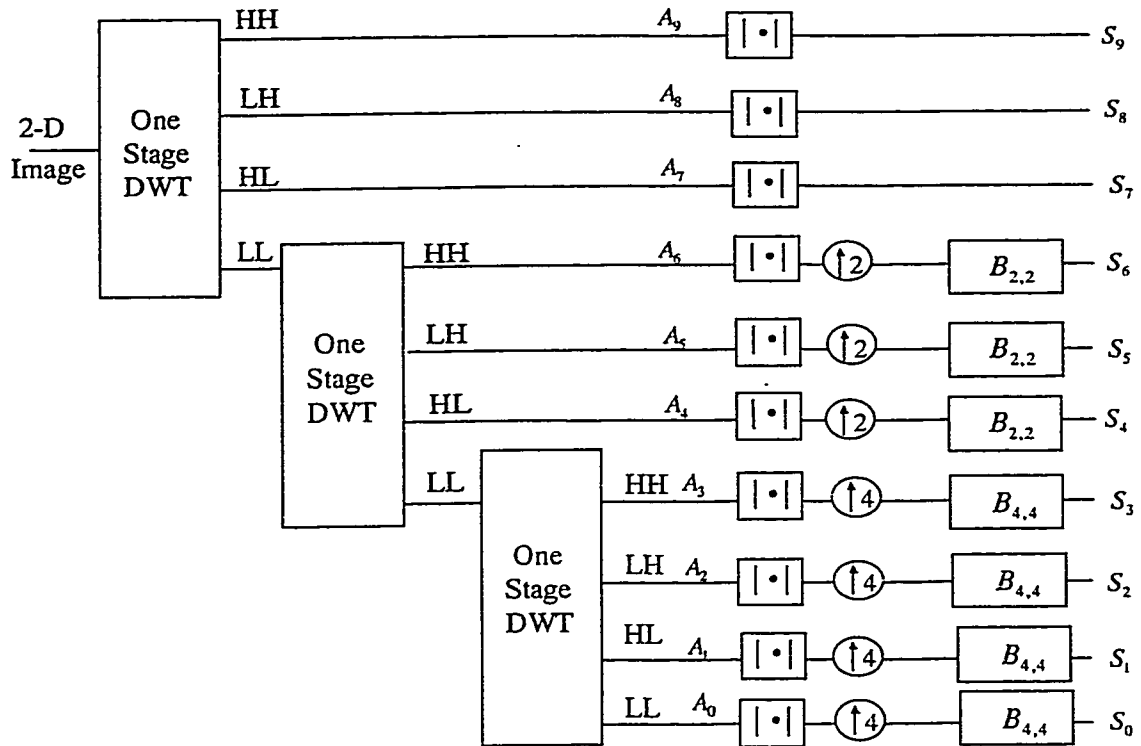


Figure 5.15. Schematic of wavelet histogram generation at level-1.
 $\uparrow_M$ indicates upsampling by 1:M both horizontally and vertically.

Table 5.8

Complexity of computing wavelet histogram in level-1. N_U : Total number of channels to be upsampled and filtered, N_B : Total number of channels, N_P : Number of image pixels.

Main Modules	Sub-Modules	No. of Operations	Typical Value ⁺
Thresholding	Absolute Value Calculation	$N_U * N_P / 4$	1.1e+5
	Thresholding	$N_U * N_P / 4$	1.1e+5
Histogram Calculation	Feature Point Generation	$N_B * N_P / 4$	1.6e+5
	Histogram Calculation	N_P	1.6e+4
Total		$\frac{(2N_U + N_B + 1)}{4} * N_P$	4.1e+5

⁺ for $N_U=7$, $N_B=10$, $N_P=65536$ (256x256)

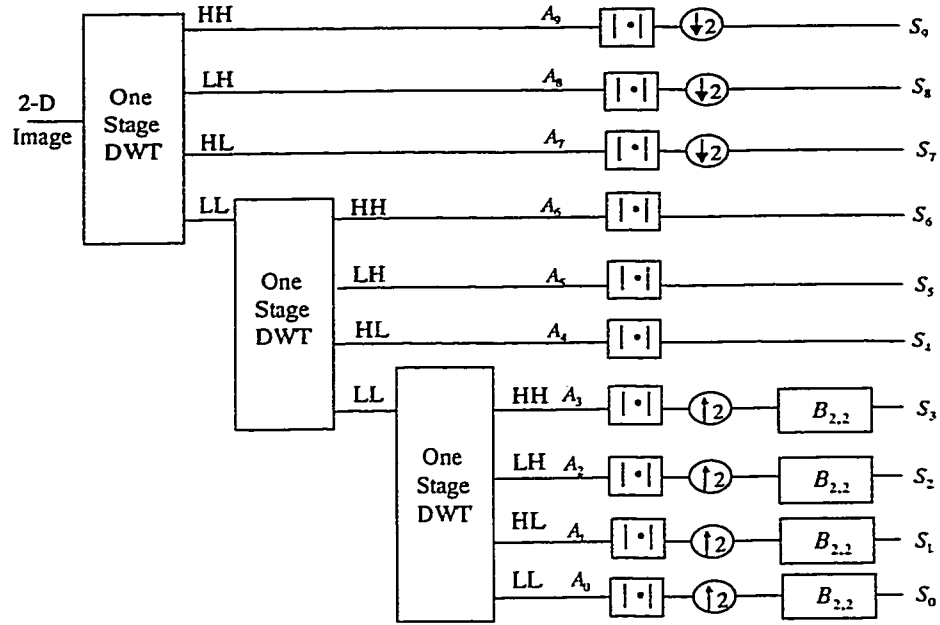


Figure 5.16. Schematic of wavelet histogram generation at level-2. $\uparrow_M$ and $\downarrow_M$ indicate upsampling and downsampling by 1:M both horizontally and vertically, respectively.

Table 5.9

Complexity of computing wavelet histogram in level-2. N_U : Total number of channels to be upsampled and filtered, N_D : Total number of channels to be downsampled, N_B : Total number of channels, N_P : Number of image pixels.

Main Modules	Sub-Modules	No. of Operations	Typical Value ⁺
Upsampling & Thresholding	Absolute Value Calculation	$N_U * N_P / 16$	1.6e+4
	Thresholding	$N_U * N_P / 16$	1.6e+4
Downsampling & Thresholding	Absolute Value Calculation	$4 * N_D * N_P / 16$	4.9e+4
	Thresholding	$N_D * N_P / 16$	1.2e+4
Histogram Calculation	Feature Point Generation	$N_B * N_P / 16$	4.1e+4
	Histogram Calculation	$N_P / 16$	4.1e+3
Total	Thresholding-1+Thresholding-2 + Histogram Calculation	$\frac{(2N_U + 5N_D + N_B + 1) * N_P}{16}$	1.4e+5

⁺ for $N_U=4$, $N_D=3$, $N_B=10$, $N_P=65536$

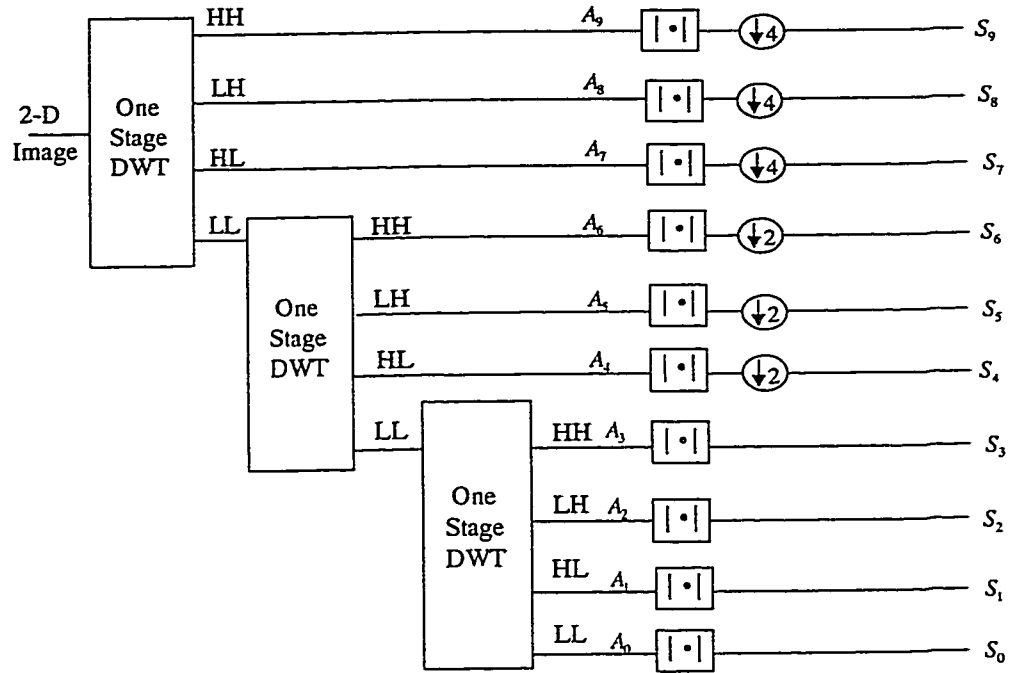


Figure 5.17. Schematic of wavelet histogram generation at level-3.

Table 5.10

Complexity of computing wavelet histogram in level-3. N_U : Total number of channels to be downsampled by 2, N_D : Total number of channels to be downsampled by 4, N_B : Total number of channels, N_P : Number of image pixels.

Main Modules	Sub-Modules	No. of Operations	Typical Value ⁺
Upsampling & Thresholding-1	Absolute Value Calculation	$4 * N_U * N_P / 64$	1.2e+4
	Thresholding	$N_U * N_P / 64$	3.0e+3
Downsampling & Thresholding	Absolute Value Calculation	$16 * N_D * N_P / 64$	4.9e+4
	Thresholding	$N_D * N_P / 64$	3.0e+3
Histogram Calculation	Feature Point Generation	$N_B * N_P / 64$	1.0e+4
	Histogram Calculation	$N_P / 64$	1.0e+3
Total	Thresholding-1+Thresholding-2 + Histogram Calculation	$\frac{(5N_U + 17N_D + N_B + 1) * N_P}{64}$	7.9e+4

⁺ for $N_U=3$, $N_D=3$, $N_B=10$, $N_P=65536$

Table 5.11

Complexity of FWHT at level-2. *No. of Bands* (N_B) includes the zeroth band. N_U : Total number of channels to be upsampled, N_D : Total number of channels to be downsampled, N_P : Number of image pixels.

No. of Bands	Complexity (OP/image)	Typical Value ⁺
7	$\frac{(2N_U + N_B + 1)}{16} * N_P$	7.8e+4
10	$\frac{(2N_U + 5N_D + N_B + 1)}{16} * N_P$	1.4e+5

⁺ for $N_U=4$, $N_D=3$, $N_B=10$, $N_P=65536$

calculating the WH histogram. Table 5.11 compares the complexity of WH generation at level-2 employing 7 and 10 bands. It is observed the complexity for the case of 7 bands is approximately half the complexity of the 10-band case.

Wavelet histograms at various levels are shown in Fig. 5.18 and Fig. 5.19. Fig. 5.18 shows wavelet histograms of the image shown in Fig. 3.3a while Fig. 5.19 shows wavelet histograms of the image shown in Fig. 5.11a. It is observed that wavelet histograms generally tend to have high peaks at a regular interval. The histograms are sparse at higher levels because of lower number of feature points. In addition, the number of bins increases from 512 (because of 9-bands) to 1024 (because of 10-bands) when the zeroth band is included resulting in a sparser histogram. We note that when the histograms are sparse, their indexing performance may not be robust. One way of reducing the sparseness of the wavelet histograms is to use fewer subbands for generating the histograms. Fig. 5.18h and Fig. 5.19h show the wavelet histogram at level-2 from seven wavelet bands (*i.e.*, 128 histogram bins).

The histograms appear to be smooth and dense. In addition, the complexity of generating and comparing the feature vectors is smaller compared to nine/ten bands cases. However, the information loss resulting from not using three subbands may lead to a marginal degradation in indexing performance.

Detailed experiments are carried out to evaluate the performance of FWHT. Both IDB1 and IDB2 databases were employed for this purpose and the results are shown in Fig. 5.20. For IDB2 database, we observe the following: i) best performance is achieved at level-2, ii) WH generated with all wavelet bands provide the best performance, and iii) the inclusion of zeroth band does not seem to influence the retrieval efficiency. On the other hand, for IDB1 database, we observe the following: i) best performance is achieved at level-1, ii) best performance is achieved around 7 bands, and iii) the inclusion of zeroth band significantly improves the retrieval efficiency.

The above observations are expected because of the properties of the images in the two databases. Images in IDB2 database have strong directional features and hence employing all highpass subbands provides the best result. Inclusion of the zeroth band is not important for this database. However, for IDB1 database, the zeroth subband provides crucial information about the low frequency texture information. Hence, inclusion of this band is very important. In addition, the subbands at the highest resolution are not crucial for the images in IDB1. The directional information achieved by a few subbands of lower resolution is generally sufficient.

The above experimental results suggest that the complexity of the wavelet histogram technique can be reduced significantly while retaining (or even improving) the performance level. Table 5.12 shows the complexity and performance of WHT technique and FWHT

technique for IDB1 and IDB2 database. It is observed that for IDB2, a complexity reduction by a factor of 10 is possible without degrading the performance. On the other hand, for IDB1 database, a complexity reduction factor of 30 is possible with maintaining the performance (actually, in the present case, the performance improves marginally).

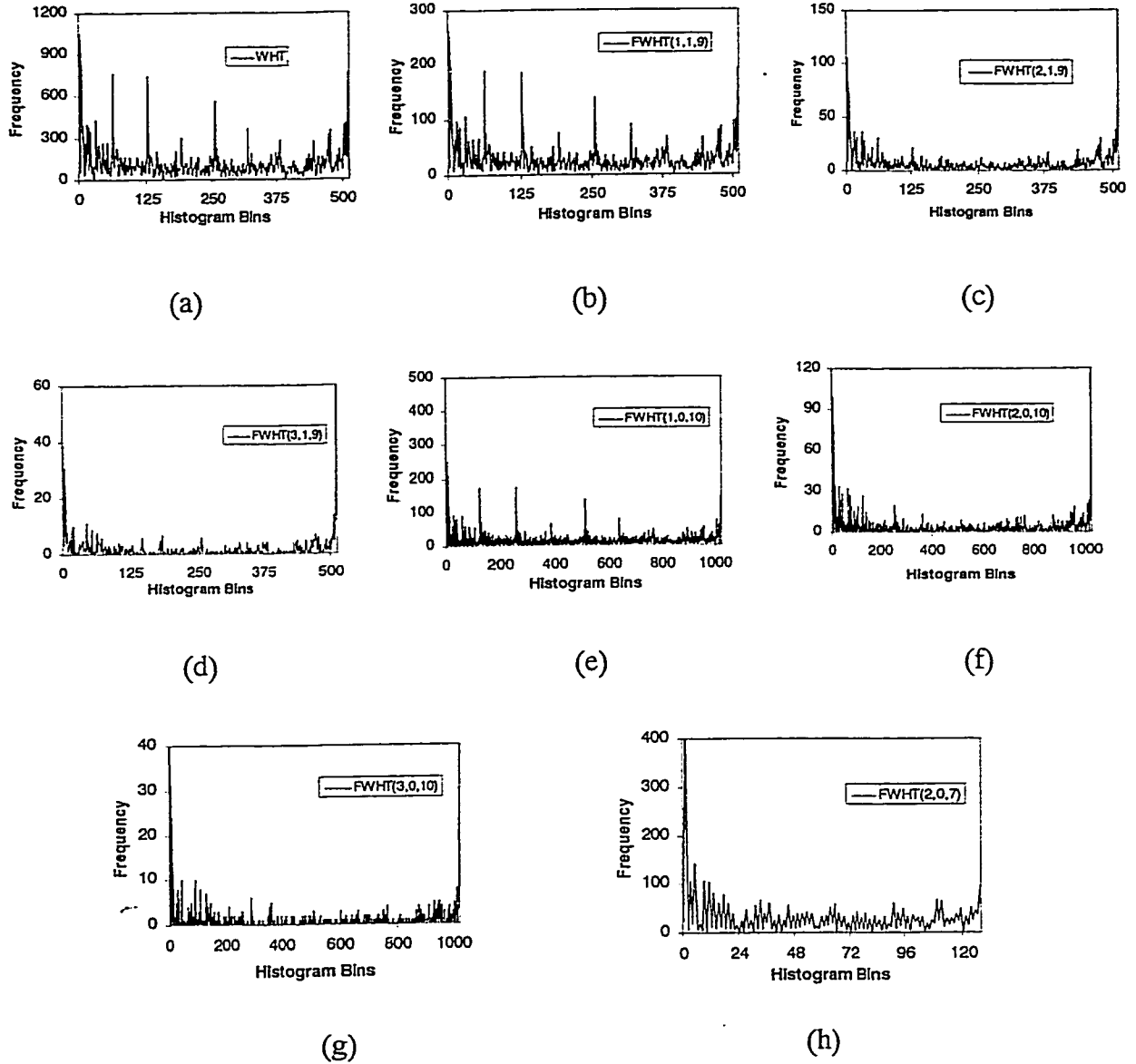


Figure 5.18. Wavelet histograms of the image shown in Fig. 3.3a (from IDB1). a) wavelet histogram proposed by Smith. (b)-(h) fast wavelet histograms at various levels. $FWHT(p,q,r)$ refers to WH calculated at level- p using subbands $\{A_q, \dots, A_{q+r-1}\}$.

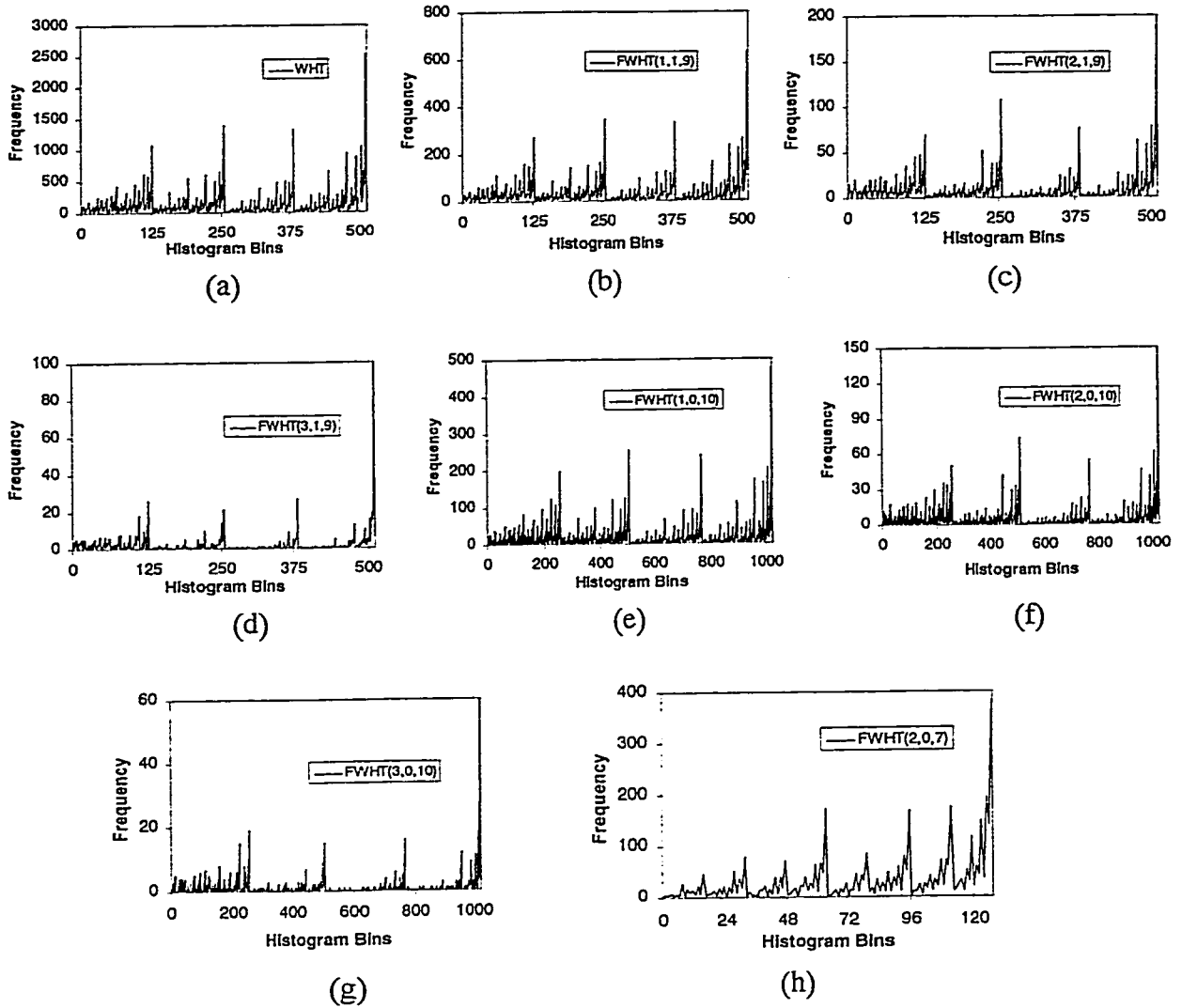


Figure 5.19. Wavelet histograms of the image shown in Fig. 5.11a (from IDB2). a) wavelet histogram proposed by Smith. (b)-(h) fast wavelet histograms at various levels. $FWHT(p,q,r)$ refers to WH calculated at level- p using subbands $\{A_q, \dots, A_{q+r-1}\}$.

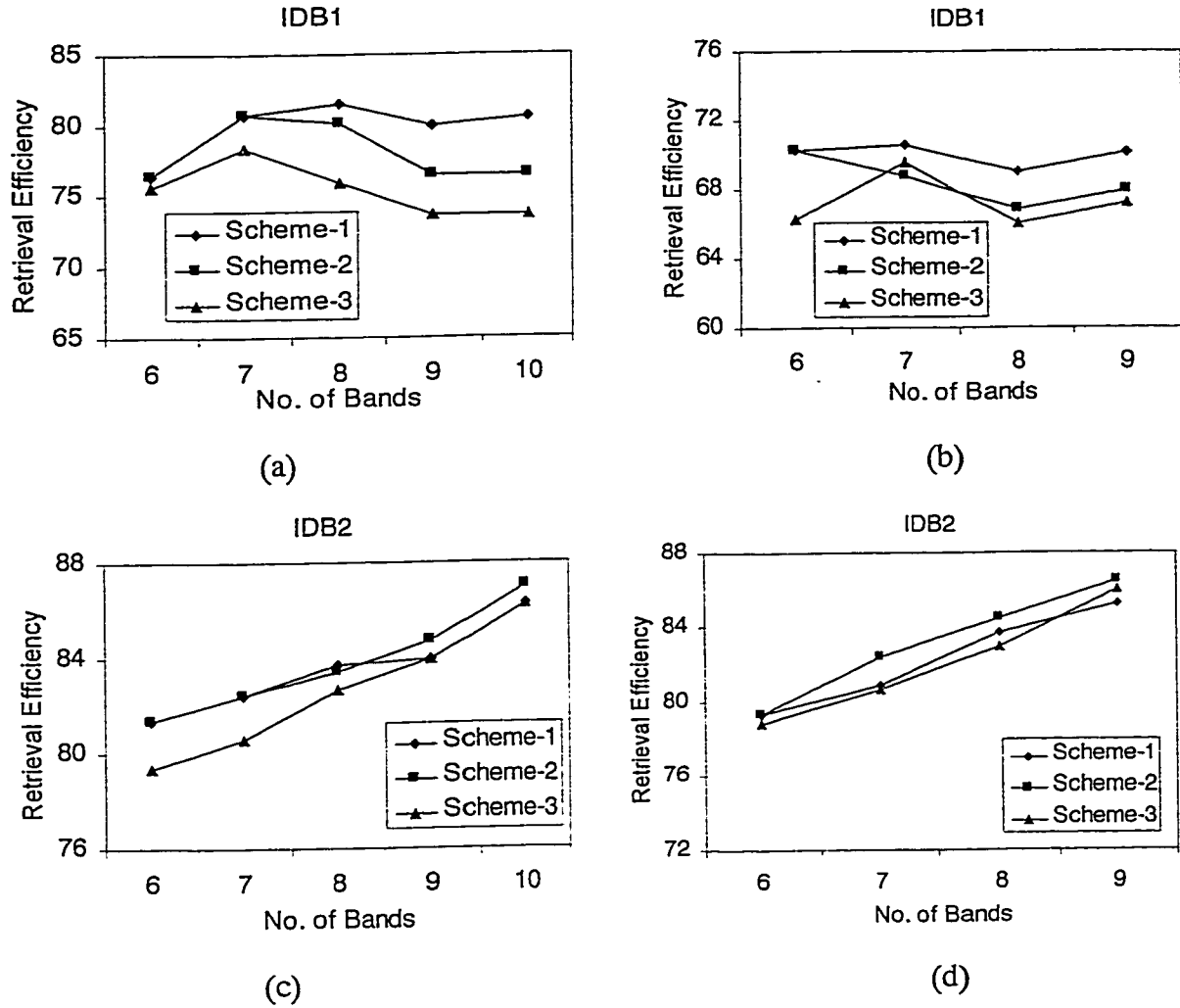


Figure 5.20. Comparison of indexing performance of various wavelet-histogram techniques. Scheme-P: DWT coefficients of all bands have been appropriately upsampled or downsampled to have same number of coefficients as in scale P. No. of bands=Q means that first Q bands (see Fig. 2.8 for nomenclature) have been employed for wavelet histogram generation. (a) & (c): No. of Bands includes zeroth band, (b) & (d): Zeroth band is not included.

We note that the complexities described so far in this section refer to the complexity of the feature generation (namely, the wavelet histogram). The runtime complexity of the FWHT is similar to that of any histogram-based technique. The complexity of comparing histograms in a straightforward manner is proportional to the number of histogram bins.

5.2.2 Image Indexing based on Distribution of Wavelet Coefficients

Various histogram-based techniques were discussed and evaluated in section 5.1.1. In addition, several indexing techniques based on wavelet coefficients were presented in section 5.2.1. It was observed in section 5.1.1 that histogram-based techniques are well suited for indexing translated and rotated images with large objects. On the other hand, experimental results, presented in section 5.2.1, showed that wavelet coefficients provide good discrimination capability based on directional information. In this section, we propose to combine these two approaches where the histograms of various subbands are used as an index.

Table 5.12

Performance versus complexity of WHT and FWHT. The proposed technique employs 10 bands for IDB2, and 7 bands for IDB1 (Zeroth band is included in both cases). Histogram is generated at level-2.

Wavelet Histogram Technique	IDB1		IDB2	
	Complexity (OP/image)	Efficiency (in %)	Complexity (OP/image)	Efficiency (in %)
WHT	1.6e+6	70.11	1.8e+6	85.20
FWHT	6.3e+4	80.68	1.3e+5	86.99

The basic assumption of histogram-based techniques is that the *pdfs* of similar images with small camera operation are similar. The same assumption can also be extended to wavelet subimages. The *pdfs* of wavelet bands of similar images, with small camera operation, will also be similar. The images can be discriminated by the amount of horizontal, vertical, and diagonal information at different scales. Different images might have similar overall histograms, but they are unlikely to have similar band statistics. Fig. 5.21 shows the

two similar histograms (of Fig. 5.1c and Fig. 5.1e) and three of their band histograms. It is clear that even though the original histogram is quite similar, the band histograms are different. The performance of the basic histogram technique can thus be enhanced by exploiting this directional information.

In the proposed technique, we compare the histogram of all subbands. However, the complexity of directly comparing all histograms is very high, and hence, this technique is impracticable. We propose to reduce the computational complexity by modeling the histograms. It was noted in section 4.1.1.3 (Eq. 4.3-4.5) that the *pdfs* of highpass wavelet subimages can be modeled using generalized Gaussian density (GGD) function [12]. For example, the *pdf* of a highpass wavelet subband (corresponding to the *lady* image shown in Fig. 5.21b) is shown in Fig. 5.22. It is observed that the *pdf* can be represented well by a GGD function with $\sigma=9.91$, $\gamma=0.58$. Hence, instead of comparing *pdfs* of highpass subbands directly, two parameters from each highpass band are compared which results in a significant reduction in complexity.

The band parameters of all the wavelet subbands of Fig. 5.1c and 5.1e are shown in Fig. 5.23. It is observed that the variances (or energy) of the vertical bands of the owl are higher compared to the lady image. This is expected because there is no appreciable change in the vertical direction of Fig. 5.1c (except at hair-forehead, eyes, and lips region). Overall, the band parameters of the lady image are very different from those of owl image. We note that the histogram of the lowpass band of the lowest resolution (S_8 subimage) cannot be modeled using a GGD function. Hence, histogram of this band has to be compared directly.

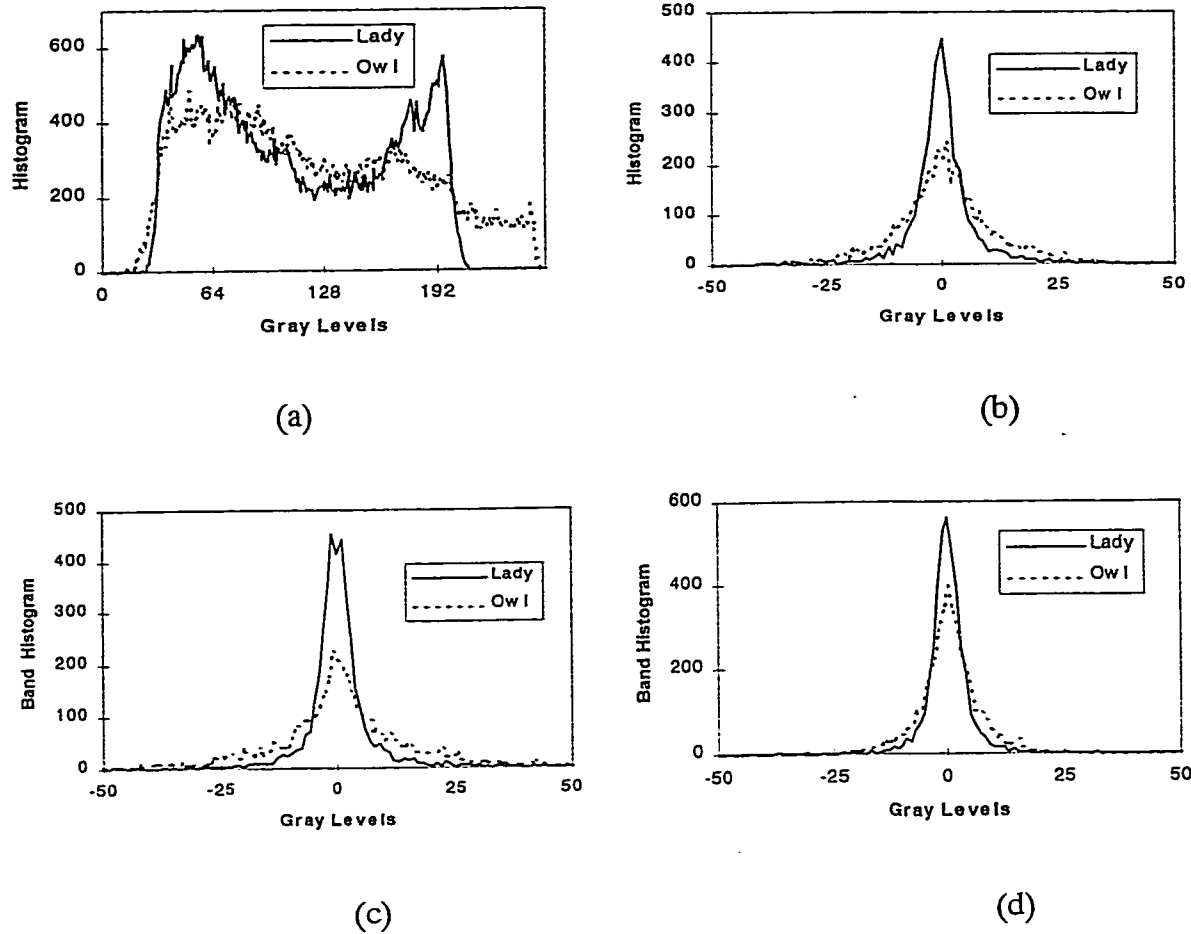


Figure 5.21. Comparison of band histogram of two different images (Fig. 5.1c and Fig. 5.1e). a) Original Histogram, b) Histogram of Band W_4^H (see Fig. 2.8), c) Histogram of Band W_4^V , d) Histogram of Band W_4^D .

The advantage of employing wavelet parameters is that these parameters are readily available from the wavelet coder employed in the thesis. Hence, they need not be recalculated which reduces the complexity of the algorithm substantially.

To summarize, in the proposed technique [68], henceforth referred to as WP (highpass wavelet parameters), we compare the (σ, γ) parameters of the wavelet bands corresponding to the query (Q) and a candidate (C) image using the following equation.

$$d_{WP}(Q, C) = \sum_{k=1}^{N_{WB}} A_k (\sigma_{Q_k} - \sigma_{C_k})^2 + \sum_{k=1}^{N_{WB}} B_k (\gamma_{Q_k} - \gamma_{C_k})^2 \quad (5.10)$$

where, N_{WB} is the number of wavelet bands used for comparison. A_k (weights of σ_k) and B_k (weights of γ_k) are estimated by a trial and error procedure (a set of typical values are provided in Table 5.14) to achieve the best performance. The images that have minimum distance are retrieved from the database.

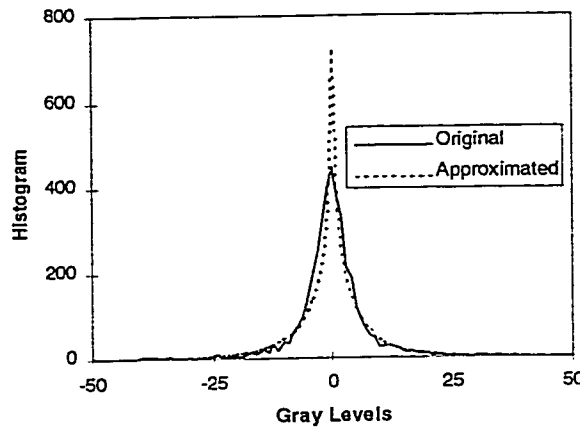


Figure 5.22. Typical histogram of a highpass wavelet subband (histogram of *lady* image shown in Fig. 5.1c) and its approximation using Generalized Gaussian density function $\sigma=9.91$, $\gamma=0.58$.

The retrieval efficiency (η_R) of the wavelet band histograms is shown in Table 5.13. The images were decomposed using Daubechies 8-tap wavelet for 3 dyadic stages; the histograms were compared in L^1 metric. The direct comparison of image histograms (*i.e.*, employing DOIH technique) provides a η_R of 92.27%. The comparison of histogram of S_8 subimage (DOIHL) provides a η_R of only 67.16%. The lower retrieval efficiency of lowpass subband (S_8) can be attributed to its noisy behavior. For example, the histogram of the S_8 subimage of Fig. 3.3a is shown in Fig. 5.24b. It is observed that the histogram is noisy compared to histogram of the full resolution image, which is shown in Fig. 5.24a.

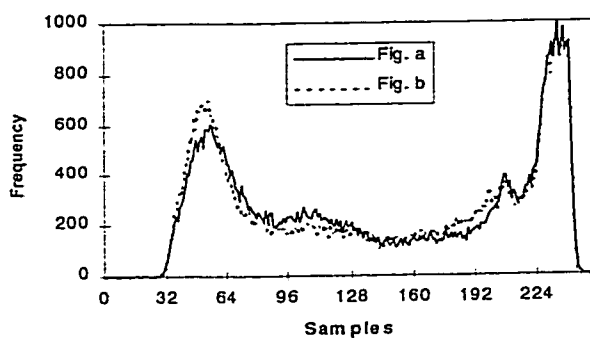
	11.74/ 16.81	9.91/11.6	9.67/5.05
11.11/ 21.63	5.55/ 11.73		
7.63/18.0		4.91/6.67	
7.09/12.41			5.13/2.29

(a)

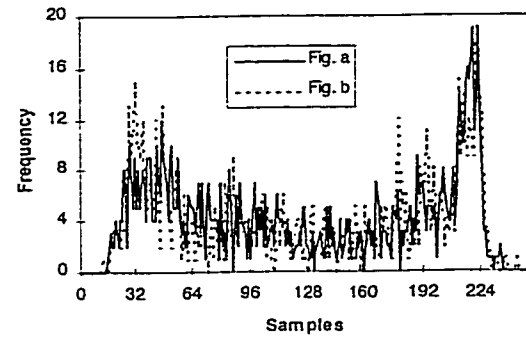
	0.64/1.05	0.58/0.97	0.65/0.70
0.55/0.91	0.69/0.98		
0.62/0.75		0.74/0.97	
0.70/0.56			0.65/1.03

(b)

Figure 5.23. Comparison of subband parameters of Fig. 5.1c and Fig. 5.1e. a) the standard deviation (σ), b) the shape parameter (γ). The parameters are shown in "x/y" form, where x refers to the parameters from Fig. 5.1c and y refers to the parameters from Fig. 5.1e. The subbands have been formed according to Fig. 2.8.



(a)



(b)

Figure 5.24. Histogram of the original image and the corresponding lowpass subimage (S_8) of Fig. 3.3a and 3.3b. a) full resolution image, b) Lowpass subimage.

The comparison of histograms of nine (for 3 level decomposition) highpass subimages (DOHHWB) provides a η_R of 59.09%. This shows that it is possible to retrieve images by comparing only the directional information at various scales (although the retrieval rate is not very high). Therefore, we expect a higher η_R by comparing the image histogram as well as histograms of wavelet bands (DOIWH). It is observed that an improvement of 2% can be

achieved by using DOIWH (with $N_{WB} = 3$) compared to using DOIH. If the wavelet band parameters, instead of band histograms, are compared (DOIHWP), an improvement of 3% is obtained. A higher value of N_{WB} improves η_R only marginally, and has hence not been employed.

Table 5.13

Performance Improvement using wavelet band information

Indexing Technique	η_R (in %)
DOIH	92.27
DOIHL	67.16
DOHHWB	59.09
DOIWH ($N_{WB} = 3$)	94.09
DOIHWP ($N_{WB} = 3$)	95.45

5.2.3 Joint Moment and Wavelet Technique

We recall from section 5.1.1 that substantial reduction in complexity can be achieved by comparing moments instead of comparing direct histograms. Therefore, in the proposed algorithm (LGM+WP), we compare the Legendre moments of the original image histogram along with the (σ, γ) parameters of the wavelet subimages using the following equation.

$$d_{LGM+WP}(Q, C) = \sum_{k=0}^{N_M} (\lambda_{Q_k} - \lambda_{C_k})^2 + \sum_{k=1}^{N_{WB}} A_k (\sigma_{Q_k} - \sigma_{C_k})^2 + \sum_{k=1}^{N_{WB}} B_k (\gamma_{Q_k} - \gamma_{C_k})^2 \quad (5.11)$$

where, λ 's are the Legendre moments, σ_k 's and γ_k 's are the standard deviation and shape parameters of wavelet subimages, respectively. The parameters N_M and N_{WB} are the number of moments and wavelet bands used for comparison. The images that have minimum distance are retrieved from the database.

We now evaluate the performance of the joint moment and wavelet technique. We note that the selection of various weights of Eq. (5.11) is crucial to achieve good performance. Generally, the weights A_k and B_k are estimated by trial and error procedure, which also change with the number of feature elements employed. A typical set of values for $N_M=12$ and $N_{WB}=9$ is shown in Table 5.14. We note that the weights of various λ 's are unity, due to Eq. 5.8. Among the three features, σ_k 's generally have a high magnitude, and hence, A_k 's have been given the smallest weight. The magnitude of B_k is higher for lower k , giving more importance to the low-resolution bands. The relative distribution of energy automatically signifies the importance of a wavelet band. Hence, identical A_k 's have been used for all bands.

Table 5.14

Relative weights of various feature parameters employed for performance evaluation. The wavelet band indices correspond to Fig. 2.8(c).

Feature Vector Indices	Weights of λ_k	A_k , Weights of σ_k	B_k , Weights of γ_k
0	-	-	-
1	1	0.02	0.30
2	1	0.02	0.30
3	1	0.02	0.30
4	1	0.02	0.15
5	1	0.02	0.15
6	1	0.02	0.15
7	1	0.02	0.07
8	1	0.02	0.07
9	1	0.02	0.07

The performance of the LGM+WP technique employing images from IDB1 and IDB2 is shown in Fig. 5.25. It is observed that the LGM technique provides a retrieval efficiency of 88% for IDB1 database. The wavelet parameters improves the retrieval efficiency (see Fig. 5.25a) by atleast 4% over the LGM technique. For IDB2 technique, the WP technique provides a superior performance (with 83% retrieval efficiency) compared to LGM technique. However, the joint technique (LGM+WP) provides an improvement of 3-5% over WP technique (see Fig. 5.25b).

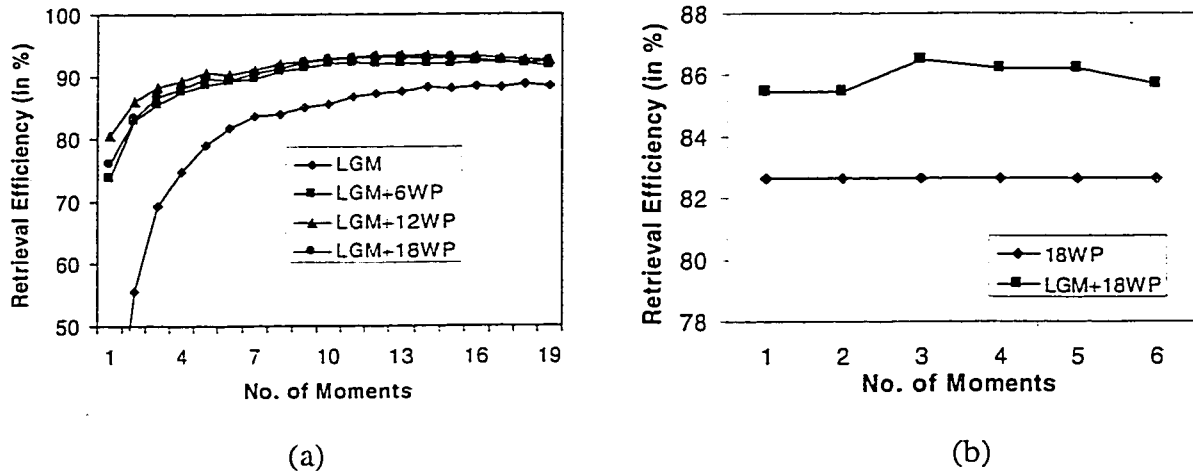


Figure 5.25. Indexing performance of Legendre moments and wavelet parameters. (a) image database IDB1, (b) image database IDB2. LGM: Legendre moments, 6WP: 6 wavelet parameters from 3 highpass bands of lowest scales, 12WP: 12 wavelet parameters from 6 highpass bands of two lowest scales, 18WP: wavelet parameters of all 9 highpass bands.

We now present a more detailed performance analysis for texture image retrieval. We recall that the DOIH technique failed to retrieve all desired images for the two visual examples shown in section 5.1.1 (see Fig. 5.10 and 5.11). We now evaluate the performance of LGM+WP technique using the same queries and image databases (with $N_M=4$, and $N_{WB}=9$). The results are shown in Fig. 5.26 and 5.27. Fig. 5.26b-j shows the nine images retrieved by LGM+WP technique in response to the query image shown in Fig. 5.26a (which

is same as Fig. 5.10a). It is observed that the proposed technique is able to retrieve all (totaling 7) similar images. In case of tile texture image, shown in Fig. 5.27a, the retrieved set contains all three similar images present in the database. Thus, the proposed technique provides a superior performance, exploiting the directional information present in the images.

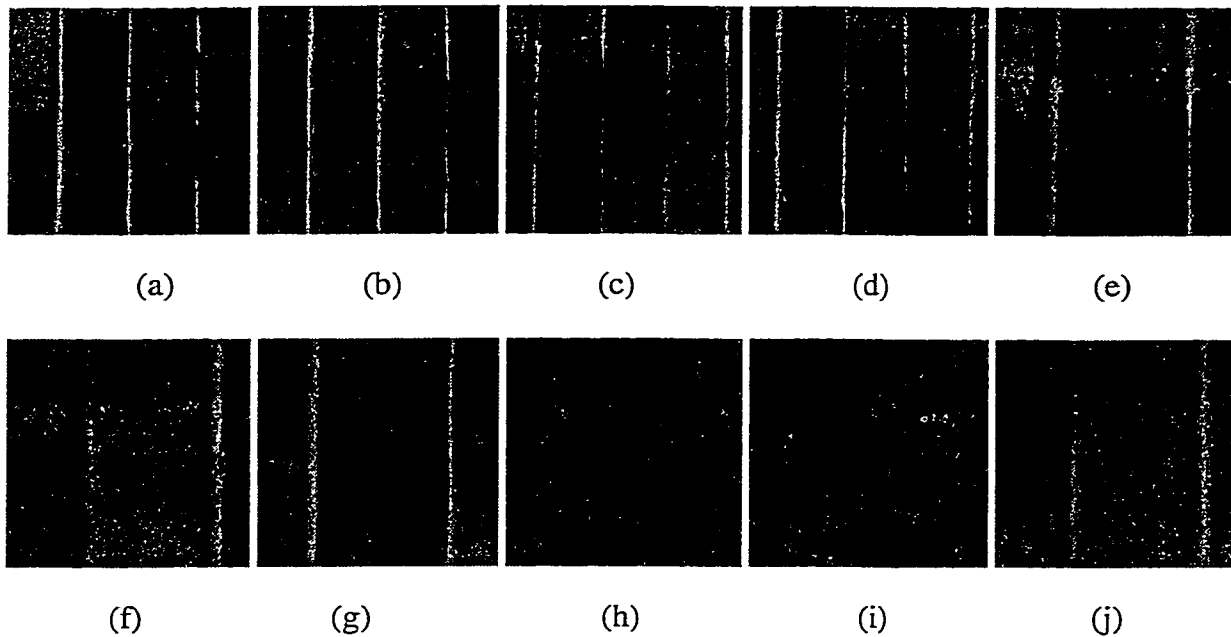


Figure 5.26. Images retrieved from IDB2 using LGM+WP technique - first query. a) query image, (b)-(j) retrieved image in monotonically increasing order of distance.

It should be noted that the proposed LGM+WP technique retrieves images having similar histograms, and directional properties. However, the visual content may not always be represented by histograms. In these cases, the LGM+WP technique may have limited success. To illustrate this further, consider a query image, of a water-rapid, shown in Fig. 5.28a. The proposed technique retrieves images shown in Fig. 5.28 (b)-(m). If our intention is to retrieve all images containing any water scenes, the retrieval technique should retrieve Fig. 5.28(b), (k)-(m). Although, the proposed technique could retrieve Fig. 5.28(b) successfully, it retrieves Fig. 5.28 (c)-(j) before retrieving the other desired images. A

careful visual inspection of the images shows that the texture of Fig. 5.28(a) is substantially different from that of Fig. 5.28(k)-(m), except the fact that they all contain water. This raises the question of defining "similarity". The proposed technique is not intended for applications where the objective is to retrieve all images containing a particular object (in the present case, it is water) regardless of the textural property or gray-level distribution.

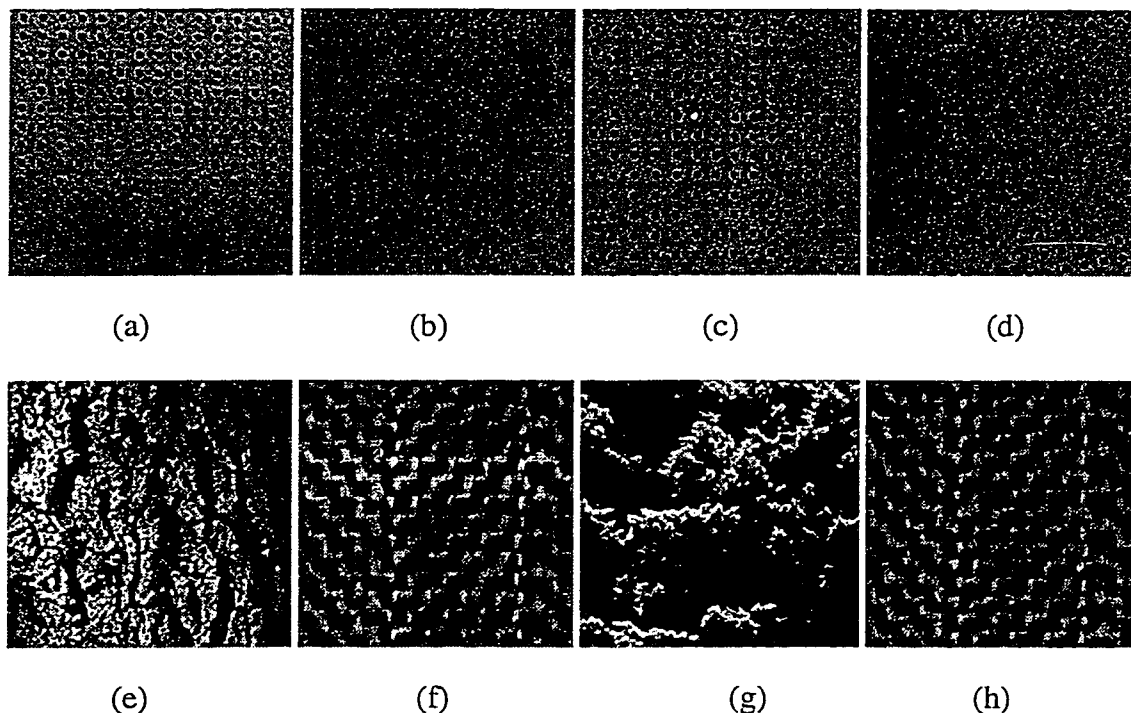


Figure 5.27. Images retrieved from IDB2 using LGM+WP technique - second query. a) query image, (b)-(h) retrieved image in monotonically increasing order of distance.

The level of confidence in a successful retrieval increases with the tolerance ρ (*i.e.*, the number of extra images retrieved). Fig. 5.29 shows the improvement in η_R with increasing ρ as a percentage of the total number of images stored in database. It is observed that in case of IDB1, a retrieval efficiency of 98% is obtained when ρ is 20%. This figure of merit,

however, depends on the characteristics of images in the database and is expected to change from one database to another.

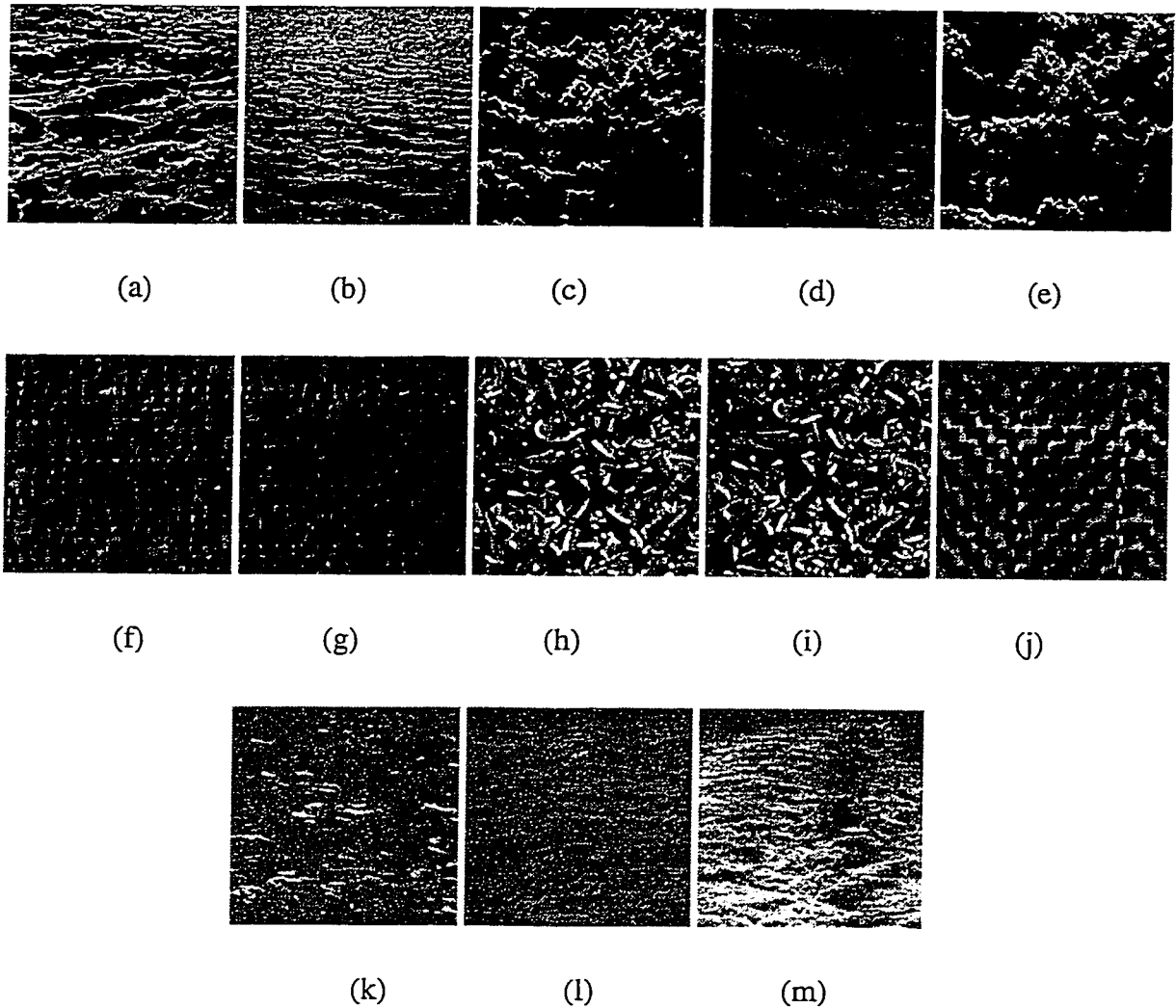


Figure 5.28. Example of failure of LGM+WP technique. a) query image, (b)-(j) retrieved image in monotonically increasing order of distance, (k)-(m) similar images, but not retrieved.

5.2.4 Section Summary

In this section, we have addressed the issue of generating feature vectors in the wavelet domain for efficient indexing. In section 5.2.1.1, we have proposed fast wavelet histogram techniques that reduce the computational complexity substantially, while retaining the

retrieval efficiency. An indexing technique based on the distribution of wavelet coefficients has been proposed in 5.2.2. In section 5.2.3, we have combined this technique with the Legendre-moment technique proposed in section 5.1.2. The joint technique, LGM+WP, provides a good indexing performance, at a low complexity. In addition, it is efficient for general as well as texture databases.

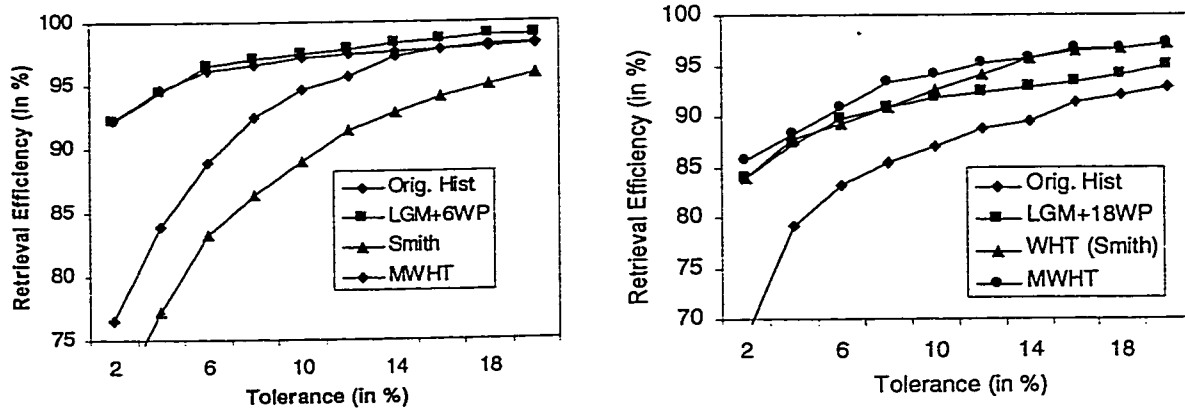


Figure 5.29. Tolerance (ρ) versus retrieval efficiency (η_R). (a) image database IDB1, (b) image database IDB2.

5.3 Summary

In this chapter, we have presented four novel techniques for image indexing. In the first technique (LGM), we proposed Legendre moments of image histogram as an index. It has been shown that Legendre moments, due to the orthogonality of Legendre polynomials, provide a superior retrieval efficiency compared to other (regular/central) moments. This technique provides a good performance for natural images. Secondly, we have proposed a fast wavelet-histogram technique (FWHT) which has a lower complexity. This technique is suitable for retrieving images with strong directional properties (e.g., texture images). Thirdly, we have proposed an indexing technique (WP) based on distribution of wavelet

coefficients. It has been shown that this technique provides good retrieval efficiency for images with strong directional properties. Fourthly, we combined LGM and WP technique. The LGM+WP technique provides good indexing performance for natural as well as texture images. In addition, the technique is suitable for real-time implementation, due to its low complexity.

Chapter 6

Illumination Invariant Image Indexing

In chapter 5, we have proposed several indexing techniques based on histogram, moments, and wavelets. These techniques provide a good retrieval performance when the query and the desired images have similar brightness level. However, the assumption of similar brightness is violated in practice due to the following reasons: i) variation in ambient illumination conditions, ii) nonlinear automatic gain control of camera, ii) variations in camera signal response, iv) nonlinearity in photo-developing, scanning, and digitization process. In this section, we propose two illumination invariant indexing techniques based on moments and wavelets. These techniques are basically modified versions of the indexing techniques proposed in chapter 5.

The organization of this chapter is as follows. The translation and scale invariant moment technique is detailed in section 6.1. The wavelet-based indexing technique is presented in section 6.2. The joint moment and wavelet technique is detailed in section 6.3. Section 6.4 presents the computational complexity of the proposed techniques. The performances of the proposed techniques are evaluated in section 6.5. We conclude the chapter with a summary.

6.1 Translation and Scale Invariant Moments

According to the coefficient image model described in section 2.1, the intensities of various color components change proportionately with the illumination. Hence, the histogram of an image, which is the *pdf* of the image intensity values, will be dilated proportionately with the illumination. Fig. 6.1 shows an image that was acquired using a CCD camera at two different lighting conditions. It is observed that the envelopes of two histograms are very similar. The histogram of Fig. 6.1(b) (*i.e.*, the brighter image) is shifted towards the right and is also stretched compared to the histogram of Fig. 6.1(a) (*i.e.*, the darker image). Generally, the histograms of an image with varying lighting conditions can be approximated as a translated and scaled version of each other. The amount of translation and dilation also depends on the photo developing and digitization process.

The L^2 distance between an original histogram and its translated and dilated version is generally large. Hence, image histograms cannot be employed directly for indexing applications when the brightness level is expected to change. Here, we propose to counter the effect of brightness change by employing translation and scale invariant (TSI) histograms. A TSI histogram is calculated such that its first moment is zero and its second moment is uniform. An elegant way of comparing TSI histograms is to compare its moments which can be easily calculated from the original histograms. We now describe the computation of TSI moments of a histogram.

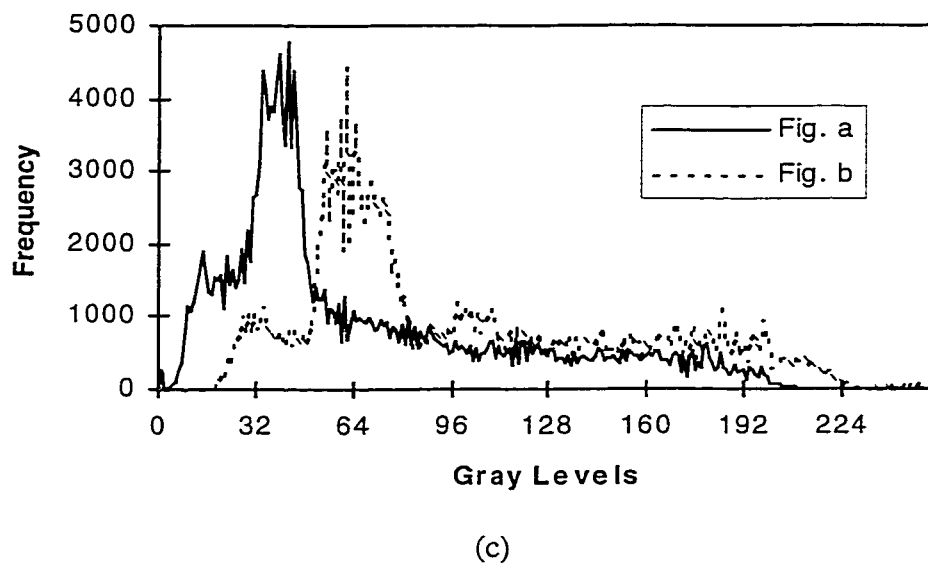
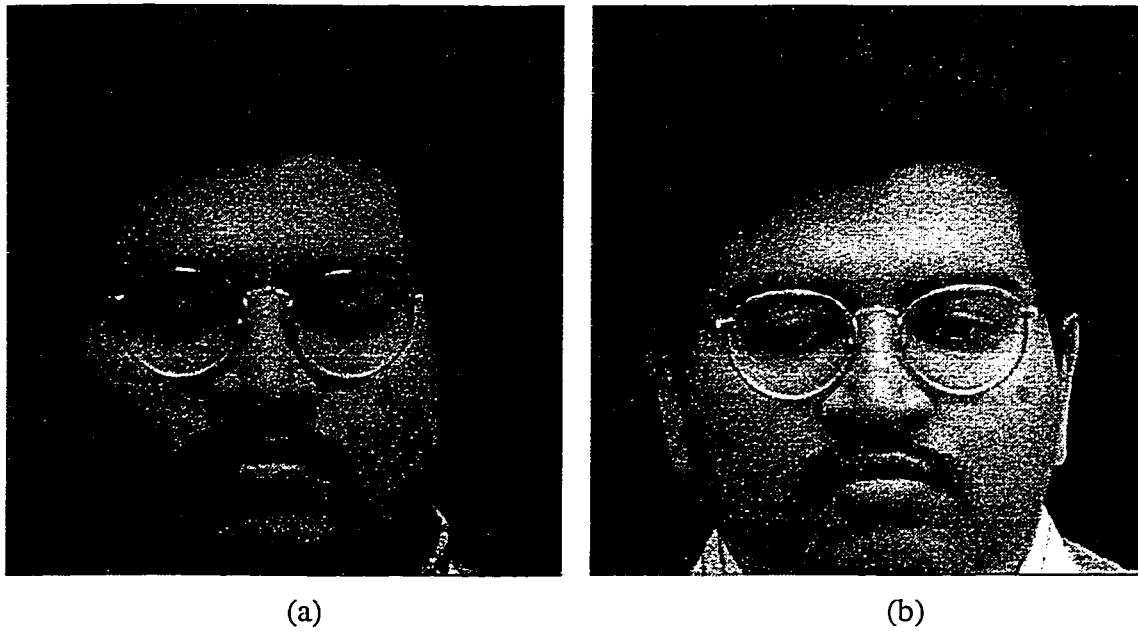


Figure 6.1. An image with different illumination levels and their histograms

Let's assume that the change in illumination has dilated the *pdf* $f(x)$ (which is a scaled version of the histogram) by a factor of α and the dilated *pdf* is represented by $f'(x) = f(x/\alpha)/\alpha$. We note that the amplitude of the new *pdf* has been scaled in order to keep the overall area unity under the *pdf*. The central moments of the new *pdf* can now be expressed as:

$$\begin{aligned}
\mu'_k &= \int (y - \bar{y})^k f'(y) dy = \int (y - \bar{y})^k \frac{f(y/\alpha)}{\alpha} dy \\
&= \int (\alpha x - \alpha \bar{x})^k \frac{f(x)}{\alpha} \alpha dx && \text{by changing variable } x = y/\alpha \\
&= \alpha^k \int (x - \bar{x})^k f(x) dx \\
&= \alpha^k \mu_k
\end{aligned} \tag{6.1}$$

The normalized central moments of the new *pdf* can then be expressed as

$$\beta'_k = \alpha \beta_k \tag{6.2}$$

It is observed that the normalized central moments of the two *pdfs* are related by a constant factor. A set of normalized moments that are invariant to scale α , can be defined as:

$$\eta_k = \frac{\beta_k}{\beta_2} \quad k > 2, k \in \mathbb{Z} \tag{6.3}$$

We note that η_k 's will be the same for both *pdf*'s irrespective of the values of α . In addition, β_1 cannot be used in Eq. 6.3, since it is zero. The translation and scale invariant (TSI) moments of Eq. 6.3 represent a *pdf* whose first moment is zero and second moment is unity. The *pdfs* of all the images are modified such that they all have unity second moment. Thus, the effect of uniform change in illumination is neutralized. It has been observed in section 5.1.2 that Legendre moments provide a superior performance compared to central moments. We now outline a procedure to convert the TSI moments to the corresponding Legendre moments.

We note that Legendre polynomials are orthogonal only in the interval $[-1,1]$. Hence, in order to have orthogonal moments, the histogram is mapped to the Legendre interval $[-1,1]$. If the images in the database have N_G gray levels (typically, $N_G = 256$), the interval is

divided into $N_G - 1$ bins. Let the center of mass of the image histogram be denoted by $\bar{m}$. The image histograms are now mapped to the Legendre interval, by aligning the $\bar{m}$ th bin of the image histograms with the origin of the Legendre axis. If $\bar{m} \geq N_G/2$, the negative bins are filled with values of *pdfs* in the range $[\bar{m} - N_G/2, \bar{m}]$ and positive bins are filled with the values of *pdfs* in the range $[\bar{m}, N_G - 1]$. We note that a few histogram bins ($\bar{m} - N_G/2$ bins to be precise) with lowest gray levels in the original histogram are ignored while mapping, and same number of bins in the mapped histogram with highest gray levels are initialized with zero. For $\bar{m} < N_G/2$, again bin- $\bar{m}$ is aligned with the origin in Legendre axis and in this case, $(N_G/2 - \bar{m})$ bins with highest gray levels will be ignored.

Fig. 6.2 shows the histogram of Fig. 6.1(b) (which has 256 gray levels) mapped to the Legendre interval. In this case, the mean ($\bar{m}$) is equal to 100 which is aligned with the origin of the Legendre axis and subsequently the whole histogram is shifted. In this process, the gray level ranging [228, 255] in the original image is lost. This loss of histogram bins (due to centralization) could have been avoided by a fine sampling of the Legendre axis. However, this will make a major portion of the *pdf* equal to zero and the corresponding moments will be less effective, resulting in a degradation in performance. However, if $\bar{m} \gg 128$ or $\bar{m} \ll 128$, the performance can be improved by a finely sampled Legendre axis.

The implementation of indexing technique using TSI Legendre moments (henceforth denoted by TSI-LGM technique) is executed in two steps. In the first step, a feature vector for each image is precomputed and stored in the database. In the second step, the feature

vector is retrieved from the database and processed for indexing. The complete procedure is as follows:

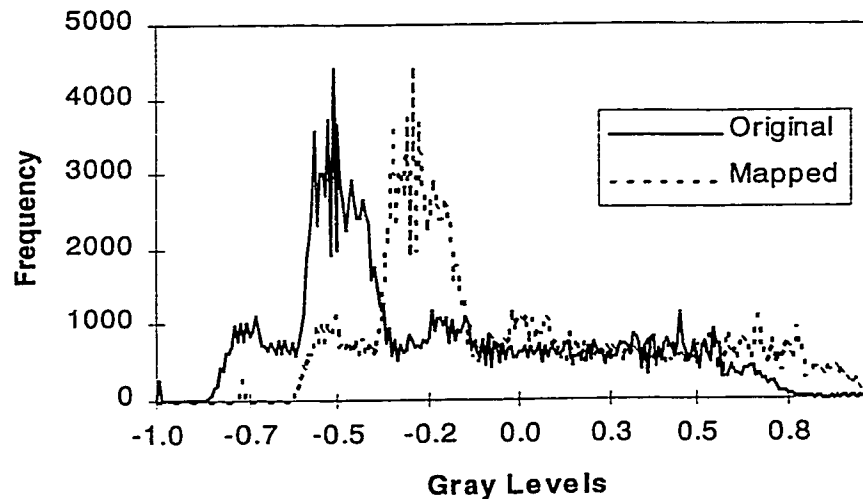


Figure 6.2. Mapping of a gray level histogram $[0,255]$ to Legendre interval $[-1,1]$ with the mean of the image ($\bar{m} = 100$) aligned to origin of the Legendre axis.

Feature Vector Generation

- i) Map the image histograms to Legendre axis by the procedure outlined above. The histograms are now translation invariant.
- ii) Compute the central moments of the above histograms. We note that the central moments and the regular moments are the same, since the histogram has a zero center of mass.
- iii) Store these moments along with the images in the database.

Indexing

- i) Compute the central (μ_{Q_i}) and Legendre (λ_{Q_i}) moments of the query image (Q) using the procedure outlined above.

- ii) Estimate α by comparing the second central moment (μ_{Q_k}) of the query image with that of a candidate image (μ_{C_k}) from the database.
- iii) Modify the central moments of the target image using Eq. 6.1. Let the moments be denoted by μ'_{C_k} .
- iv) Compute the Legendre moments (λ'_{C_k}) corresponding to the μ'_{C_k} using the following equation (a_{kj} are constant values, see section 5.1.2 for more details).

$$\lambda'_{C_k} = \sqrt{(2k+1)/2} \sum_{j=0}^k a_{kj} \mu'_{C_k}$$

- v) Calculate the distance between the Legendre moments of the query image and the Legendre moments of the target image computed in step (iv) using the following equation.

$$d_{TSI-LGM}(Q, C) = \sum_{k=1}^{N_M} (\lambda_{Q_k} - \lambda'_{C_k})^2 \quad (6.4)$$

where N_M is the number of moments employed for indexing.

- vi) Repeat (ii)-(v) for all images in the database.
- vii) The images that provide the least distance are retrieved.

6.2 Wavelet-based indexing

It was observed in section 5.2.2 and 5.2.3 that the indexing performance of histogram techniques can be improved using wavelet-band histograms (see Eq. 5.11). In this section, we present a modified indexing technique that is robust to changes in illumination. First, we show the change in the standard deviation of wavelet bands when the gray levels of an image

are translated and scaled. We can illustrate this change with a 1-D signal, which can be easily extended to 2-D image. Suppose, a signal $f(x)$ is translated by ε , and scaled by α , *i.e.*, the changed signal can be expressed as $f'(x) = \alpha f(x) + \varepsilon$. The wavelet transform of $f'(x)$ can be expressed as (see Eq. 2.10):

$$\begin{aligned}
 F'(a, b) &= \int_{-\infty}^{\infty} [\alpha f(x) + \varepsilon] \Psi_{a,b}^*(x) dx \\
 &= \alpha \int_{-\infty}^{\infty} f(x) \Psi_{a,b}^*(x) dx + \varepsilon \int_{-\infty}^{\infty} \Psi_{a,b}^*(x) dx \\
 &= \alpha F(a, b), \quad \text{since } \int_{-\infty}^{\infty} \Psi(x) dx = 0
 \end{aligned} \tag{6.5}$$

where $F(a, b)$ is the wavelet transform of $f(x)$.

We note from Eq. 6.5 that when the gray levels of an image are translated only by a fixed amount (*i.e.*, $\alpha=1$, ε fixed), the highpass DWT coefficients remain unchanged, since the integral of a wavelet function is equal to zero. In other words, the statistics of the wavelet bands are invariant to the translation of gray levels in an image.

We now investigate the effect of scaling on the distribution parameters of the wavelet bands. Let us assume that the mean and variance of k th wavelet band $F(b)$ are ξ_{F_k} and $\sigma_{F_k}^2$, respectively. The mean of the corresponding band in $F'(b)$ will be $\alpha \xi_{F_k}$, and the variance can be expressed as:

$$\begin{aligned}
 (\sigma'_{F_k})^2 &= \int [F'(b) - \alpha \xi_{F_k}]^2 db \\
 &= \int (\alpha F(b) - \alpha \xi_{F_k})^2 db \\
 &= \alpha^2 \int F^2(b) db - 2\alpha^2 \xi_{F_k} \int F(b) db + \alpha^2 \xi_{F_k}^2 \int db \\
 &= \alpha^2 \sigma_{F_k}^2 - 2\alpha^2 \xi_{F_k} \int F(b) db + \alpha^2 \xi_{F_k}^2 \int db
 \end{aligned} \tag{6.6}$$

$$\approx \alpha^2 \sigma_{F_i}^2 \quad (6.7)$$

We note that the mean ξ of coefficients of a wavelet band is generally close to zero and hence the second and third terms in Eq. 6.6 have been ignored. We can see that the standard deviation σ of a highpass subimage is directly proportional to the scale factor α , *i.e.*,

$$\sigma'_{F_i} = \alpha \sigma_{F_i} \quad (6.8)$$

Fig. 6.3 shows the standard deviation of wavelet coefficients of different bands of an image along with the corresponding standard deviations of lower illumination (LI) and higher illumination (HI) versions of the same image. The scale factors of the LI and HI images are 0.67 and 1.5, respectively. Different illumination conditions were simulated using histogram mapping which is linear in the middle levels and marginally nonlinear at the edges. It is observed that the standard deviation closely follows Eq. 6.8.

A similar experiment was performed to find the relationship between the shape parameter γ and the scale factor α . Fig. 6.4 shows the γ of wavelet coefficients of different bands of an image along the corresponding γ of LI ($\alpha = 0.67$) and HI ($\alpha = 1.5$) version of the same image. It is observed that i) γ generally decreases with increase in α and vice versa, ii) the rate of change is much smaller compared to the rate corresponding to σ , and iii) the rate of change varies across bands. We have assumed for simplicity that the change in *shape* parameter is proportional to the change in scale factor (from the ideal value of 1), *i.e.*,

$$\gamma' = \gamma + c(1 - \alpha) \quad (6.9)$$

where, γ' is the new shape parameter.

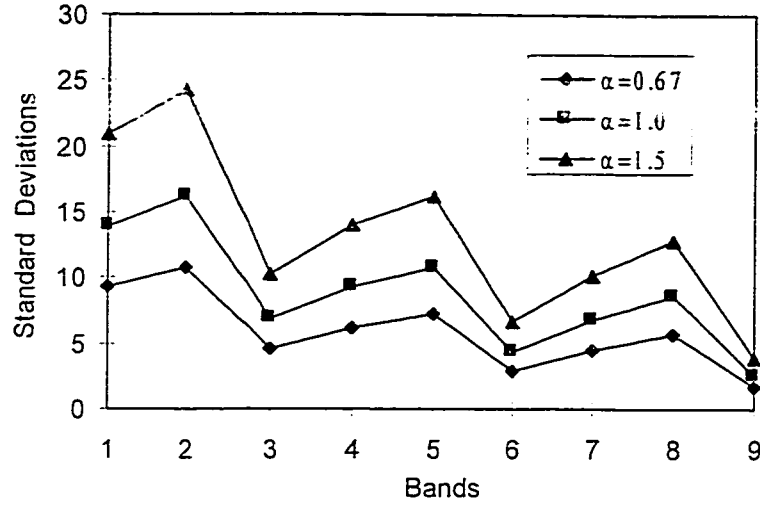


Figure 6.3. Change of standard deviation parameter (σ) with scale factor

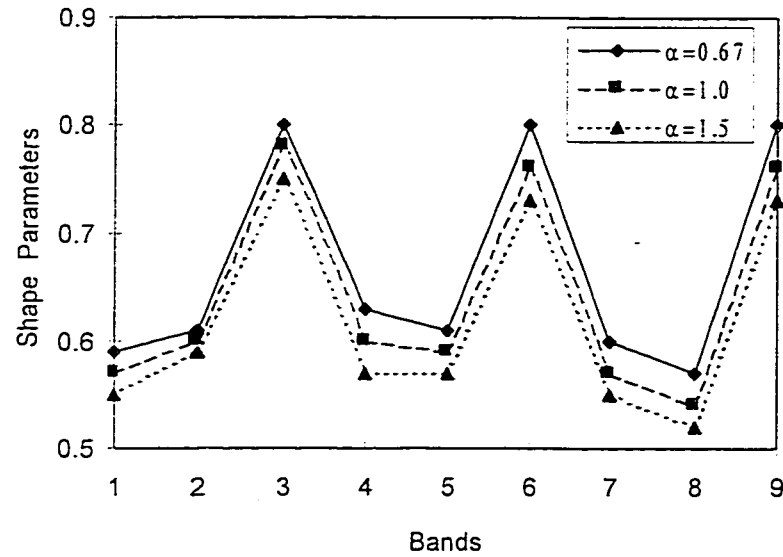


Figure 6.4. Change of shape parameter (γ) with scale factor

The proposed wavelet parameter based indexing technique (henceforth denoted by WP technique) is implemented as follows. Given a query image, the scale factor α for each reference image in the database, is estimated by comparing the β_2 (second order normalized central moment) of the query image histogram with that of the reference image. Illumination compensated standard deviation and shape parameters of the highpass wavelet subimages are

then calculated by Eqs. 6.8 and 6.9, respectively. The similarity of the two images is estimated using the weighted Euclidean distance of these parameters.

$$d_{wp}(Q, C) = \sum_{k=1}^{N_B} A_k (\sigma'_{Q_i} - \sigma'_{C_i})^2 + \sum_{k=1}^{N_B} B_k (\gamma'_{Q_i} - \gamma'_{C_i})^2 \quad (6.10)$$

where, σ' 's and γ' 's are, respectively, the illumination compensated *standard deviation* and *shape* parameters of wavelet highpass subimages. The parameter N_B refers to the number of wavelet bands used in the indexing process. A_k and B_k are the weights of the parameters which are estimated by trial and error to achieve the best performance. The images which have minimum distance $d(.)$ with respect to the query image, are retrieved from the database.

6.3 Joint Moment and Wavelet Based Indexing

The moment (TSI-LGM) and the wavelet techniques (WP) can be combined (henceforth referred as TSI-LGM+WP technique) to obtain a superior coding performance. The distance between the query image Q and the candidate image C corresponding to the TSI-LGM+WP technique is expressed as:

$$d_{TSI-LGM+WP}(Q, C) = \sum_{k=1}^{N_M} (\lambda'_{Q_i} - \lambda'_{C_i})^2 + \sum_{k=1}^{N_B} A_k (\sigma'_{Q_i} - \sigma'_{C_i})^2 + \sum_{k=1}^{N_B} B_k (\gamma'_{Q_i} - \gamma'_{C_i})^2 \quad (6.11)$$

where, λ'_{C_i} 's are the TSI Legendre moments of the target image, σ'_{C_i} 's and γ'_{C_i} 's are the illumination compensated *standard deviation* and *shape* parameters of wavelet bands of the query image, respectively. The parameters N_M and N_B refer to the number of moments and wavelet bands used in the indexing process. The images that have minimum distance $d(.)$ with respect to the query image are retrieved from the database.

We note that the weights A_k and B_k are again estimated by trial and error procedure. A typical set of values for $N_M=4$ and $N_B=9$ is shown in Table 6.1. The weights tabulated in Table 6.1 are different than those employed in Table 5.14. It is observed that greater weights have been assigned to σ_k 's and γ_k 's in Table 6.1. This has been done to give more importance to the wavelet parameters, as the moments in Eq. 6.11 are not as robust as in 5.11.

6.4 Computational Complexity

In this section, we provide an estimate of the computational complexity of the proposed techniques. There are two types of complexities involved – i) complexity (ξ_G) for feature vector generation, and ii) complexity (ξ_C) for feature vector comparison. We note that the feature vector may be generated only once, and stored along with the image. On the other hand, the complexity for feature comparison is involved in each case of retrieval. While computing the complexity, only the arithmetic operations (addition, subtraction, multiplication, and division) have been considered. We assume the following - N_p : Number of pixels in an image, N_G : No. of histogram bins, N_{GR} : No. of histogram bins for 3-D ratio histogram, N_M : No. of moments.

The computational complexity for feature generation is tabulated in Table 6.2. Entries in row 2 provide the joint complexity ($=N_p + 2N_M N_G$) of histogram and moment calculation. Entries in row 3 provide the complexity ($0.70N_p$) of wavelet parameter generation employing Sharifi *et al.*'s algorithm [100]. The joint complexity of moment and wavelet parameter estimation is provided in row 4. Table 6.2 also shows the complexity of the ratio

histogram (RHT) technique discussed in section 3.3. We note that RHT technique requires logarithm, and convolution operations in addition to histogram computation. The complexity of these operations depends on the implementation hardware and the size of the convolution filters. We have employed K_1 and K_2 which provide the constant multiplicative factor for logarithm and convolution operations, respectively. It is observed that the overall complexity of generating features for RHT technique is substantially greater than the proposed technique.

Table 6.1

Typical relative weights of various feature parameters.

Feature Vector Indices	Weights of λ_k	A_k , Weights of σ_k	B_k , Weights of γ_k
0	-	-	-
1	1	0.013	2.2
2	1	0.013	2.2
3	1	0.013	2.2
4	1	0.013	1.1
5	-	0.013	1.1
6	-	0.013	1.1
7	-	0.013	0.5
8	-	0.013	0.5
9	-	0.013	0.5

The complexity of feature comparison is tabulated in Table 6.3. Entries in row 6 provide an estimate of the complexity/image/channel of TSI-LGM+WP technique. The complexity for three-color channels is approximately three times higher. The entries in the last row show the complexity for comparing 3-D ratio histogram. The factor $N_{GR} = 343$ is employed for a 3-D histogram where each dimension has been quantized to 7 bins. It is observed that the

overall complexity of comparing features for RHT technique is several times greater compared to the proposed technique.

Table 6.2

Computational Complexity of feature generation for TSI indexing techniques. The values are given in complexity/image/channel.

Operations	Complexity	Typical Value ⁻
Centralized Histograms	$\approx N_p$	6.5e+4
Regular/Central Moments	$N_p + 2N_M N_G$	6.7e+4
Wavelet Parameters ($N_B=6$)	$0.70N_p$	4.5e+4
TSI-LGM+WP ($N_B=6$)	$N_p + 2N_M N_G + 0.70N_p$	1.12e+5
Logarithm	$K_1 N_p$	3.9e+5
2-D Convolution	$K_2 N_p$	3.9e+5
Histogram	N_p	6.5e+4
RHT	$(K_1 + K_2 + 1) * N_p$	8.5e+5

⁻for $N_p = 65536$ $N_M = 3$ $N_G = 256$, $K_1 = 6$, $K_2 = 6$

6.5 Performance of Illumination Invariant Indexing Techniques

In this section, we evaluate the performance of the proposed illumination invariant indexing technique. Simulations have been carried out using two databases of images IDB3 and IDB4 (see appendix B). All the images are 24-bit color images $\{Y, Cb, Cr\}$ of size 248×256 . In the first database (IDB3), we have chosen forty four images, where each image has three derivatives corresponding to i) standard, ii) low, and iii) high illumination. Thus, we have a total of 132 images with varying illumination conditions. The different illumination conditions were simulated using histogram modification that is linear in the

central region and marginally nonlinear at the edges. The LI and HI conditions were simulated using $\alpha = 1.25$ and $\alpha = 0.8$, respectively.

Table 6.3

Computational complexity of feature comparison for TSI indexing techniques. The values are given in complexity/image/channel.

Operations	Complexity	Typical Value [*]
Scaling of central moments	$2N_M$	6
Central-> Legendre moments	$((N_M + 3)^2 - 9)/2$	14
Wavelet parameters (σ')	1	1
Wavelet parameters (γ')	3	3
Feature comparison	$2N_M + 4N_B$	30
TSI-LGM+WP (gray)	$\approx 4N_M + (N_M + 3)^2/2 + 4N_B$	44
TSI-LGM+WP (color)	$\approx 3(4N_M + (N_M + 3)^2/2 + 4N_B)$	132
RHT (color)	$2N_{GR}$	686

^{*}for $N_M = 3$, $N_B = 6$, and $N_{GR} = 343$

In addition to changes in illumination conditions, the users may be interested in retrieving images which are translated and rotated versions of the query image. Therefore, in the second image database (IDB4), we have chosen forty four images, where each image has three derivatives corresponding to i) standard position and illumination, ii) clockwise rotated along with low illumination, and iii) anti-clockwise rotated along with high illumination. Thus, we have a total of 132 images with varying illumination levels as well as spatial changes. Given a randomly selected query image, the retrieval is considered to be successful if an indexing technique retrieves the other similar images. We note that the square error (L^2) metric has been used to evaluate the performance of all moment and wavelet techniques.

In order to estimate γ 's in Eq. 6.11, an optimal constant c has to be determined to calculate the shape parameters using Eq. 6.9. Here, we employ a statistical method to obtain a value of c for each wavelet band. Wavelet transforms are calculated for the entire database of images with Daubechies 8 tap (minimum-phase) wavelets. Ten wavelet bands are obtained with a three stage dyadic decomposition for each image and its LI and HI versions. The value of γ is calculated for each wavelet band from these histograms. Employing these γ 's, c_k is estimated for each band of an image combination- k (each combination consists of original, LI and HI images). The final value of c is estimated by averaging c_k 's over all combination- k . The nine c 's corresponding to the nine highpass bands range between 0.05 and 0.09 for the current image database. This verifies that shape parameters are marginally affected by changes in scale.

To evaluate the performance of Funt's [25] ratio histogram technique (RHT), the images in IDB3 and IDB4 are first converted from the $\{Y, Cb, Cr\}$ to the $\{R, G, B\}$ color space. The logarithms are calculated for each color channel and passed through a Laplacian operator. A 3-D ratio histogram is then computed with 7 bins in each color channel. We note that the histogram is nonuniformly quantized to achieve a superior performance. In addition, three 1-D ratio histograms are also calculated with 256 bins in each channel. The retrieval performance of RHT is shown in Table 6.4. It is observed that combined performance of three 1-D ratio histograms is 57.57% and 50.75%, respectively, for IDB3 and IDB4. The 3-D ratio histogram provides a superior retrieval performance, namely 77.65% and 64.39% for IDB1 and IDB2, respectively.

Table 6.4

Performance of ratio histogram [25] on IDB1 and IDB2 database of images (in {R,G,B} color space).

Image Database	3, 1-D Ratio Histogram	3-D Ratio Histogram
IDB3	57.57	77.65
IDB4	50.75	64.39

The performance of various indexing techniques on the $\{Y\}$ images of IDB3 is shown in Fig. 6.5. The retrieval efficiencies were calculated with $T = 6$ and $B = 3$. It is observed that the general moments (LGM and CNM) provide a retrieval efficiency of only 15-30%. We note that the retrieval efficiency depends on the degree of illumination change and generally degrades with higher illumination change. The TSI moments (TSI-M) and TSI Legendre moments (TSI-LGM) provide a retrieval efficiency of 65% and 80%, respectively. The retrieval efficiency of the wavelet-parameter (WP) technique (shown in Table 6.5) is approximately 97%. The joint Legendre moment and wavelet (TSI-LGM+WP) technique provides a retrieval efficiency of 99%.

It was observed in section 5.1.2 that the indexing performance of Legendre moments saturates beyond 15-16 moments for general databases (without illumination change) of images. However, in the present case (see Fig. 6.5) the optimal number of moments (excluding the zeroth, first and second order moments) ranges between two and four. This could possibly be a result of the fact that the dynamic range of image intensity levels is limited to 256 levels. Therefore, the effect of illumination is not truly linear for digital images as has been assumed in the model. As a result, the central moments are not strictly

related by an exponential factor as shown in Eq. 6.1. This is especially true if the illumination condition is changed significantly.

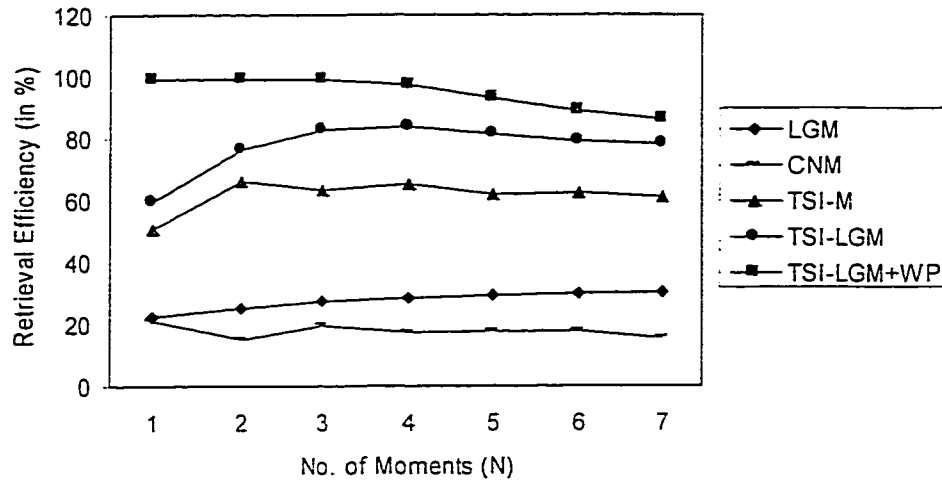


Figure 6.5. Retrieval efficiency (η_R) of various indexing techniques on $\{Y\}$ images of IDB3 (LGM: Legendre moments, CNM: Central Normalized Central moments, TSI-M: Translation and scale invariant moments, TSI-LGM: Legendre moments corresponding to TSI-M, TSI-LGM+WP: Combined TSI-LGM and Wavelet parameters). N: Number of moments excluding the zeroth, first and second order.

Table 6.5

Performance of Wavelet Parameter (WP) Technique.

Image Database	Performance
IDB3	97%
IDB4	55%

The performance of various techniques on all the three color channels $\{Y, Cb, Cr\}$ of IDB3 is shown in Fig. 6.6. While comparing the feature vectors, the weights of the two chrominance components $\{Cb, Cr\}$ were set to 30% of the $\{Y\}$ component, since these components generally provide less information. We observe in Fig. 6.6 that the performance

of CNM and TSI-M has significantly improved compared to Fig. 6.5. However, the improvement is not as significant for TSI-LGM+WP technique since the retrieval performance is already very high for $\{Y\}$.

The performance of various indexing techniques on $\{Y\}$ and $\{Y, Cb, Cr\}$ images of IDB4 is shown in Fig. 6.7 and Fig. 6.8, respectively. It is observed that the overall retrieval performance is worse compared to that of IDB3. This is expected since the order of similarity among different images obtained from the same original images in IDB4 is much lower compared to that of the IDB3. The WP technique has been found to provide a retrieval efficiency of approximately 55% (shown in Table 6.5). The TSI-LGM+WP technique provides a 10% improvement in performance over TSI-LGM. Comparison of Fig. 6.7 and Fig. 6.8 reveals that the color components provide a further 3-4% improvement over $\{Y\}$ components.

A visual comparison of TSI-LGM+WP technique with 3-D RHT technique is presented in Fig. 6.10 and Fig 6.11. The image shown in Fig. 6.9 is used as a query. We note that the database contains two other images similar to the query image. The first six images retrieved employing 3-D RHT technique are shown in Fig. 6.10. It is observed that only one similar image (Fig. 6.10b) has been retrieved. On the other hand, Fig. 6.11 shows the first six retrieved images employing TSI-LGM+WP technique. Here we note that the proposed technique could retrieve both similar images (Fig. 6.11a-b) corresponding to lower and higher illumination.

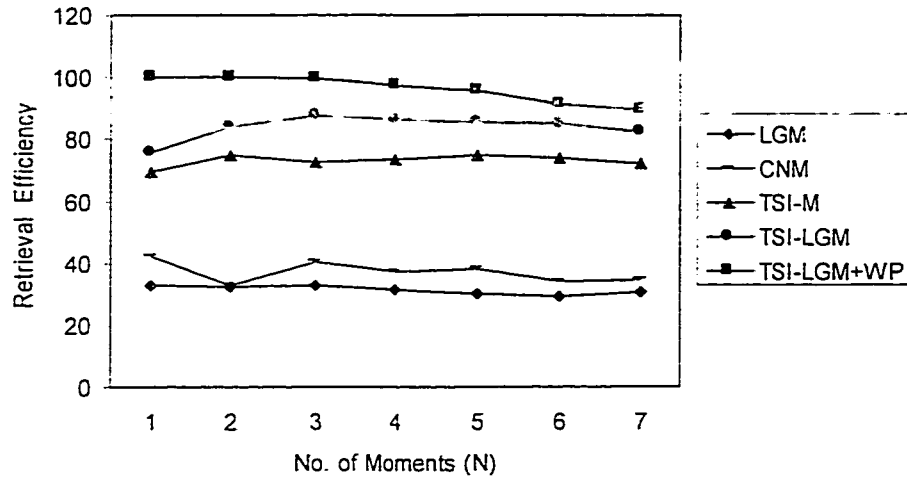


Figure 6.6. Retrieval efficiency (η_R) of various indexing techniques on color {Y,Cb,Cr} images of IDB3. N: Number of moments excluding the zeroth, first and second order

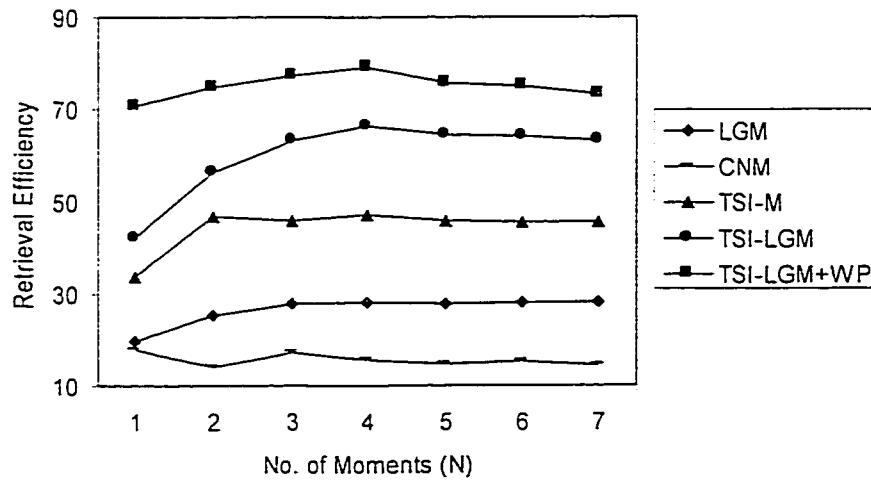


Figure 6.7. Retrieval efficiency (η_R) of various indexing techniques on gray level {Y} images of IDB4.

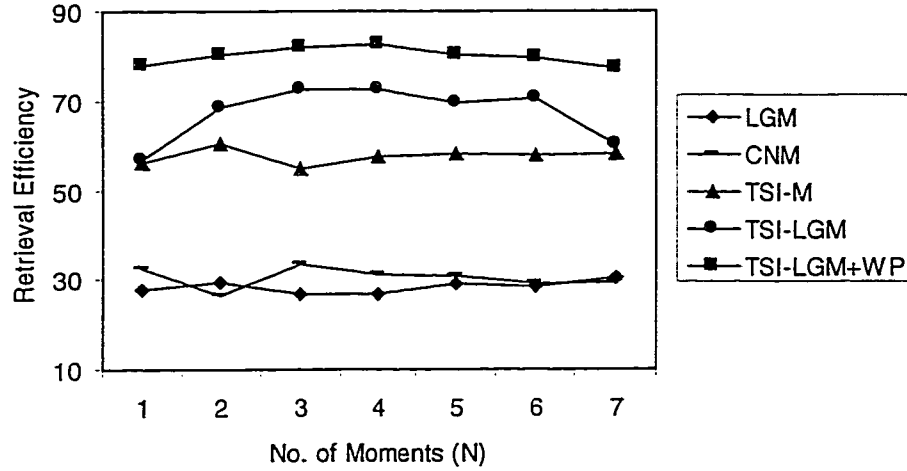


Figure 6.8. Retrieval efficiency (η_R) of various indexing techniques on color {Y,Cb,Cr } images of IDB4.

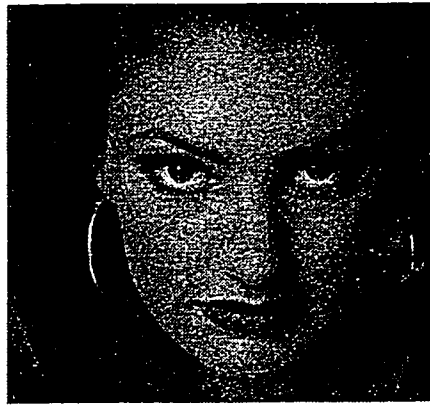


Figure 6.9. A typical query image.

It can be seen that the TSI-LGM+WP technique provides a superior indexing performance compared to Funt's algorithm (RHT). A possible explanation is as follows. The logarithmic operation on each color component substantially reduces the dynamic range of the pixel intensities. When the logarithmic image is convolved with a Laplacian operator, most of the data samples will be zero and the low frequency information is removed. This technique is similar to WP technique proposed here. However, while RHT compensates for changes in illumination through the logarithm operation, WP first estimates the changes in illumination

and then compensates for it explicitly through the use of standard deviation and gamma parameters. We note that the TSI-LGM component in the combined TSI-LGM+WP technique helps to provide an overall superior performance.

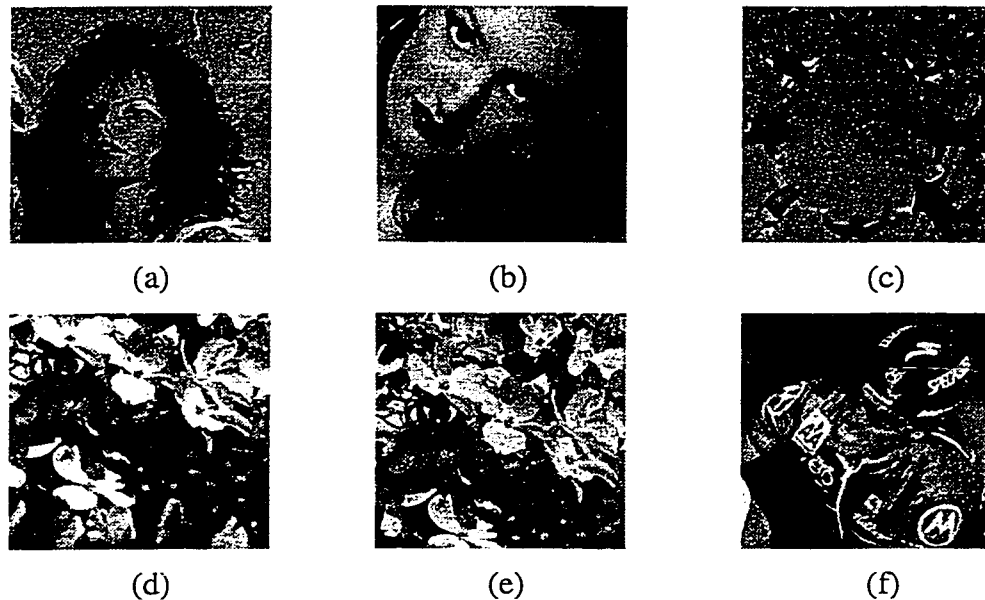


Figure 6.10. First six retrieved images from IDB2 employing 3-D RHT technique for query image shown in Fig. 6.9.

To achieve a high efficiency of retrieval, a large tolerance (T) should be used. The level of confidence in a successful retrieval increases with T . Fig. 6.12 shows how η_R improves with increasing T as a percentage of the total number of images stored in database. The performance was evaluated on images from IDB4 with the TSI-LGM+WP technique. It is observed that when T is about 20%, a retrieval efficiency of 92% may be obtained. We also note that the chrominance components, $\{Cb, Cr\}$, offer a marginal improvement in performance (2-4%). However, these figures of merit depend on the characteristics of the image database. Although, a higher T improves the retrieval efficiency, it also increases the number of retrieved images. If the number of retrieved images increases beyond a limit,

visual browsing might be difficult. In that case, a more sophisticated technique may be required to do a second retrieval pass.

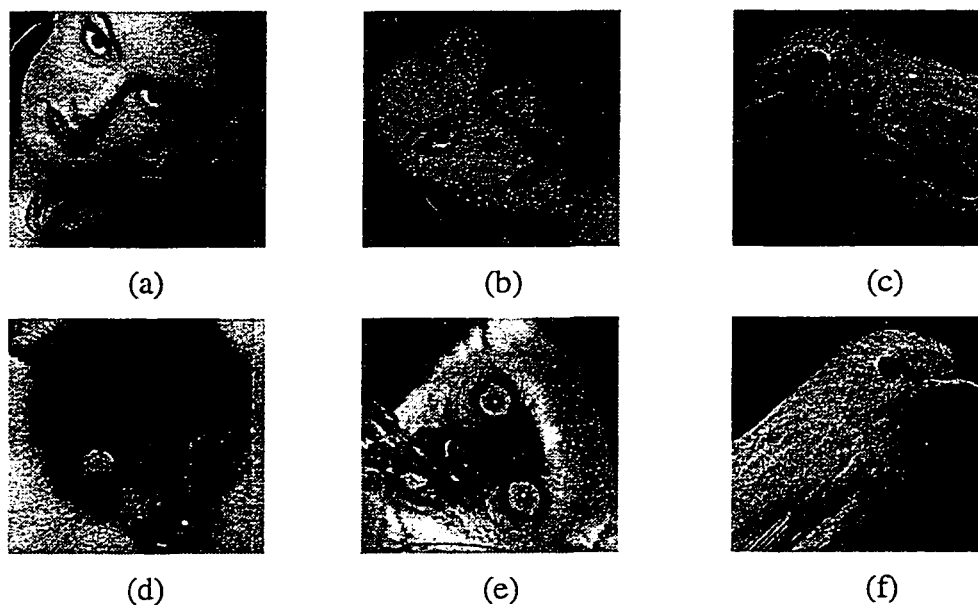


Figure 6.11. First six retrieved images from IDB2 employing TSI-LGM+WP technique for query image shown in Fig. 6.9.

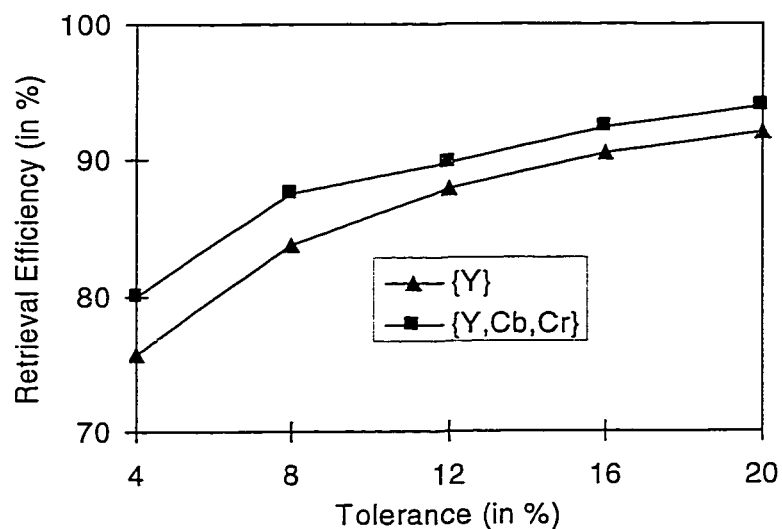


Figure 6.12. Tolerance (T) versus retrieval efficiency (η_R) of TSI-LGM+WP technique.

It is known that the coding performance of wavelet-based scheme depends considerably on the choice of wavelets. Wavelets with higher vanishing moments generally provide a better compression efficiency. Thus, we have carried out experiments to test the sensitivity of the indexing performance of the proposed technique with the choice of wavelets. Minimum phase Daubechies wavelets of 2-16 taps are chosen for this purpose. The performance has been seen to be marginally sensitive (up to 1-2%) to the number of wavelet filter taps. No correlation between the performance and the vanishing moments of the wavelets has been observed. This is expected since energy compaction is not an important issue in indexing. As long as the feature vectors (of the query and target images) are obtained with the same wavelet, the proposed techniques are expected to perform well.

6.6 Summary

In this chapter, we have presented two techniques for illumination invariant image indexing. First, we have presented a moment-based technique where translation and scale invariant Legendre moments are used as an index. Secondly, we have presented a wavelet-based indexing technique where the indexes are statistical parameters of different directional wavelet bands. These parameters are appropriately modified so that the performance is robust to changes in illumination. This technique will be attractive when a wavelet-based coding scheme is used for storing the images efficiently in the database. It has been shown that the two techniques can be combined to provide a superior indexing performance that is robust to illumination changes. The run-time complexity of this combined technique is small compared to the technique existing in the literature.

Chapter 7

Joint Coding and Indexing of Video in Wavelet Domain

We recall from chapter 3 that joint coding and indexing is crucial for designing an efficient video storage and archival system. We note that several wavelet-based video compression techniques have been proposed in chapter 4 while image indexing techniques have been proposed in chapter 5. In this chapter, we present a generic video storage and archival system in the wavelet domain by synthesizing the techniques presented in the previous chapters. The organization of the chapter is as follows: a brief introduction to the requirements of video indexing is given in section 7.1. Section 7.2 details pixel and compressed domain features that can be employed in detecting shots. Performance evaluation of these features is also included in section 7.2. A fast video segmentation technique in the wavelet domain is proposed in section 7.3. Section 7.4 presents the selection of representative frames and matching of query images. We conclude the chapter with a summary.

7.1 Video Indexing Requirements

The requirements of a typical image archival and retrieval system (see Figs. 3.1 and 3.7) was discussed in chapter 3. It was noted that a multidimensional feature vector is generally computed for each image. Indexing is then performed based on the similarities of the feature vectors. Typical image features employed for indexing (see Figs. 3.2 and 3.8) include histogram, color, texture, transform coefficients, *etc.*

Since video is a sequence of image frames ordered in time, the image indexing techniques discussed in chapter 3 and 5 can be applied to each frame individually. Since adjacent frames in a video sequence belonging to a shot are similar, the sequence is first segmented into shots. Image indexing techniques are then applied on a representative frame from each shot for retrieval, resulting in a reduced complexity. A typical video storage and archival system is shown in Fig. 7.1 where shots are detected either in the pixel domain (Fig. 7.1a), or in the compressed domain (Fig. 7.1b). We note that when an input video sequence is stored in the compressed format, it is more efficient to detect shots in the compressed domain. This eliminates the need to decompress the video and hence results in significant savings in execution time.

In this chapter, we develop techniques for detecting shots with abrupt transition (AT). We note that detection of AT's is very important in video indexing applications. For example, Fig. B.6 (see Appendix B) shows first 480 frames of the Beverly Hills sequence. Visual inspection shows that there are 24 abrupt transitions in the Beverly Hills sequence. The frame numbers at which the transitions occur are shown in Table 7.1. The objective of the indexing techniques is to automatically detect these transitions.

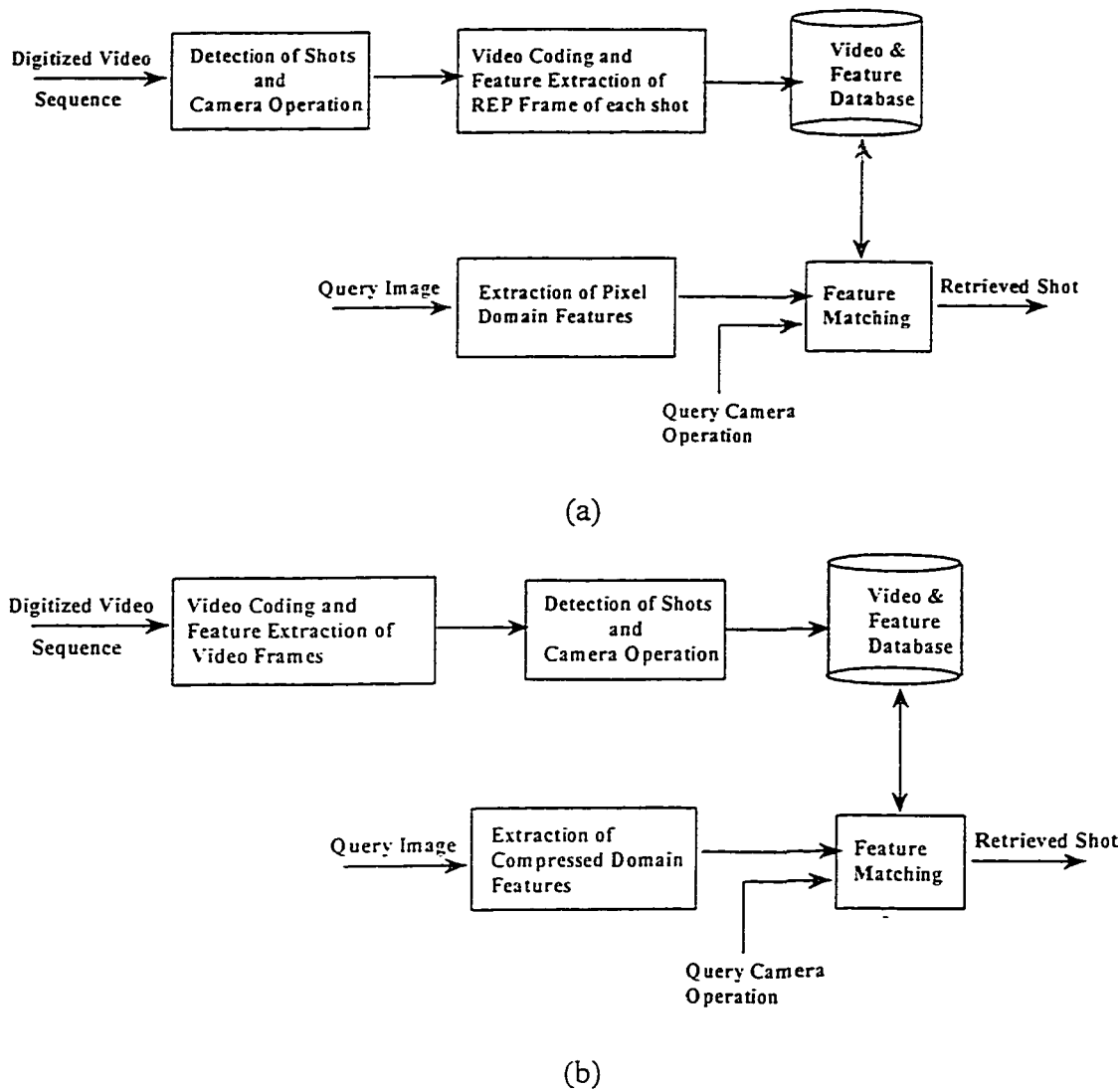


Figure 7.1. A schematic of image archival and retrieval system. Shots and camera operations are detected in (a) pixel domain, (b) compressed domain.

7.2 Potential Features for Abrupt Transition Detection

In this section, we explore various features that can be used to detect abrupt transition in a wavelet-based framework. The pixel domain features include image histogram or its moments whereas the compressed domain features include the DWT coefficients and the band parameters. In addition, motion vectors can also be employed to detect abrupt transitions.

7.2.1 Image Histogram and it's Moments

In this technique, the histograms of consecutive video frames are compared to detect shot changes. The (L^1) distance between histograms H of frames i and j is calculated using the following equation:

$$DoIH(i, j) = \sum_{k=0}^M |H_i(k) - H_j(k)| \quad (7.1)$$

where M is the total number of gray levels. If the distance is greater a pre-specified threshold, then an AT is declared.

Table 7.1

Shots with abrupt transition in Beverly Hills sequence

Sequence	Abrupt transition (AT) at frame #
Beverly Hills	13,17,29,46,54,80,88,125,141,155,170,187,200,221,252,269,282,318,344,358,382,419,435,457

It was shown in section 5.1.2 that Legendre moments effectively describe a histogram and hence can be employed for detecting AT's. The distance between two sets of modified Legendre moments corresponding to histograms of frames i and j is calculated using the following equation:

$$DoLMH(i, j) = \sum_{k=1}^{N_M} (\lambda_{i_k} - \lambda_{j_k})^2 \quad (7.2)$$

where N_M is the number of moments used for comparison.

Fig 7.2 shows the variation of $DoIH$ and $DoLMH$ (calculated using consecutive frames, i.e., $j = i + 1$) with respect to frame numbers. It is observed that both $DoIH$ and $DoLMH$ have peaks at most transitions.

7.2.2 DWT Coefficients

We note that *DoIH* and *DoLMH* are essentially pixel domain techniques that cannot be applied directly on compressed data. Here, we present a technique employing DWT coefficients. We note that the gray values of pixels of image frames from two different shots are likely to be substantially different. Since, lowpass DWT coefficients represent a thumbnail version of the original frame, these coefficients are also likely to be different for frames from two different shots. Hence, lowpass DWT coefficients can be used as a feature for detecting abrupt transitions. Since, the number of DWT coefficients is generally large, only a selected number of DWT coefficients are employed. Here, we propose to employ K blocks of DWT coefficients with each block containing 4 coefficients. The blocks are selected from a regular interval spanning the entire lowpass subimage. Block- k of frame i is compared with corresponding block of frame j using the following equation:

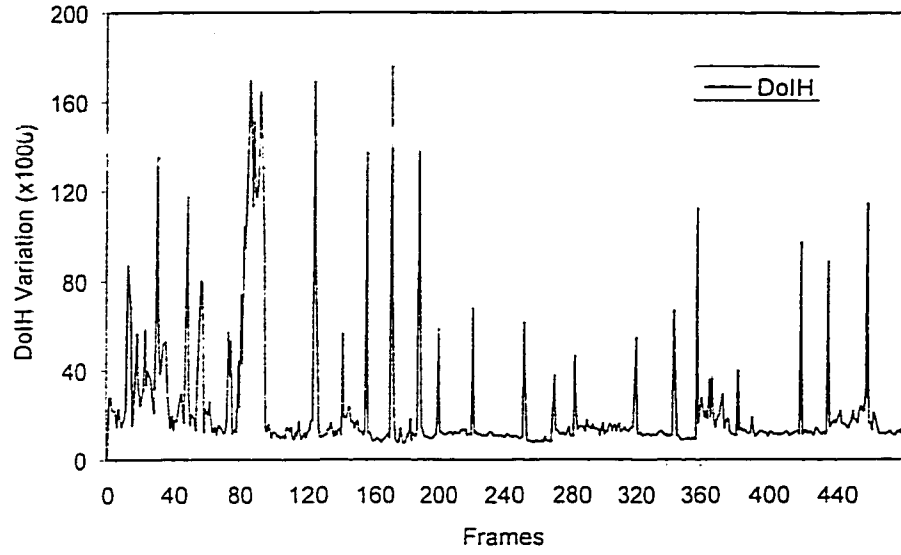
$$S_k(i, j) = \frac{1}{4} \sum_{l=1}^4 \frac{|c(i, k, l) - c(j, k, l)|}{|\max(c(i, k, l), c(j, k, l))|} \quad (7.3)$$

where $c(x, k, l)$ is the l th DWT coefficient of block- k in frame x . Block- k is declared similar if $S_k(i, j)$ is below a threshold, τ . $S_k(i, j)$ is thresholded to two levels according to the following criteria:

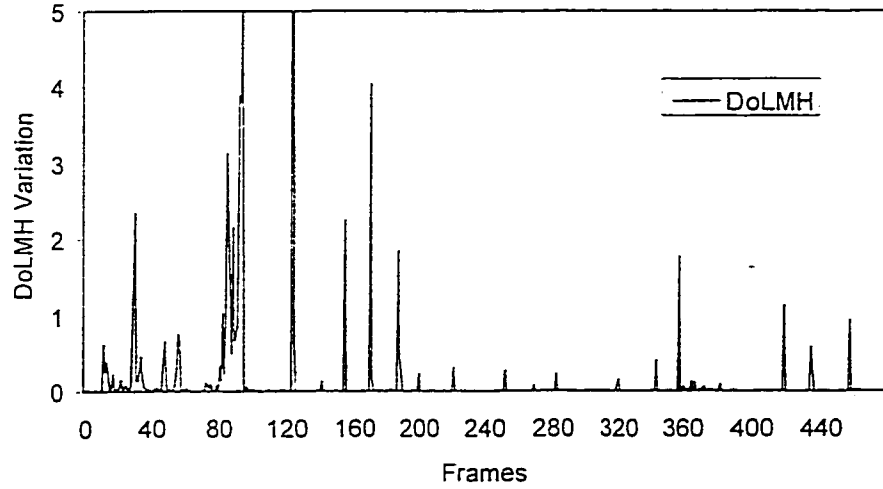
$$S_k(i, j) = \begin{cases} 1 & \text{if } S_k(i, j) \geq \tau \\ 0 & \text{Otherwise} \end{cases} \quad (7.4)$$

The distance between frames i and j is then estimated using the following equation:

$$DoDWTC(i, j) = \sum_{k=0}^K S_k(i, j) \quad (7.5)$$



(a)



(b)

Figure 7.2: DoIH (Difference of Image Histogram) and DoLMH (Difference of Legendre Moments of Histogram) variation with respect to frames in Beverly Hills sequence.

If $DoDWTC(i, i+1)$ exceeds a given threshold, an abrupt transition is likely to be present at frame $(i+1)$. Fig. 7.3 shows the variation of $DoDWTC$ with respect to frames. Peaks are observed at most of the transition frames. However, a significant fluctuation of $DoDWTC$ with respect to frames is also observed. We note that while difference in histograms is not

sensitive to small camera operations, such as panning, zooming, *etc*, a small camera operation affects *DoDWTC* values. For example, if there is a small panning (horizontal movement), almost all DWT coefficients will move horizontally, and a high *DoDWTC* may be observed.

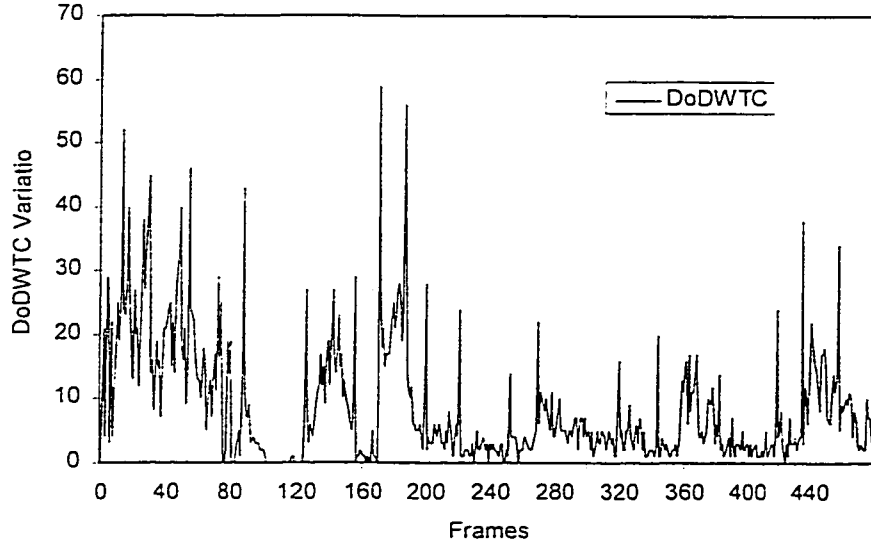


Figure 7.3: DoDWTC (Difference of selected DWT coefficients) variation with respect to frames in Beverly Hills sequence.

7.2.3 Wavelet Band Parameters

It was shown in section 5.2.2 that wavelet-band distribution parameters (namely, shape parameter, and standard deviation) can be effectively used for image indexing. Hence, there is a potential for the same parameters to be employed for detecting AT's. The distance between two sets of band parameters corresponding to frames i and j is calculated using the following equations:

$$\text{Difference of Standard Deviation, } DoSDWB(i, j) = \sum_{k=1}^Q A_k (\sigma_{i_k} - \sigma_{j_k})^2 \quad (7.6)$$

$$\text{Difference of Shape Parameter, } DoSPWB(i, j) = \sum_{k=1}^Q B_k (\gamma_{i_k} - \gamma_{j_k})^2 \quad (7.7)$$

where σ_k is the standard deviation of band k , γ_k is the shape parameter of band k . The weighted difference of both wavelet parameters can be expressed as:

$$DoWP(i, j) = \sum_{k=1}^Q A_k (\sigma_{i_k} - \sigma_{j_k})^2 + \sum_{k=1}^Q B_k (\gamma_{i_k} - \gamma_{j_k})^2 \quad (7.8)$$

Fig. 7.4 and 7.5 shows the variation of $DoSPWB$ and $DoSDWB$ with respect to frame numbers. It is observed that the difference peaks occur at most transitions.

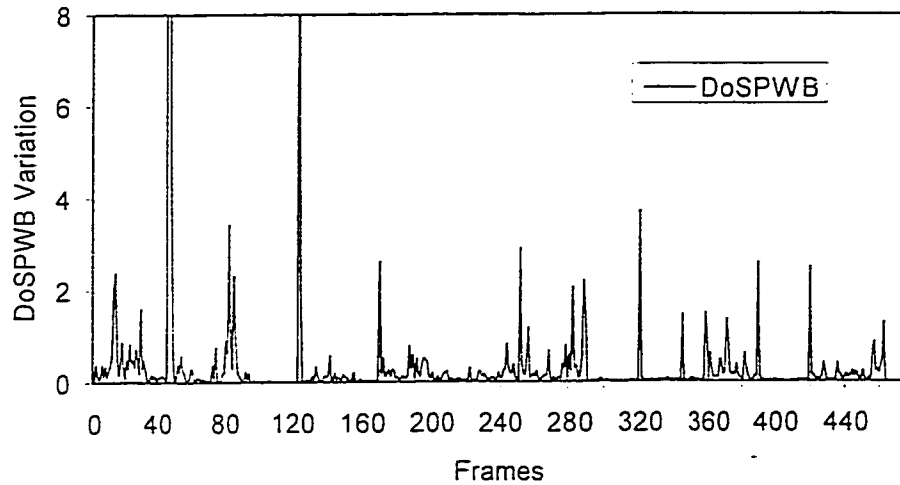


Figure 7.4: DoSPWB (Difference of shape parameters of wavelet band distribution) variation with respect to frames in Beverly Hills sequence.

7.2.4 Motion Vectors

In addition to the DWT coefficients and their distributions, object and camera motions are important features of video content. Within a single camera shot, the motion vectors generally show relatively small continuous changes. However, the continuity of motion vectors is disrupted between frames across different shots. Hence, the continuity of motion vectors [5] can serve as an effective criterion for detecting AT's. This is especially useful since most of the coding techniques perform motion compensation and hence, the motion vectors are already available.

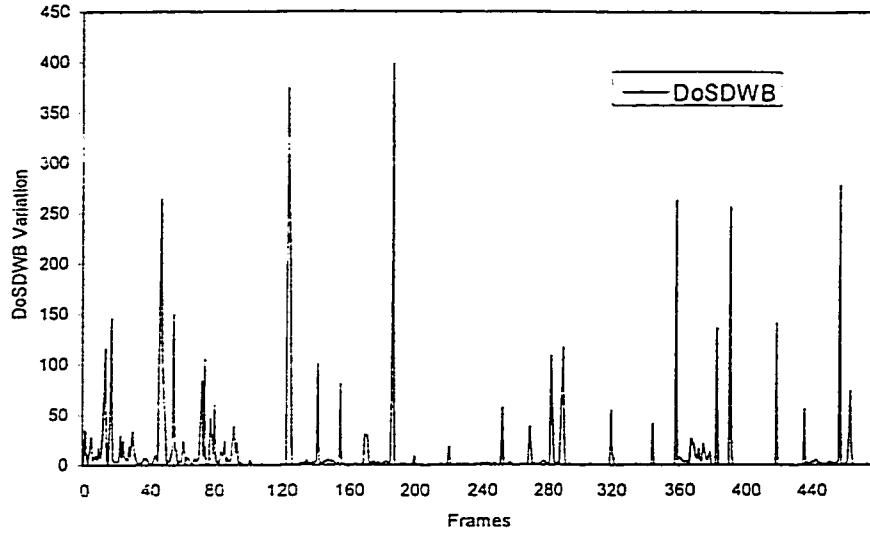


Figure 7.5: DoSDWB (Difference of standard deviation of wavelet band distribution) variation with respect to frames in Beverly Hills sequence.

Here, we outline a technique suitable for a GOP framework employing I, P, or B frames. We note that the GOP (as used in MPEG standard) duration is generally fixed for a sequence irrespective of the temporal change. Hence, the AT's can occur at frames that can be I-, P-, and B- frames. We note that I-frames are intracoded, and hence there are no associated motion vectors. On the other hand P-frames have only forward motion vectors, and B-frames have both forward and backward motion vectors. Hence, the criterion of deciding an AT at I-, P-, or B- frames will be different. We define the following parameters that will be useful for making a decision.

For P frames:

$$R_{MP} = \frac{\text{Number_of_Motion_Compensated_Blocks}}{\text{Total_Number_of_Blocks}} \quad (7.9)$$

For B- frames:

$$R_{FB} = \frac{\text{Number_of_Forward_Motion_Compensated_Blocks}}{\text{Number_of_Backward_Motion_Compensated_Blocks}} \quad (7.10)$$

7.2.4.1 Detecting Abrupt Transition using Motion Vectors

We assume that AT does not occur more than once between two consecutive reference frames. A frame[n] will be a candidate for abrupt transition if the following holds true.

Case-1: frame[n] is an I-frame

1. The $R_{FB}[n-1]$ (*i.e.*, R_{FB} corresponding to frame (n-1), which is a B-frame) is large.

Case-2: frame[n] is a P-frame

1. The $R_{MP}[n]$ is very small. and
2. The R_{FB} corresponding to the previous B-frame is large.

Case-3: frame[n] is a B-frame

1. The $R_{FB}[n]$ corresponding to the current frame will be small, and
2. If frame (n-1) is B-frame, $R_{FB}[n-1]$ is large, and
3. If frame (n+1) is P-frame, $R_{MP}[n+1]$ is small.

We note that appropriate thresholds should be selected in the above cases to achieve a good detection performance.

7.2.5 Indexing Performance of Individual Features

In this section, we evaluate the performance of various features for AT detection. We note that a technique may correctly detect an AT, may fail to detect an AT (*i.e.*, missed AT), or may detect an AT that does not exist (*i.e.*, false AT). An ideal video segmentation technique should not provide any false AT (FAT) or missed AT (MAT). However, the detection techniques are generally not perfect and hence the FATs or MATs are almost inevitable. Threshold techniques generally reduce the number of MATs by increasing the number of FATs, and vice versa. Hence, thresholds for each technique should be selected appropriately to balance the number of FATs and MATs.

We now compare the performance of the various pixel- and wavelet- domain features described in section 7.2. The DoIH features are generated from the image frames directly. DoLMH employs the first ten Legendre moments of an image histogram. DoDWTC employs 63 blocks of 4 coefficients, i.e., a total of 252 coefficients, spread over the entire lowpass subband. DoWP technique employs standard deviation and shape parameters from 9 highpass bands (see section 4.1.1.3). DoMV employs coarse motion vectors generated using ARS (L-3) technique detailed in section 4.5.1.

The AT detection performance on the Beverly Hills sequence is shown in Table 7.2. It is observed that DoIH, and DoLMH have detected 20, and 21 correct-AT, respectively. On the other hand, DoDWTC and DoWP have provided superior performance with 23 and 21 correct-AT, respectively. It is observed that DoDWTC technique provide the lowest number of FATs and MATs.

Table 7.2

Detection of camera breaks in Beverly Hills sequence using various algorithms. AT: Abrupt Transition, DoIH: Image Histogram, DoLMH: Legendre moments of histogram, DoDWTC: Selected lowpass DWT coefficients, DoWP: Statistical parameters of nine highpass wavelet bands.

Algorithm Type	Correct AT	False ATs	Missed ATs
DoIH	13,17,29,46,54,80,88,125,141,155,170,187,200,221,252,344,358,419,435,457	22,72,83,86,92	269,282,318,382
DoLMH	13,17,29,46,54,80,88,125,155,170,187,200,221,252,282,318,344,358,419,435,457	22,32,83,86,92	141,269,382
DoDWTC	13,17,29,46,54,88,125,141,155,170,187,200,221,252,269,282,318,344,358,382,419,435,457	25	80
DoWP	13,17,29,46,54,80,125,141,155,170,187,252,269,282,318,344,358,382,419,435,457	49,72,288,390,463	88,200,221

The performance of MV technique for AT detection is shown in Table 7.3. AT's were detected using coarse motion vectors estimated at different levels (see section 4.5.1). It is observed that the coarse motion vectors (at all levels) provide a good performance on AT detection. Although, there are 3-4 false AT's, only 1-3 AT's (out of 24) have been missed. Hence, MV technique is very useful in AT detection, especially as they are readily available from the video coder.

7.3 Fast Video Segmentation

The indexing performance of histogram (and its moments), DWT coefficients, wavelet band parameters, and motion vectors were evaluated in section 7.2. The performance was evaluated by comparing features from each individual frame. These features may not be available in the compressed bit-stream in which case it needs to be calculated. This increases the complexity of the segmentation technique. For example, if we want to employ DoDWTC technique, the lowpass DWT coefficients of P- and B- frames have to be generated first. To do this, we have to generate motion compensated DWT coefficients, and add them with the reconstructed error coefficients.

Table 7.3

Detection of AT's using motion vectors at different levels.

Motion Vectors	Correct AT	False AT	Missed AT
MV (L-0)	23	3	1
MV (L-1)	23	3	1
MV (L-2)	22	4	2
MV (L-3)	21	4	3

In this section, we propose a fast video segmentation technique (FVST) in the wavelet domain. We note that an AT generally occurs every few tens of frames, or even every few hundred frames. This is exploited by the proposed technique that segments the video in two passes. In the first pass, the feature parameters corresponding to I-frames are matched. Depending on the distance between the parameters, the GOP's are marked as containing potential AT's. In the second pass, a more detailed examination is carried out in the marked GOP's.

Among the features considered in this section, none except the wavelet parameters are available for I-frames. We recall from Chapter 4 that the video coder (see section 4.1.1.3) encodes the I-frame using the distribution parameters of wavelet bands. If the same coder is employed, the wavelet parameters of the I-frames are readily available. In the proposed technique, a GOP marked as having a potential AT is obtained by comparing the wavelet parameters of I-frames.

After the GOP's are marked in the first pass, the second pass tracks down the location of SC's within a GOP. We note that there may not be any SC in the concerned GOP, or there may be more than one SC. This can be determined by employing DoDWTC, DoWP, DoLMH, or MV technique. The relevant feature vector is generated for each frame, and the difference is calculated for SC detection. Two or more features can be employed to obtain a superior performance. The features can be added using two methods. First, the detected AT set can be considered as a union of AT sets detected employing individual features. In the second method, the intersection of the individual sets can be considered as the final AT set. We note that the first method generates an AT set with very small number of MATs, but also generates a large number of FATs. The second method, on the other hand, generates small

number of FATs, and large number of MATs. We note that the penalty for having MATs is generally more compared to having more FATs. Hence, we have implemented the first method in this thesis.

The flowchart of the FVST techniques is shown in Fig. 7.6. The performance of the FVST techniques is shown in Table 7.4. All techniques employ DoWP in the first pass. The technique employed in the second pass is shown the second column. We note that the number of MATs has almost been reduced to zero, but the number of FATs has increased.

7.4 Joint Coding and Indexing

A fast video segmentation technique has been proposed in the previous section. We note that efficient video coding techniques have been proposed in chapter 4 while image indexing techniques have been proposed in chapters 5 and 6. In this section, we combine all techniques and develop a video storage and retrieval system.

7.4.1 Video Coding

In chapter 4, we proposed several video coding techniques, such as AMRME, BMRME, ARS, ABA, and ARS+BMRME. In this section, we employ ARS+BMRME technique because of its superior coding performance. The GOP (group of pictures) employed for motion estimation is IBBBBPBBBBPBBBBI. The coarse and fine *thresh_factor*'s are 0.5 and 0.9, respectively. Daubechies-8 has been employed for video decomposition.

The video coding performance with Beverly Hills sequence is shown in Fig. 7.7. It is observed that an average PSNR of 32.97 dB is achieved at an average bit-rate of 0.24 *bpp*. We note that compressed bit-stream includes wavelet band parameters of I frames, and the

motion vectors. In addition, we also store selected DWT coefficients (see section 7.2.2) in a temporary archive. These coefficients are discarded when the video segmentation is completed.

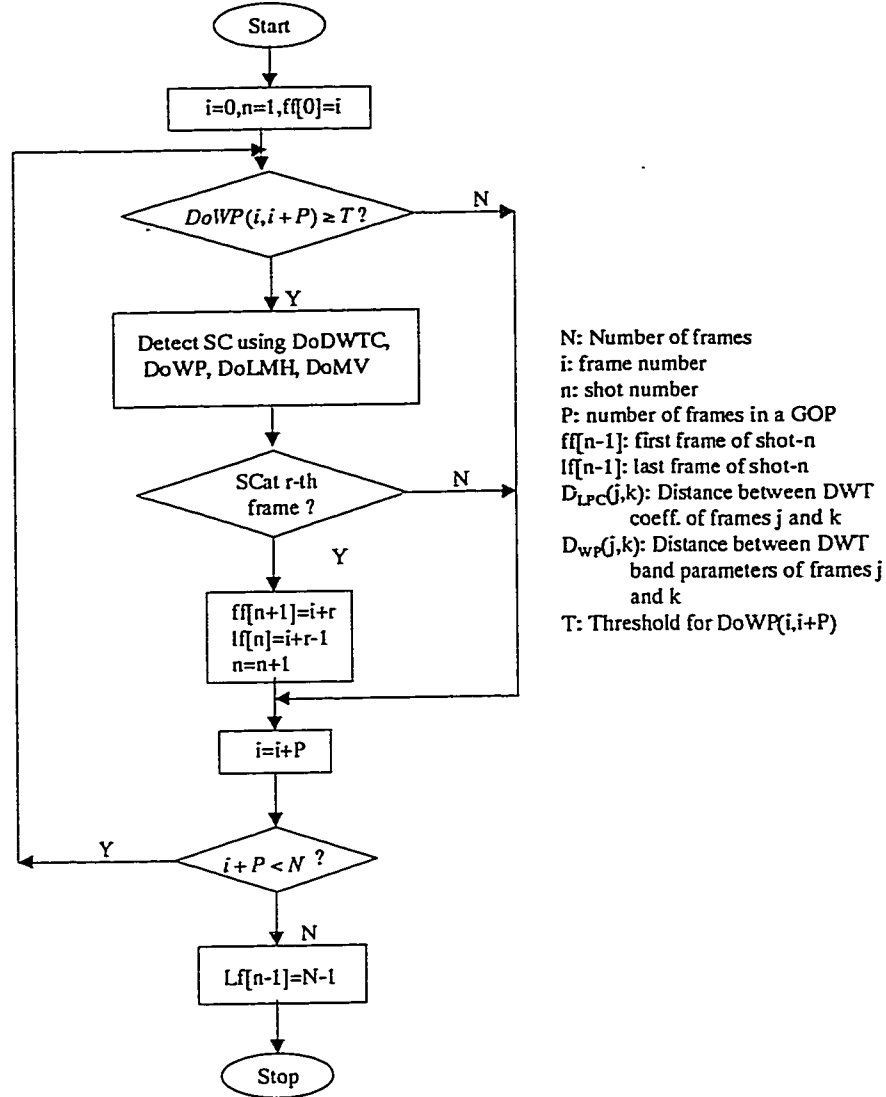


Figure 7.6: Flowchart of the proposed fast video segmentation technique in the wavelet domain.

7.4.2 Video Segmentation and Query Matching

Video segmentation can be executed along with video coding, or after video coding is completed. We note that the motion vectors and wavelet parameters have to be decoded (*i.e.*,

entropy decoded) when the segmentation is performed after the video coding. The complexity of entropy decoding can be avoided if segmentation is done along with the video coding. However, because of the frame-ordering (due to GOP) and the multipass nature of indexing techniques, the decision of an abrupt transition at a particular frame has to be delayed by several frames. In order to have a simple procedure, we perform indexing after the coding is completed.

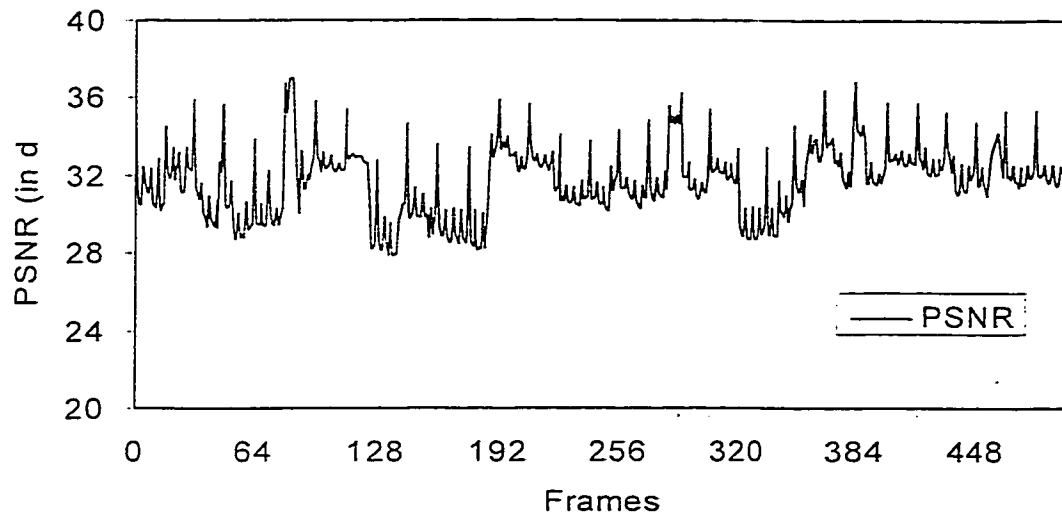
Table 7.4

Detection of camera breaks in Beverly Hills sequence using FVST techniques.

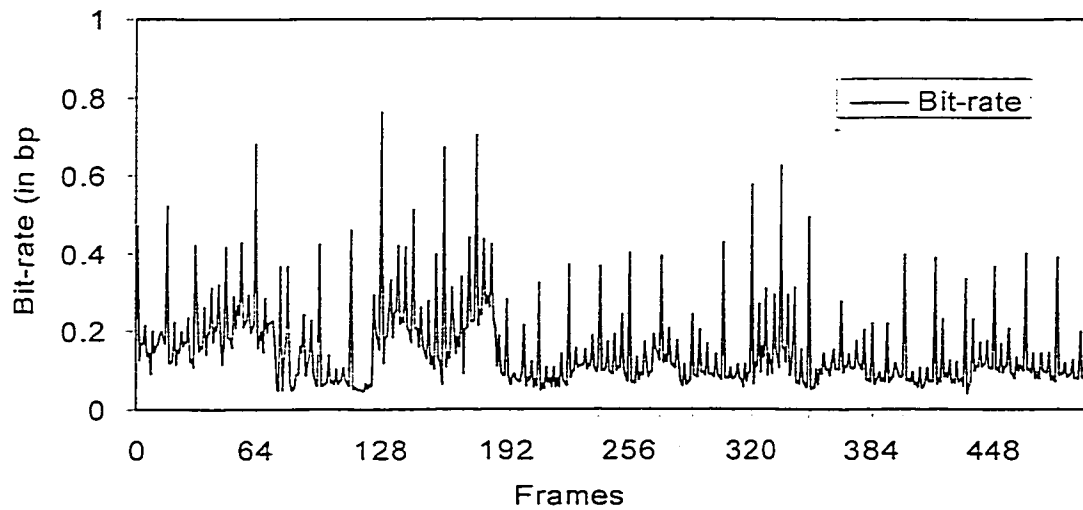
Algorithm Type	Algorithm (2 nd pass)	Correct ATs	False ATs	Missed ATs
FVST-1	DoDWTC	13,17,29,46,54,88,125,141,155,170,187,200,221,252,269,282,318,344,358,382,419,435,457	25	80
FVST-2	DoDWTC + DoMV	13,17,29,46,54,80,88,125,141,155,170,187,200,221,252,269,282,318,344,358,382,419,435,457	25,91,334,367,377	0
FVST-3	DoWP + DoDWTC	13,17,29,46,54,80,88,125,141,155,170,187,200,221,252,269,282,318,344,358,382,419,435,457	25,49,72,288,390,463	0
FVST-4	DoWP + DoMV	13,17,29,46,54,80,88,125,141,155,170,187,200,221,252,269,282,318,344,358,382,419,435,457	49,72,91,288,334,367,390,377,463	0

The fast video segmentation technique proposed in section 7.3 is employed to detect the AT's. Video shots are then formed between the consecutive AT's. After the shots are formed, a representative frame (RF) is chosen from each shot, which is then matched with a given query image. We note that the choice of RF in a shot is crucial in obtaining a good retrieval performance. Generally, the first frame of a shot is employed as the representative frame [26], although some authors [59] have also employed both the first and last frame. Here, we employ the middle frame of each shot as the representative frame. This is likely to

provide a better performance than the first frame, since the middle frame on average is likely to be more correlated to other frames in a shot.



(a)



(b)

Figure 7.7: Coding performance of Beverly Hills sequence frames (average bit-rate = 0.24 *bpp*, average PSNR 32.97dB).

It is observed from Table 7.4 that FVST-1 technique detects 24 AT's (including one FAT, and one MAT) in the Beverly Hills sequence. In otherwords, FVST-1 technique segments

the entire Beverly Hills video (of 480 frames) into 25 shots. The RF's, which are the middle frames of the shots, are shown in Fig. 7.8.

We now evaluate the performance of the proposed storage and retrieval system. A query image is shown in Fig. 7.9. In order to find images similar to the query, the query image is compared to all representative frames. Retrieval has been performed based on three criteria - i) histogram (DOIH), ii) moments of histogram and wavelet parameters (LGM+WP), and iii) translation and scale invariant moments and wavelet parameters (TSI-LGM+WP).

Based on visual inspection, it is obvious that RF-13 should be the closest match of the query image shown in Fig. 7.9. The actual images retrieved by the above mentioned three techniques are shown in Fig. 7.10. We observe that rank of RF-13 in the retrieved list is 2, 2, and 1 for DOIH, LGM+WP, and TSI-LGM+WP techniques, respectively. The simulation result suggests that RF-10 (which is shown in Fig. 7.10 (a1,b1)) is closer to the query image with respect to histogram than RF-13, and hence has a rank of 1 in the retrieved list. This is verified in Fig. 7.11 which plots histogram of query image, RF-13, and RF-10. It is observed that although the histogram envelopes of query image and RF-13 are similar (observe the peaks), histogram of the query image is translated towards right side compared to that of RF-13. On the other hand, the histogram envelopes of query image and RF-10 are not similar, but the individual frequency values are very close. As a result, RF-10 produces a histogram difference smaller than RF-13, and is wrongly retrieved by DOIH or LGM+WP techniques.

Fig. 7.10 (c1)-(c3) shows the images retrieved by TSI-LGM+WP technique. It is observed LGM+WP could correctly retrieve RF-13. This has been possible as the TSI-LGM+WP technique can compensate the translation and stretching of histogram which generally happens due to small illumination changes.



Figure 7.8: Representative frames (RF) from Beverly Hills sequence. The AT's were determined employing FVST-1 technique. F-0006 denotes that the RF is the 7th frame (frame index starts from zero) in the sequence.

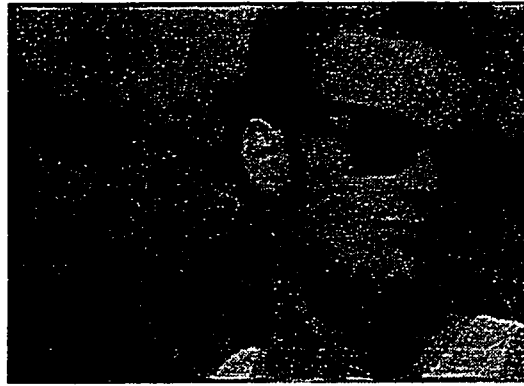


Figure 7.9: Query image

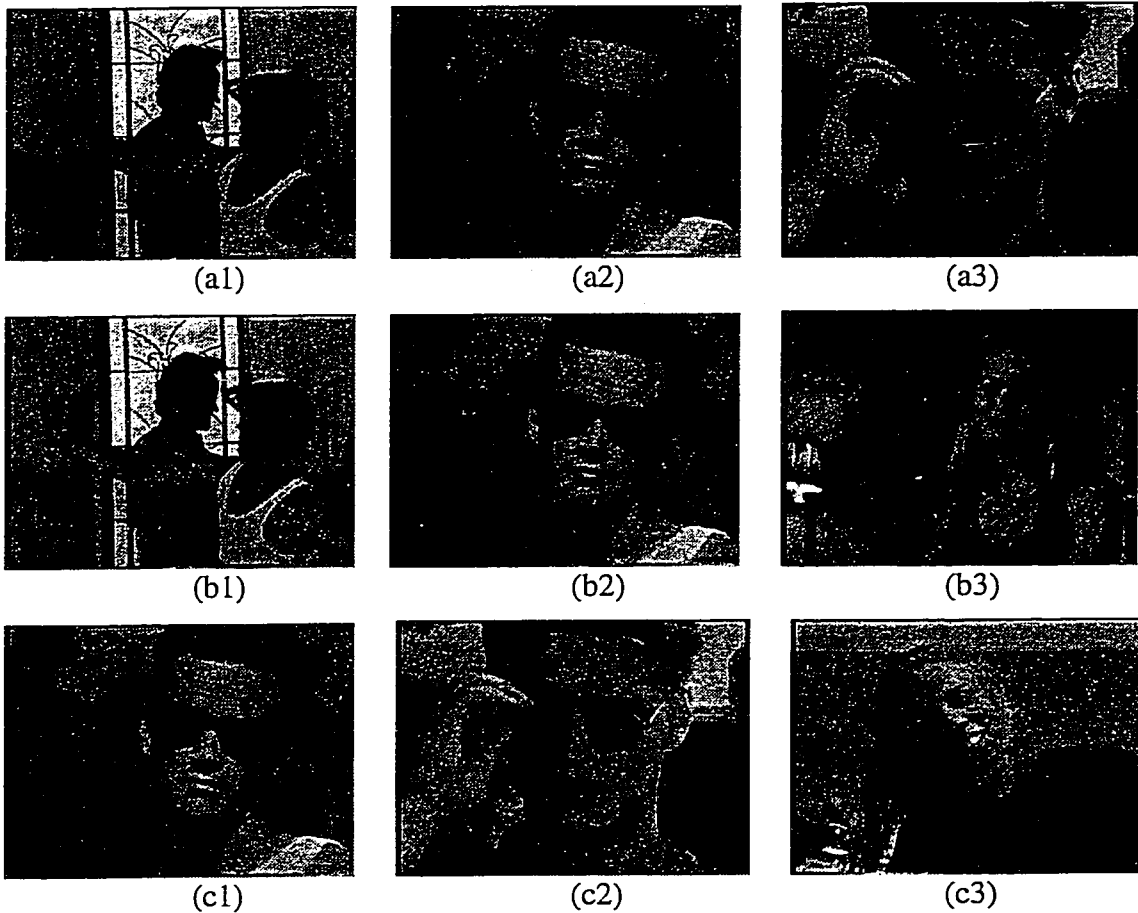


Figure 7.10: Retrieved representative frames from Beverly Hills sequence for query image shown in Fig. 7.9. a1-a3 represents first three frames (a1 being the first) retrieved employing histogram technique, b1-b3 has been retrieved employing joint moment and wavelet parameter technique, and c1-c3 have been retrieved employing TSI moment and wavelet parameter technique.

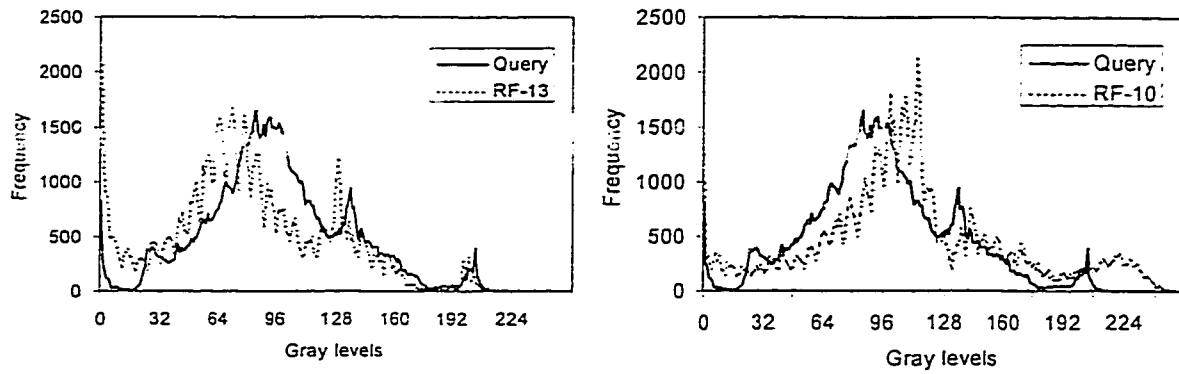


Figure 7.11: Histogram comparison of query image and representative frames (RF) 13 and 10.

7.5 Summary

In this chapter, we have presented a video storage and retrieval system in wavelet domain based on the techniques proposed in this thesis. First, we presented a typical architecture of a video storage and archival system. Next, we evaluated several pixel and wavelet features for video segmentation. We then proposed a fast video segmentation technique. The segmentation technique was then combined with coding and indexing techniques proposed earlier to provide the overall storage and archival system. The performance of the system was then evaluated for Beverly Hills sequence.

Chapter 8

Summary, Conclusions and Future Research Directions

8.1 Summary and Conclusions

Efficient data compression and indexing techniques are essential to meet the growing needs for multimedia data storage, transmission, and retrieval. Image and video compression standards, such as JPEG, MPEG-1, and MPEG-2, are based on discrete cosine transform [91]. Although, these standards provide a good coding performance, the performance degrades significantly at high compression ratios. In addition, these standards also lack content access functionality. Hence, the International Standards Organization (ISO) is presently working towards efficient multimedia standards, such as MPEG-4 (which is being finalized), and MPEG-7.

Recently, wavelet transform has been established as a potential technique for image and video coding. These techniques are being investigated for possible inclusion in JPEG-2000 and MPEG-4. However, a detailed investigation of motion estimation techniques in wavelet domain is required for efficient video coding. We also recall from chapter 1 that a majority of indexing techniques proposed in the recent literature operates in the pixel domain. These techniques are slow and do not exploit the compression parameters of the image/video stored

in a database. Hence, by jointly developing compression and indexing techniques, a superior performance can be achieved.

This thesis has proposed several novel compression and indexing techniques in the wavelet domain. We have also developed a joint compression and indexing scheme where both compression and indexing employ the same wavelet features. We hope this thesis will provide valuable inputs to the multimedia community, including the upcoming standards MPEG-7. The proposed techniques have been implemented and the simulation results demonstrate their superior performance. The contributions are in the following areas:

Motion Estimation and Video Compression

- We reviewed the limitation of motion compensation in the wavelet domain. A thresholding technique has been developed (in section 4.3) to adaptively discard the failed motion blocks.
- A bi-directional motion estimation technique, suitable for wavelet domain has been developed (section 4.4). This technique provides a good coding performance employing GOP (group of pictures) framework.
- An adjustable resolution selection technique (section 4.5.1) has been developed to further improve the coding performance. Here, coarse motion estimation is executed at a selected resolution employing unquantized DWT coefficients of present and previous reference frames.
- We have also developed an adaptive bit-allocation technique, (section 4.5.2) where the size of the motion blocks is determined adaptively to distribute bits optimally between the motion vectors and predictive error frames employing a Lagrangian cost function.

Indexing of Images and Video

- We first developed low complexity indexing techniques based on moments of image histograms. It was shown that Legendre moments (section 5.1.2) provide a good performance at reduced complexity.
- Fast wavelet histogram techniques (section 5.2.1) providing a superior indexing performance for texture images were then proposed.
- We have developed an indexing technique (section 5.2.2) based on the distribution of wavelet coefficients at different bands. The simulation results have shown that the wavelet and image-histogram techniques provide a good indexing performance for texture, and general images, respectively. Subsequently, a joint moment and wavelet technique that provides a superior performance for both types of images has been developed.
- A translation and scale invariant moment and wavelet technique (section 6.3) that is robust to changes in illumination has been proposed.
- A joint video coding and indexing scheme (chapter 7) has been proposed for an efficient video storage and archival system based on a combination of the above systems.

8.2 Future Research Directions

In this thesis, we have addressed several crucial issues in wavelet-based coding and indexing. However, many research challenges still remain to be investigated. Some of the unresolved issues that follow from this work are:

- The choice of wavelet bases (the filter coefficients, number of decomposition levels, *etc.*) for joint motion estimation and spatial coding needs to be investigated.
- Most coding and indexing techniques proposed in this thesis are based on gray level images. Their adaptation to color images needs to be explored.
- Wavelets provide edge information at various scales. There is a potential to exploit this information to obtain a superior coding and indexing performance.
- Joint 3-D wavelet coding and indexing is another potential area of research.
- Various implementation issues involving hardware design and computational complexity need to be evaluated.

We note that most of the current storage and retrieval techniques are based on features, such as texture, color, and sketch which are not very efficient. The human brain, although not well understood, is believed to process an image at a semantic level. Hence, the ultimate objective should be to develop techniques that emulate the human brain. Although, it is difficult to develop these techniques with current technology, we hope that future research will enable us to achieve this goal.

References

1. G. Ahanger and T. D. C. Little, "Survey of technologies for parsing and indexing digital video," *Journal of Visual communications and Image Representation*, Vol. 7, pp. 28-43, March 1996.
2. P. Aigrain, H. Zhang and D. Petkovic, "Content-based representation and retrieval of visual media: a state of the art review," *Multimedia Tools and Applications* 3, pp. 179-202, 1996.
3. K. Aizawa and T. S. Huang, "Model-based image coding: advanced video coding techniques for very low bit-rate applications," *Proceedings of the IEEE*, Vol. 83, pp. 259-271, Feb. 1995.
4. A. N. Akansu and R. A. Haddad, *Multiresolution Signal Decomposition: Transform, Subbands and Wavelets*, Academic Press, 1993.
5. A. Akutsu *et al.*, "Video indexing using motion vectors," *Proc. of SPIE: Visual Communications and Image Processing*, Vol. 1818, pp. 1522-1530, 1992.
6. M. Antonini *et al.*, "Image coding using wavelet transform," *IEEE Trans. on Image Processing*, Vol. 1, pp. 205-220, April 1992.
7. F. Arman, A. Hsu and M. Y. Chiu, "Image processing on compressed data for large video databases," *Proc. of SPIE: Storage and Retrieval for Image and Video Databases*, pp. 267-272, 1993.
8. N. Baaziz and C. Labit, "Multigrid motion estimation on wavelet pyramids for image sequence coding," in: Johansen and Olsen, Eds, *Theory and Applications of Image Analysis - Selected papers from the 7th Scandinavian Conference on Image Analysis*, 1992.
9. E. A. Barria and J. M. Jagadeesh, "Multiresolution motion estimation using the wavelet transform," *Proc. of SPIE: Wavelet applications in Signal and Image Processing II*, Vol. 2303, pp. 542-553, 1994.

10. M. Bierling, "Displacement estimation by hierarchical block-matching," *Proc. of SPIE: Visual Communications and Image Processing*, Vol. 1001, pp. 942-951, Boston, Nov 1988.
11. A. D. Bimbo, P. Pala and S. Santini, "Visual image retrieval by elastic deformation of object sketches," *Proc. of the IEEE Symposium on Visual Languages*, pp. 216-223, 1994.
12. K. A. Birney and T. R. Fischer, "On the modelling of DCT and subband image data for compression," *IEEE Trans. on Image Processing*, Vol. 4, pp. 186-193, Feb. 1995.
13. C. Caffario *et al.*, "Motion compensation and multiresolution coding," *Signal Processing: Image Communication*, Vol. 6, pp. 123-142, May 1994.
14. S. K. Chang, C. M. Lee and C. R. Dow, "A 2-D string matching algorithm for conceptual pictorial queries," *Proc. of SPIE: Image Storage and Retrieval Systems*, Vol. 1662, pp. 47-58, Feb 1992.
15. A. Celentano and V. D. Lecce, "A FFT based technique for image signature generation," *Proc. of SPIE: Storage and Retrieval for Image and Video Databases V*, Vol. 3022, pp.457-466, Feb 1997.
16. T. Chang and C. C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Trans. on Image Processing*, Vol. 2, pp. 429-441, Oct. 1993.
17. R. R. Coifman and M. V. Wickerhauser, "Entropy based algorithms for best basis selections," *IEEE Trans. on Information Theory*, Vol. 38, pp. 713-718, March 1992.
18. G. J. Conklin and S. S. Hemami, "Multiresolution motion estimation," *Proc. of ICASSP*, pp. 2873-2876, 1997.
19. I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, 1992.
20. R. A. Devore, B. Jawerth and B. J. Lucier, "Image compression through wavelet coding," *IEEE Trans. on Information Theory*, Vol. 38, pp. 719-746, March 1992.
21. Digital Libraries, Special issue of *Communications of the ACM*, Vol. 38, April 1995.
22. M. O'Docherty and C. Daskalakis, "Multimedia information systems: the management and semantic retrieval of all electronic datatypes," *The Computer Journal*, Vol. 34, pp. 225-238, 1991.

23. M. Ehlers, G. Edwards and Y. Bedard, "Integration of remote sensing with geographic information systems: a necessary evolution," *Photogrammetric Engineering and Remote Sensing*, Vol. 55, pp. 1619-1627, 1989.
24. M. Flickner *et al.*, "Query by image content: the QBIC system," *IEEE Computer*, Vol. 28, pp. 23-32, Sept 1995.
25. B. V. Funt and G. D. Finlayson, "Color constant color indexing," *IEEE Trans. on PAMI*, Vol. 17, pp. 522-529, May 1995.
26. B. Furht, S. W. Smoliar and H. Zhang, *Video and Image Processing in Multimedia Systems*, Kluwer Academic Publishers, 1995.
27. D. L. Gall, "MPEG: a video compression standard for multimedia applications," *Communications of the ACM*, Vol. 34, pp. 46-58, April 1991.
28. H. Gharavi, "Subband coding of video signals," in *Subband Image Coding*, J. Woods Ed., Kluwer, 1992.
29. R. C. Gonzalez and R. C. Woods, *Digital Image Processing*, Addison Wesley, 1992.
30. A. Grzeszczak, M. K. Mandal, S. Panchanathan and T. Yeap, "VLSI Implementation of Discrete Wavelet Transform" *IEEE Trans. on VLSI Systems*, Vol. 4, pp. 421-433, Dec. 1996.
31. J. Hafner *et al.*, "Efficient color histogram indexing for quadratic form distance function," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 17, pp. 729-736, July 1995.
32. T. Hamano, "A similarity retrieval method for image databases using simple graphics," *Proc. of the IEEE Workshop on Languages for Automation, Symbiotic and Intelligent Robotics*, pp. 149-154, University of Maryland, Aug. 1988.
33. H. A. Hampel *et al.*, "Technical features of the JBIG standard for progressive bi-level image compression," *Signal Processing: Image Communications*, Vol. 4, pp. 103-111, April 1992.
34. G. Heising, "A wavelet-based video coding scheme using image warping prediction," *Proc. of IEEE International Conference on Image Processing (ICIP'98)*, Chicago, Illinois, Oct. 1998.

35. K. Hirata and T. Kato, "Rough sketch-based image information retrieval," *NEC Research and Development*, Vol. 34, pp. 263-273, April 1993.
36. M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. on Information Theory*, Vol. 69, pp. 179-187, Feb 1962.
37. J. Huang, Z. Liu, and Y. Wang, "Integration of audio and visual information for content-based video segmentation," *Proc. of IEEE International Conference on Image Processing (ICIP'98)*, Chicago, Illinois, Oct. 1998.
38. T. Ida and Y. Sambonsugi, "Image segmentation using fractal coding," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 5, pp. 567-570, Dec. 1995.
39. F. M. Idris and S. Panchanathan, "Image indexing using wavelet vector quantization," *Proc. of SPIE: Digital Image Storage and Archiving Systems*, Vol. 2606, pp. 269-275, Philadelphia, Oct. 1995.
40. F. Idris and S. Panchanathan, "Indexing of compressed video sequences," *Proc. of SPIE*, Vol. 2670, pp. 247-253, 1996.
41. F. Idris and S. Panchanathan, "Review of image and video indexing techniques," *Journal of Visual Communication and Image Representation*, Vol. 8, pp. 146-166, June 1997.
42. *IEEE Trans. on Circuits and Systems for Video Technology, Special issue on MPEG-4*, Vol. 7, Feb 1997.
43. M. Irani and P. Anandan, "Video indexing based on mosaic representations," *Proc. of IEEE*, Vol. 86, pp. 905-921, May 1998.
44. ISO/IEC 11172-2:1993 (MPEG-1), *Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s - Part 2: Video*, 1993.
45. ISO/IEC DIS 13818-2:1996 (MPEG-2), *Information Technology - Generic Coding of Moving Pictures and Associated Audio Information: Video*, 1996.
46. ISO/IEC JTC1/SC29/WG1, Document N390R, *New Work Item: JPEG 2000 image coding system*, March 21, 1997.
47. ISO/IEC JTC1/SC29/WG11, Document N2207, *MPEG-7: Context and Objectives (Version 7 - San Jose)*, March, 1998.

48. C. E. Jacobs, A. Finkelstein and D. H. Salesin, "Fast multiresolution image querying," *Proc. of ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques*, pp. 277-286, Los Angeles, Aug 1995.
49. A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.
50. J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. on Communications*, Vol. COM-29, pp. 1799-1806, Dec. 1981.
51. N. S. Jayant, J. Johnston and R. Sofranek, "Signal compression based on models of human perception," *Proc. of IEEE*, Vol. 81, pp. 1385-1421, Oct. 1993.
52. N. S. Jayant and P. Noll, *Digital coding of Waveforms: Principles and Applications to Speech and Video*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.
53. R. Kasturi and R. Jain, "Dynamic Vision," *Computer Vision: Principles*, Eds. R. Kasturi and R. Jain, pp. 469-480, IEEE Computer Society Press, Washington, 1991.
54. S. Kim, S. Ree, J. G. Jeon, and K. T. Park, "Interframe coding using two-stage variable block-size multiresolution motion estimation and wavelet decomposition," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 8, pp. 399-410, Aug. 1998.
55. R. Koenen, F. Pereira, and L. Chiariglione, "MPEG-4: context and objectives," *Signal Processing: Image Communications*, Vol. 9, pp. 295-304, May 1997.
56. T. Koga *et al.*, "Motion compensated interframe coding for video conferencing," *Proc. NTC'81*, pp. G5.3.1-G5.3.5, 1981.
57. E. Kreyszig, *Advanced Engineering Mathematics*, Wiley, New York, 1988.
58. M. Kunt, A. Ikonomopoulos and M. Kocher, "Second generation image coding techniques," *Proc. of the IEEE*, Vol. 73, pp. 549-574, April 1985.
59. J. Lee and B. W. Dickinson, "Multiresolution video indexing for subband coded video databases," *Proc. of SPIE: Storage and Retrieval for Image and Video Databases*, Vol. 2185, San Jose, CA, Feb. 1994.
60. Y. Linde, A. Buzo and R. Gray, "An algorithm for vector quantizer design", *IEEE Trans. on Communications*, Vol. COM-28(1), pp. 84-95, Jan 1980.
61. G. Ling, "Wavelet image modelling and its application in model-based image coding," *Proc. of SPIE: VCIP*, Vol. 2308, pp. 526-534, 1994.

62. T. D. C. Little *et al.*, "A digital video-on-demand service supporting content-based queries," *Proc. of 1st International Conference on Multimedia*, pp. 427-436, Anaheim CA, Aug. 1993.
63. H. C. H. Liu and G. L. Zick, "Scene decomposition of MPEG compressed video," *Proc. of SPIE: Digital Video Compression: Algorithms and Technologies*, Vol. 2419, pp. 26-37, Feb 1995.
64. S. G. Mallat, "A theory for multiresolution signal representation: the wavelet decomposition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, pp. 674-693, July 1989.
65. M. K. Mandal, *Wavelets for Image Compression*, M.A.Sc. Thesis, University of Ottawa, Ottawa, Canada, 1995.
66. M. K. Mandal, S. Panchanathan and T. Aboulnasr, "Wavelet-based image coding using HVS characteristics" *Proc. of SPIE: Wavelets Applications in Signal and Image Processing III*, Vol. 2569, pp. 345-352, San Diego, July 1995.
67. M. K. Mandal, E. Chan, X. Wang and S. Panchanathan, "Multiresolution motion estimation techniques for video compression," *Optical Engineering*, Vol. 35, pp. 128-136, Jan 1996.
68. M. K. Mandal, T. Aboulnasr and S. Panchanathan, "Image indexing using moments and wavelets," *IEEE Trans. on Consumer Electronics*, Vol. 42, pp. 557-565, Aug 1996.
69. M. K. Mandal, S. Panchanathan and T. Aboulnasr, "Choice of wavelets for image compression," *Lecture Notes in Computer Science*, Vol. 1133, pp. 239-249, Springer Verlag, 1996.
70. M. K. Mandal, S. Panchanathan, T. Aboulnasr, "Image indexing using translation and scale invariant moments and wavelets", *Proc. of SPIE: Storage and Retrieval for Image and Video Databases V*, San Jose, California, Feb 13-14, 1997.
71. M. K. Mandal, T. Aboulnasr, and S. Panchanathan, "Adaptive multiresolution motion estimation techniques for wavelet-based video coding," *Proc. of SPIE: Visual Communications and Image Processing*, Vol. 3309, pp. 965-974, San Jose, Jan 1998.

72. M. K. Mandal, F. Idris, and S. Panchanathan, "A critical evaluation of image and video indexing techniques in the compressed domain," *Image and Vision Computing Journal*, Vol. 16, 1998.
73. M. K. Mandal, S. Panchanathan and T. Aboulnasr, "Illumination invariant image indexing using moments and wavelets," *Journal of Electronic Imaging*, Vol. 7, pp. 282-293, April 1998.
74. M. K. Mandal, S. Panchanathan, and T. Aboulnasr, "Fast wavelet histogram techniques for image indexing," *Proc. of IEEE Workshop on Content-Based Access of Image and Video Libraries*, Santa Barbara, California, June 1998.
75. M. K. Mandal, T. Aboulnasr and S. Panchanathan, "Video indexing in wavelet compressed domain," *Proc. of IEEE International Conference on Image Processing (ICIP'98)*, Chicago, Illinois, Oct. 1998.
76. B. M. Mehtre *et al.*, "Color matching for image retrieval," *Pattern Recognition Letters*, Vol. 16, pp. 325-331, March 1995.
77. M. Liou, "Overview of the px64 kbit/s video coding standard," *Communications of the ACM*, Vol 34, pp. 59-63, April 1991.
78. H. G. Musmann, P. Pirsch and H. J. Grallert, "Advances in picture coding," *Proc. of the IEEE*, Vol. 73, pp. 523-548, April 1985.
79. A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-video search for object appearance," *IFIP: Visual Database Systems II*, pp. 113-127, 1992.
80. H. Nakano, "Classifying motions of objects in digital video sequences for automatic indexing," *SPIE 43rd Annual Meeting*, Vol. 3458, San Diego, CA, July 1998.
81. J. Nam and A. H. Tewfik, "Progressive resolution motion indexing of video object," *Proc. of ICASSP*, Vol. VI, pp. 3701-3704, Seattle, WA, May 1998.
82. T. Naveen and J. W. Woods, "Motion compensated multiresolution transmission of high definition video," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 4, pp. 29-41, Feb. 1994.
83. K. Panusopone and K. R. Rao, "Efficient motion estimation for block-based video compression," *Proc. of ICASSP*, Part 4, pp. 2677-2680, Munich, 1997.

84. A. P. Pentland, R. W. Picard and S. Scarloff, "Photobook: tools for content-based manipulation of image databases," *Proc. of SPIE: Storage and retrieval for Image and Video Databases II*, Vol. 2185, pp. 34-47, San Jose, 1994.
85. R. W. Picard and F. Liu, "A new Wold ordering for image similarity", *Proc. of the International Conference on Acoustics, Speech and Signal Processing*, Vol. V, pp. 129-132, April 1994.
86. R. J. Prokop and A. P. Reeves, "A survey of moment-based techniques for unoccluded object representation and recognition," *CVGIP: Graphical Models and Image Processing*, Vol. 54, pp. 438-460, Sept. 1992.
87. F. Qi, D. Shen and L. Quan, "Wavelet transform based rotation invariant feature extraction in object recognition," *Proc. of Intl. Symp. on Information Theory and its Applications*, pp. 221-224, Nov 1994.
88. R. D. Queiroz, C. K. Choi, Y. Huh, and K. R. Rao, "Wavelet transforms in a JPEG-like image coder," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 7, pp. 419-424, April 1997.
89. K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. on Image Processing*, Vol. 2, pp. 160-175, April 1993.
90. A. R. Rao, N. Bhushan, and G. L. Lohse, "The relationship between texture terms and texture images: A study in human texture perception," *Proc. of SPIE: Storage and Retrieval for Still Image and Video Databases IV*, Vol. 2670, pp. 206-214, Feb 1996.
91. K. R. Rao and J. J. Hwang, *Techniques and Standards for Image, Video and Audio Coding*, Prentice Hall PTR, New Jersey, 1996.
92. P. Rashkovskiy and L. Sadovnik, "Scale, rotation and shift invariant wavelet transform," *Proc. of SPIE: Optical Pattern Recognition V*, Vol. 2237, pp. 390-401, 1994.
93. E. Reusens and T. Ebrahimi, "New techniques for subband/wavelet transform coefficient coding applied to still image compression," *Proc. of SPIE: Applications of Digital Image Processing XV*, Vol. 1771, pp. 444-457, 1992.
94. Y. Rui, T. Huang, and S. -F. Chang, "Digital image/video library: standardization and research issues," *Proc. of ICASSP*, Vol. VI, pp. 3785-3788, Seattle, WA, May 1998.

95. A. Said, and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 6, pp. 243-250, June 1996.
96. M. Sajadieh and A.N. Venetsanopoulos, "Wavelet-based methods in image compression", *Proc. of the Canadian Conference on Electrical and Computer Engineering*, pp. MM4.5.1 - MM4.5.4, Toronto, Sept 1992.
97. B. Shahraray, "Scene change detection and content based sampling of video sequences," *Proc. of SPIE: Digital Video Compression: Algorithms and Technologies*, Vol. 2419, pp. 2-13, Feb 1995.
98. C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *IRE National Convention Record*, Part 4, pp. 142-163, 1959.
99. J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Processing*, Vol. 41, pp. 3455-3462, Dec. 1993.
100. K. Sharifi and A. L. Garcia, "Estimation of shape parameter for generalized Gaussian distributions in subband decompositions of video," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 5, pp. 52-56, Feb. 1995.
101. M. Shneier and M. A. Mottaleb, "Exploiting the JPEG compression scheme for image retrieval," *IEEE Trans on Pattern Analysis and Machine Intelligence*, Vol. 18, pp. 849-853, Aug 1996.
102. J. R. Smith and S. F. Chang, "Transform features for texture classification and discrimination in large image databases," *Proc. of International Conference on Image Processing*, pp. 407-411, 1994.
103. J. R. Smith and S. F. Chang, "Automated binary texture feature sets for image retrieval," *Proc. of ICASSP*, Atlanta, May 1996.
104. R. Srinivasan and K. R. Rao, "Predictive coding based on motion estimation," *IEEE Trans. on Communications*, Vol. COM-33, pp. 888-896, Aug 1985.
105. H. S. Stone, and C. S. Li, "Image matching by means of intensity and texture matching in the Fourier domain," *Proc. of SPIE*, Vol. 2670, pp. 337-349, 1996.
106. M. A. Stricker, "Bounds for the discrimination power of color indexing techniques," *Proc. of SPIE*, Vol. 2185, pp. 15-24, 1994.

107. M. Stricker and M. Orengo, "Similarity of color images," *Proc. of SPIE: Storage and Retrieval for Image and Video Databases III*, Vol. 2420, pp. 381-392, San Jose, Feb 1995.
108. M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, Vol. 7, pp. 11-32, 1991.
109. M. D. Swanson, S. Hosur and A. H. Tewfik, "Image coding for content-based retrieval," *Proc. of SPIE: VCIP*, Vol. 2727, pp. 4-15, 1996.
110. H. Tamura and N. Nokoya, "Image database systems: A survey", *Pattern Recognition*, Vol. 17, pp. 29-43, 1984.
111. M. R. Teague, "Image analysis via the general theory of moments," *Journal of Optical Society of America*, Vol. 70, pp 920-30, August 1990.
112. D. Tegolo, "Shape analysis for image retrieval," *Proc. of SPIE: Storage and Retrieval for Image and Video databases II*, Vol. 2185, pp. 59-69, 1994.
113. A. M. Tekalp, *Digital Video Processing*, Prentice Hall, New Jersey, 1995.
114. C. Tomlin, *Geographic Information Systems and Cartographic Modeling*, Prentice Hall, Englewood Cliffs, NJ., 1990.
115. M. J. Tsai, J. D. Villasenor, and F. Chen, "Stack-run image coding," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 6, pp. 519-521, Oct. 1996.
116. M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, Vol. 3, pp. 73-86, 1991.
117. K. M. Uz, M. Vetterli and D. J. Legall, "Interpolative multiresolution coding of advanced television with compatible subchannels," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 1, pp. 86-99, March 1991.
118. A. Vellaikal, C. -C. J. Kuo and S. Dao, "Content-based retrieval of remote-sensed images using vector quantization," *Proc. of SPIE*, Vol. 2488, pp. 178-189, 1995.
119. S. Vinayagamoorthy *et al.*, "A multichannel filter for TV signal processing", *IEEE Trans. on Consumer Electronics*, Vol. 42, pp. 199-206, May 1996.
120. G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM*, Vol. 34, pp. 30-45, April 1991.

121. D. Wang and L. Wang, "Global motion parameter estimation using a fast and robust algorithm," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 7, pp. 823-826, Oct 1997.
122. J. Z. Wang *et al.*, "Wavelet-based image indexing techniques with partial sketch retrieval capability," *Proc. of the Forum on Research and Technology Advances in Digital Libraries*, pp. 13-24, Washington DC, May 1997.
123. X. Wang, E. Chan, M. K. Mandal and S. Panchanathan, "Wavelet-based image coding using nonlinear interpolative vector quantization," *IEEE Trans. on Image Processing*, Vol. 5, pp. 518-522, March 1996.
124. I. H. Witten, R. M. Neal and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, Vol 30, pp 520-540, June 1987.
125. J. W. Woods and S. D. O'Neil, "Subband coding of images," *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. 34, pp. 1278-1288, Oct 1986.
126. J. Woods, *Subband Image Coding*, Kluwer Academic Publishers, 1991.
127. J. K. Wu *et al.*, "CORE: a content-based retrieval engine for multimedia information systems," *Multimedia Systems*, Vol. 3, pp. 25-41, 1995.
128. Z. Xiong *et al.*, "Space-frequency quantization for a space-varying wavelet packet image coder," *Proc. of ICIP*, pp. 614-617, Oct 1995.
129. S. Zafar, Y. Zhang, and B. Jabbari, "Multiscale video representation using multiresolution motion compensation and wavelet decomposition," *IEEE Journal of Selected Areas in Communications*, Vol. 11, pp. 24-35, Jan. 1993.
130. W. R. Zettler, J. Huffman and D. C. P. Linden, "Application of compactly supported wavelets to image compression," *Proc. of SPIE: Image Processing Algorithms and Techniques*, Vol. 1244, pp. 150-160, 1990.
131. A. Zhang *et al.*, "Comparison of wavelet transforms and fractal coding in texture-based image retrieval," *Proc. of SPIE: Visual Data Exploration and Analysis III*, Vol. 2656, pp. 116-125, 1996.
132. H. Zhang and D. Zhong, "A scheme for visual feature based image indexing", *Proc. of SPIE: Storage and Retrieval for Image and Video Databases III*, Vol. 2420, pp. 36-46, Feb 1995.

133. H. Zhang, C. Y. Low and S. W. Smoliar, "Video parsing and browsing using compressed data," *Multimedia Tools and Applications*, Vol. 1, pp. 89-111, 1995.
134. H. J. Zhang *et al.*, "Automatic partitioning of full motion video," *ACM Multimedia Systems*, Vol. 1, pp. 10-28, 1993.
135. Y. Q. Zhang and S. Zafar, "Motion-compensated wavelet transform coding for color video compression," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 2, pp. 285-296, Sept 1992.

Appendix A

Wavelet Bases and Filter Coefficients

In this thesis, proposed compression and indexing techniques have been evaluated using several Daubechies wavelets [19] with different supports. The filter coefficients are given below. N refers to the number of taps, L refers to the number of vanishing moments of the wavelet. All the filter coefficients are normalized so that $\sum h[n] = \sqrt{2}$.

Table A.1

Minimum phase Daubechies wavelets for N=2,4,6,8,10,12, and 16 taps.

	n	h[n]
N=2 L=1	0	0.70710678
	1	0.70710678
N=4 L=2	0	0.482962913144
	1	0.836516303737
	2	0.224143868042
	3	-0.129409522551
N=6 L=3	0	0.332670552950
	1	0.806891509311
	2	0.459877502118
	3	-0.135011020010
	4	-0.085441273882
	5	0.035226291885
N=8 L=4	0	0.230377813308
	1	0.714846570552
	2	0.630880767939
	3	-0.027983769416
	4	-0.18703481171
	5	0.030841381835
	6	0.032883011666
	7	-0.010597401785
N=10 L=5	0	0.160102397974193
	1	0.603829269797190
	2	0.724308528437772
	3	0.138428145901320
	4	-0.242294887066381
	5	-0.032244869584637
	6	0.077571493840045
	7	-0.006241490212798
	8	-0.012580751999082

	n	h[n]
N=12	0	0.111540743350000
	1	0.494623890398000
	2	0.751133908021000
	3	0.315250351709000
	4	-0.226264693965000
	5	-0.129766867567000
	6	0.097501605587000
	7	0.027522865530000
	8	-0.031582039318000
	9	0.000553842201000
	10	0.004777257511000
	11	-0.001077301085000
N=16	0	0.054415842243000
	1	0.312871590914000
	2	0.675630736297000
	3	0.585354683654000
	4	-0.015829105256000
	5	-0.284015542961000
	6	0.000472484573000
	7	0.128747426620000
	8	-0.017369301001000
	9	-0.044088253930000
	10	0.013981027917000
	11	0.008746094047000
	12	-0.004870352993000
	13	-0.000391740373000
	14	0.000675449406000
	15	-0.000117476784000

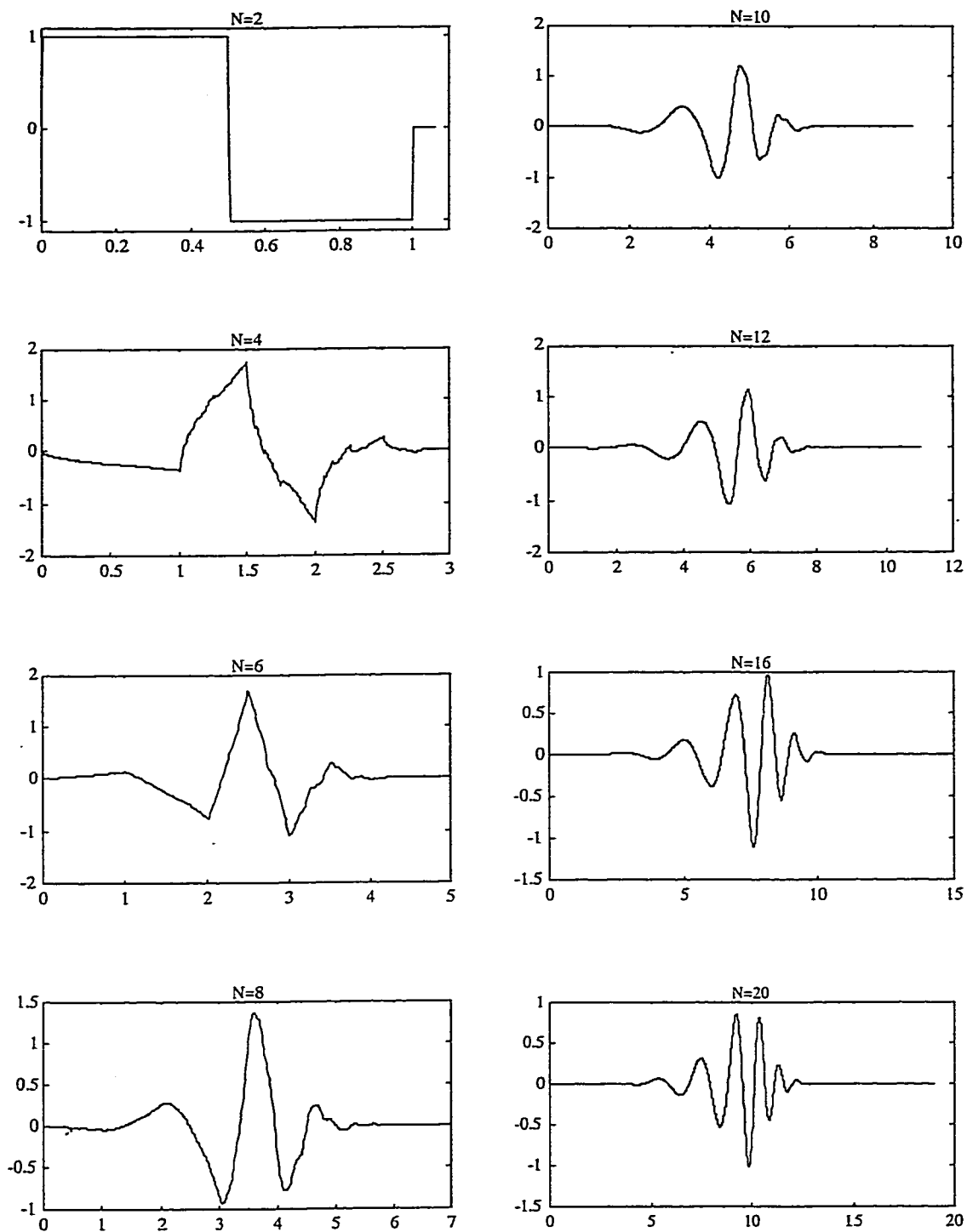


Figure A.1: Daubechies mother wavelets with different supports. N is the number of corresponding filter taps.

Appendix B

B.1 Image Databases

In this thesis, various indexing techniques were evaluated based on three image databases. The details of these databases are presented below.

B.1.1 General Database (IDB1)

This database contains 44 images (size 248×256) of wide varieties, including faces, natural scenes, animals, and birds. Each image has five derivatives corresponding to i) normal, ii) translated rightward, iii) translated leftward, iv) rotated clockwise, and v) rotated anti-clockwise. Thus, we have a total of 220 images. The images are shown in Fig. B.1.

B.1.2 Texture Database (IDB2)

This database contains 50 Brodatz textures. Each texture has four similar images, i.e., the total number of images is 200. The images are 8 bit gray level (0-255), and of size 256×256 . The textures are shown in Fig. B.2.

B.1.3 Illumination Database-1 (IDB3)

This database contains the same 44 images as in IDB1. However, all the images are 24-bit color images $\{Y, Cb, Cr\}$. Each image has three derivatives corresponding to i) normal, ii) low, and iii) high illumination (see Fig. B.3). Thus, we have a total of 132 images with varying illumination conditions. The different illumination conditions were simulated using

histogram modification, which is linear in the central region and marginally nonlinear at the edges. The LI and HI conditions were simulated using $\alpha = 1.25$ and $\alpha = 0.8$, respectively.

B.1.4 Illumination Database-2 (IDB4)

This database contains the same 44 images as in IDB3. However, each image has three derivatives corresponding to i) normal position and normal illumination, ii) clockwise rotated along with low illumination, and iii) anti-clockwise rotated along with high illumination (see Fig. B.4). Thus, we have a total of 132 images with varying illumination conditions as well as spatial changes. The different illumination conditions were simulated using histogram modification, which is linear in the central region and marginally nonlinear at the edges. As in IDB3, the LI and HI conditions were simulated using $\alpha = 1.25$ and $\alpha = 0.8$, respectively.

B.2 Video Databases

In this thesis, various video coding techniques are evaluated based on three video sequences - i) Pingpong, ii) Salesman, and iii) Miss America. Pingpong (with size 360×240) has high spatial details and fast motion. On the other hand, Salesman and Miss America (CIF format, 360×288), are typical video conferencing sequences with slow motion and low spatial details. One frame from each video sequence is shown in Fig. B.5. All three video sequences are in 8-bit gray level format.

The video segmentation techniques are evaluated using Beverly Hills sequence which is a digitized version of the movie "Beverly Hills". First 480 frames of the sequence are shown in Fig. B.6. The video sequence is in 8-bit gray level format with a frame size of 384×288 .

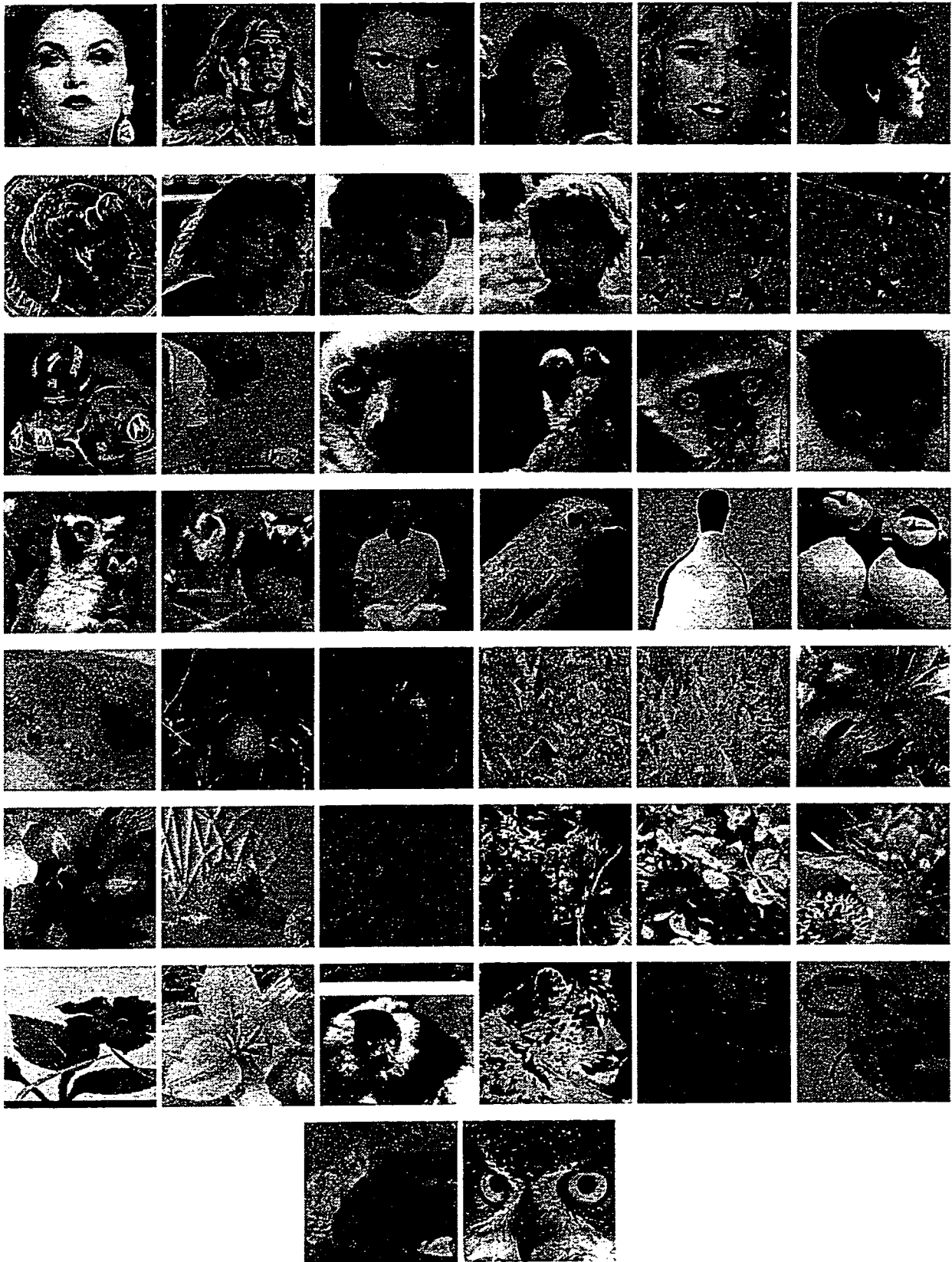


Figure B.1: General Database (IDB1)

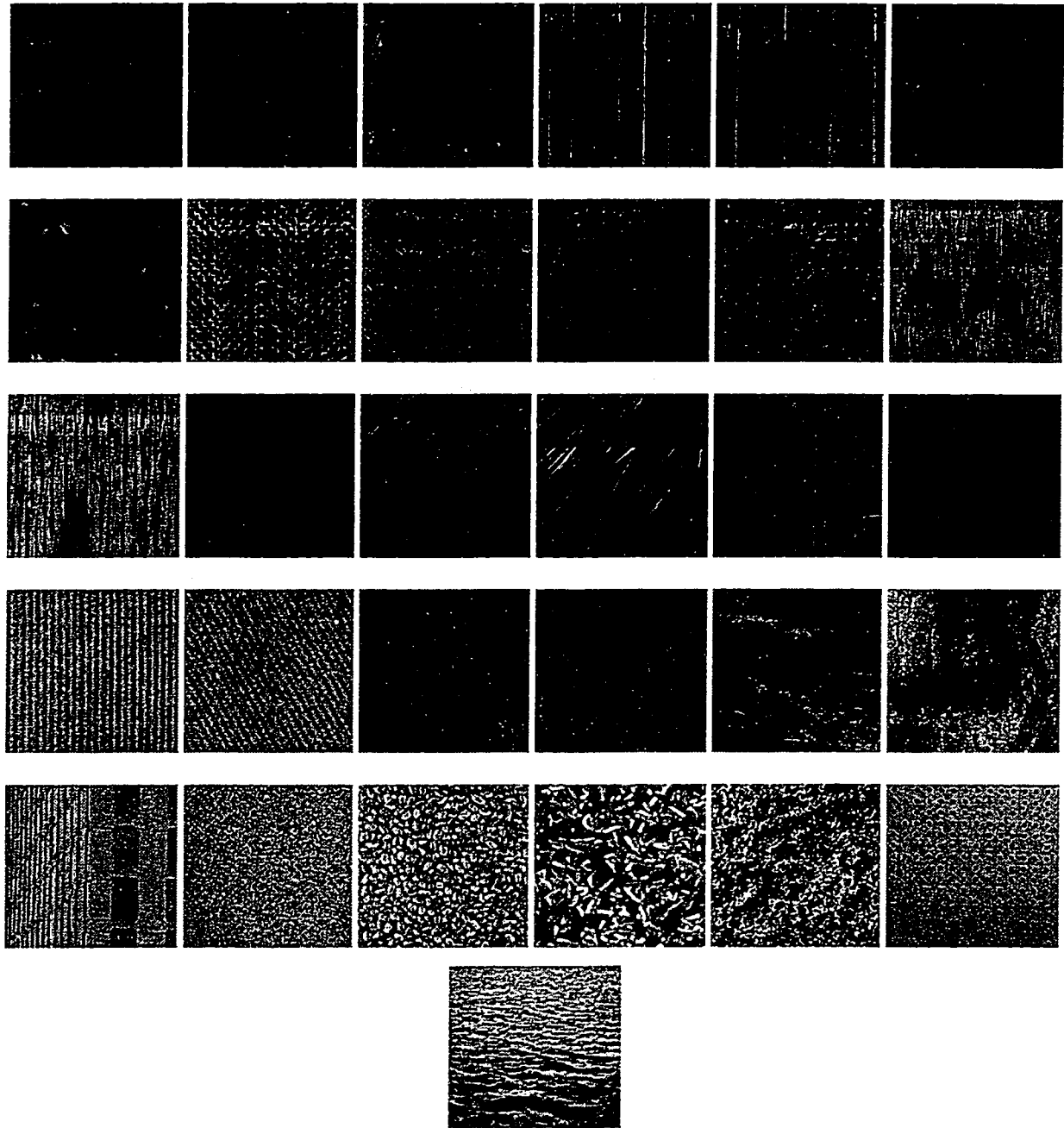


Figure B.2. Texture database (IDB2)

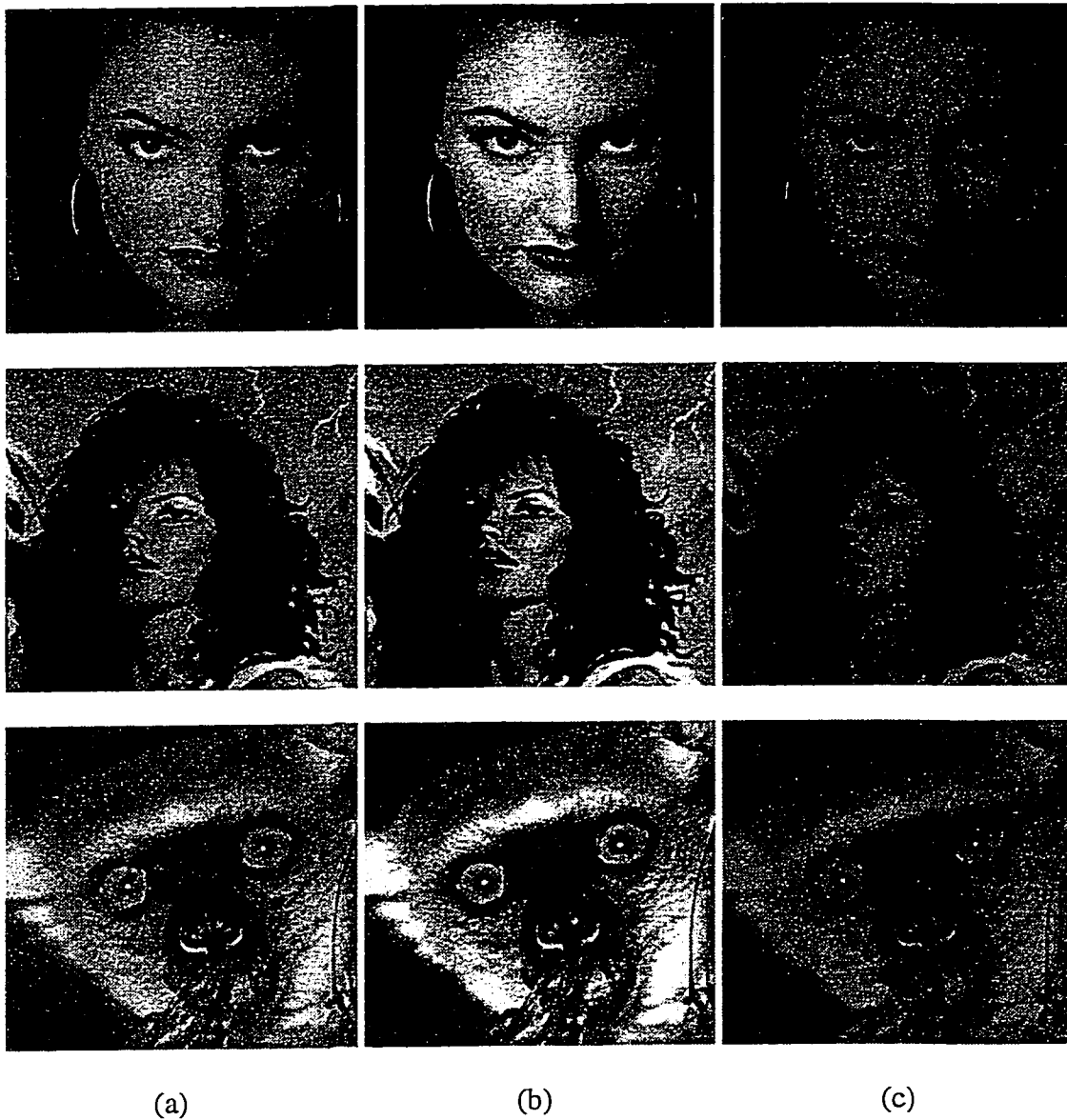


Figure B.3. A few sample images from illumination database-1 (IDB3). a) images with standard illumination, b) images with high illumination, c) images with low illumination.

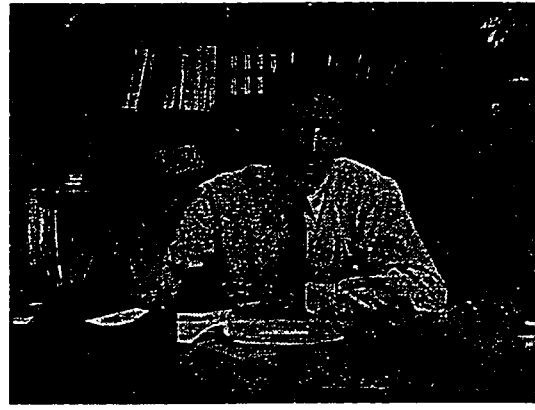


(a)

(b)

(c)

Figure B.4. A few sample images from illumination database-2 (IDB4). a) images with standard position and illumination, b) images with clockwise rotation and low illumination, c) images with anti-clockwise rotation and high illumination.



(a) (b)



(c)

Figure B.5. One frame each from three video sequences employed in this thesis. a) Pingpong, b) Salesman, and Miss America.

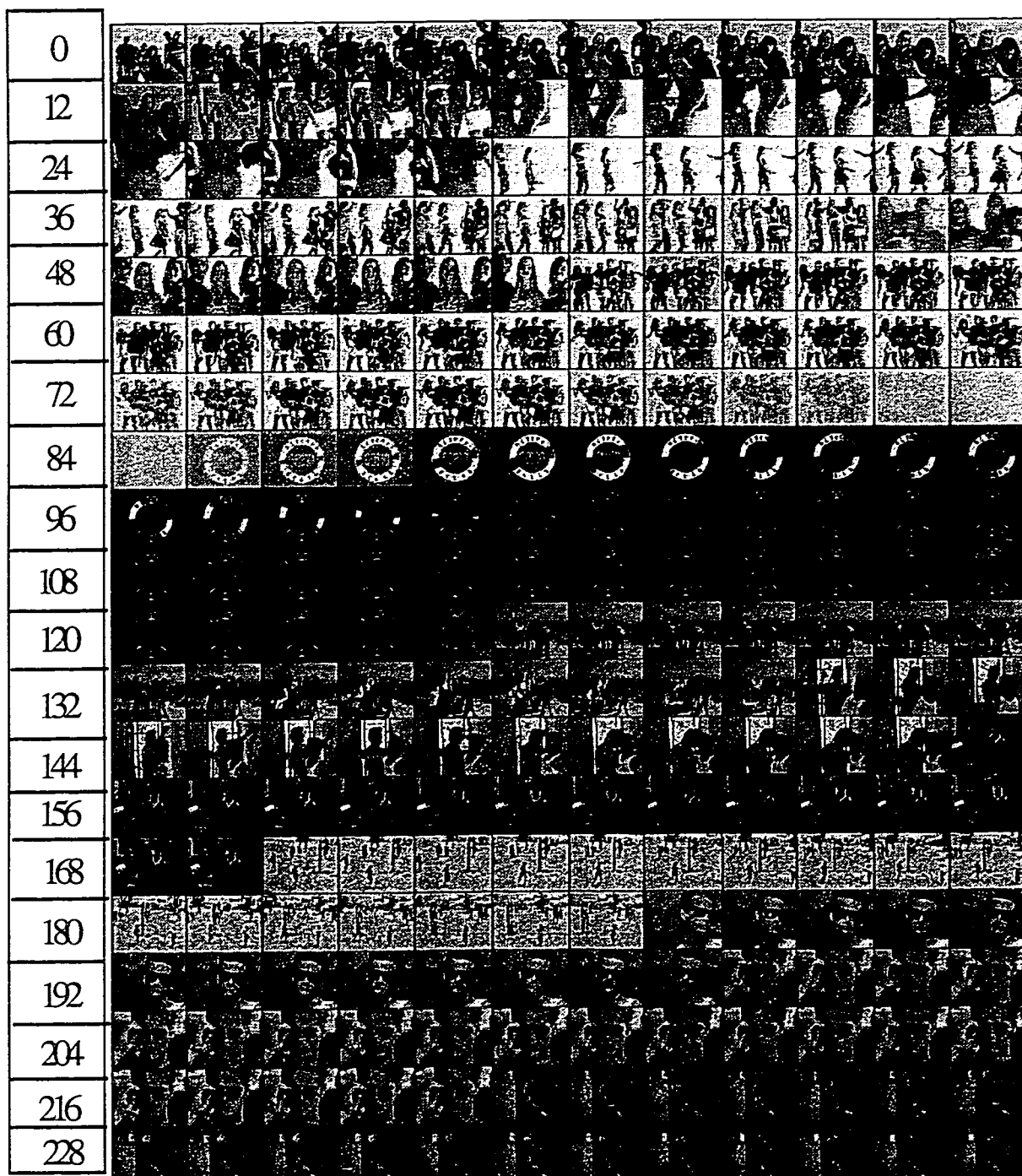


Figure B.6. First 480 frames of Beverly Sequence

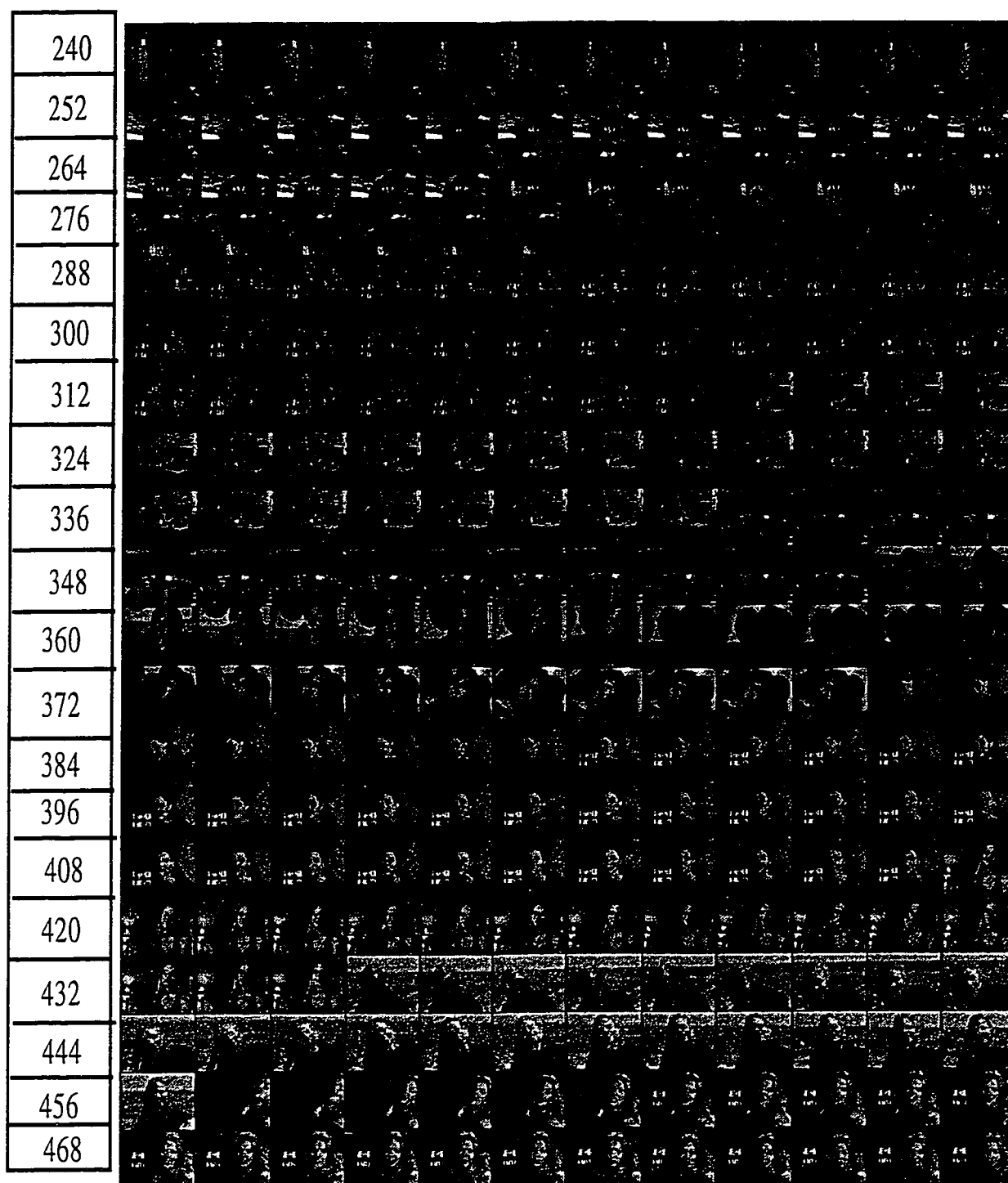
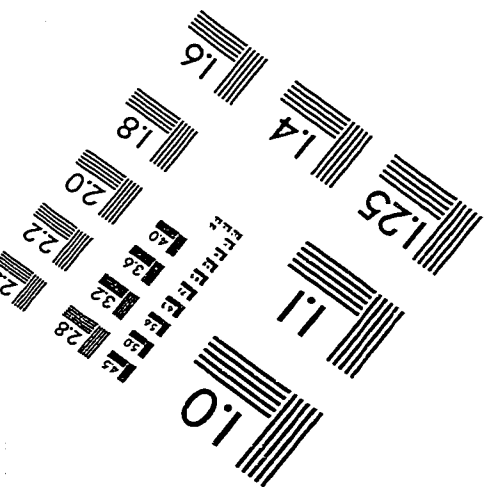
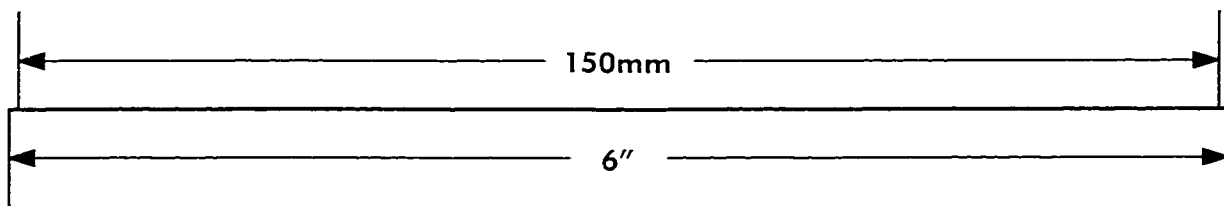
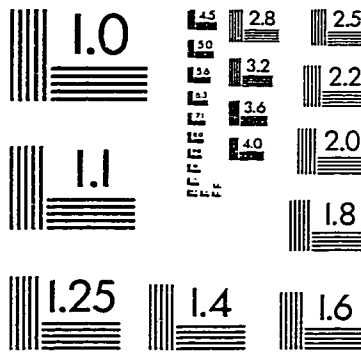
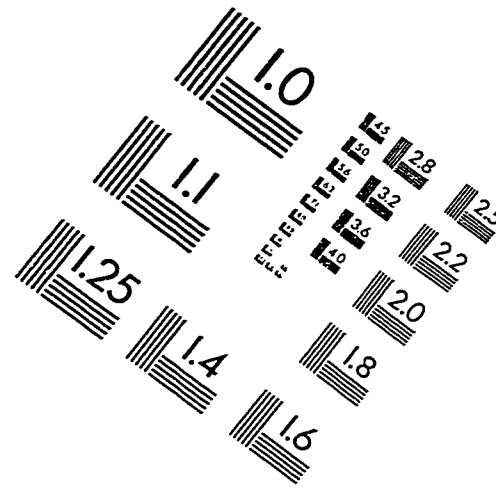
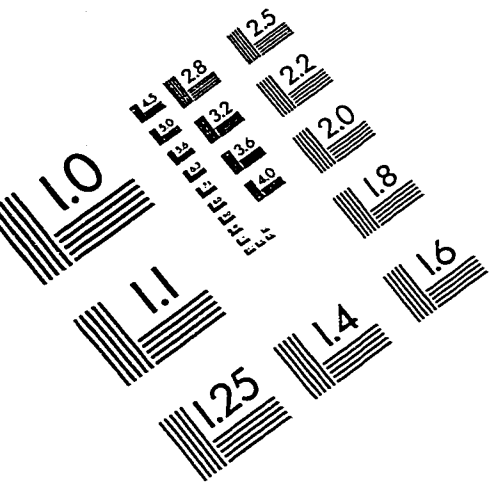


Figure B.6 continued

IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved

