

RESTORATION OF BLURRED IMAGES USING GEOMETRIC AND
TCHEBICHEF MOMENTS

AHLAD KUMAR

FACULTY OF ENGINEERING
UNIVERSITY OF MALAYA
KUALA LUMPUR

2016

**RESTORATION OF BLURRED IMAGES USING
GEOMETRIC AND TCHEBICHEF MOMENTS**

AHLAD KUMAR

**THESIS SUBMITTED IN FULFILMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY**

**FACULTY OF ENGINEERING
UNIVERSITY OF MALAYA
KUALA LUMPUR**

2016

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Ahlad Kumar

Registration/Matrix No.: KHA120025

Name of Degree: Doctor of Philosophy (Ph.D)

Title of Thesis("RESTORATION OF BLURRED IMAGES USING GEOMETRIC AND TCHEBICHEF MOMENTS"):

Field of Study: Electronic

I do solemnly and sincerely declare that:

- (1) I am the sole author/writer of this Work;
- (2) This work is original;
- (3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
- (4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
- (5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
- (6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature

Date

Subscribed and solemnly declared before,

Witness's Signature

Date

ABSTRACT

Blur affects the edges of an image that leads to the degradation of the image quality. Several methods have been developed in both spatial and frequency domains to deblur Gaussian and motion blurred images by using iterative methods to estimate the blur parameters. In this study geometric moments (GM) and Tchebichef moments (TM), from the family of non-orthogonal and orthogonal moments respectively, are utilized for deblurring of images. Here, three methods are proposed for deblurring of images. In the first method, the framework of variational method is formulated in moment domain to implement deblurring of the Gaussian and motion blurred images using Euler-Lagrange identity and alternate minimization (AM) algorithm. It uses an iterative procedure in the form of partial differential equations (PDE) to restore the deblurred GMs. This is addressed for both non-blind and blind methods which use an iterative procedure to restore the deblurred GMs. Then, a reconstruction method using Stirling numbers is used to restore the deblurred image from the deblurred GMs. Three experiments are carried out to demonstrate the effectiveness of the proposed method on the quality of the restored images by considering the effects of the regularization parameter and blur size. In the second method, Gaussian blur estimation problem is modelled as regression problem and is solved using Weighted Geometric moments (WGM) and extreme learning machine (ELM). In particular, WGMs are formulated as linear combination of fundamental basis GMs which are used as feature vectors that can effectively capture the behavior of edges present in an image subjected to Gaussian blur. These feature vectors along with ELM are used in estimating the blur parameters. Once the blur parameters are estimated, the restoration of the degraded image is performed in moment domain using the cascaded digital filters operating as subtractors to perform the task of image reconstruction. Here, two experiments are performed on six publicly available standard databases of images in

order to validate the performance of the proposed method. In the first experiment, the cross database analysis of the proposed method for blur estimation is carried out and the results show that the blur parameters can be estimated. In the second experiment, the proposed methods are compared with the five existing methods and the quality of the restored images is evaluated using BRISQUE and SSIM. The results show the proposed method performed well in most cases. In the third method, Tchebichef moments (TM) of low order are selected as features used as inputs to ELM to estimate the Gaussian blur parameters. Once the blur parameters are estimated, image restoration of the proposed method is carried out using split Bregman algorithm. The performance analysis using the proposed TM method is compared with the same five existing methods. It has been observed that TMs based image restoration perform well compared to the five existing methods when evaluated using image quality metrics.

ABSTRAK

Kabur memberi kesan kepada bahagian tepi imej yang akan mengakibatkan penurunan kualiti imej. Beberapa kaedah telah dibangunkan dalam kedua-dua domain spasial dan frekuensi untuk menyah-kabur Gaussian dan kabur gerakan di dalam imej, dengan menggunakan kaedah lelaran untuk menganggarkan parameter kabur. Dalam kajian ini, momen geometri (GM) dan momen Tchebichef (TM) dari keluarga momen yang tidak ortogonal dan ortogonal, digunakan untuk menyahkaburkan kabur yang terdapat pada imej. Dalam kaedah pertama, rangka kerja kaedah variasi dirumuskan dalam domain masa untuk melaksanakan proses nyahkabur imej menggunakan kedua-dua identiti Euler-Lagrange dan algoritma Alternate Minimization (AM). Ia menggunakan prosedur lelaran dalam bentuk persamaan pembezaan separa (PDE) untuk memulihkan GM. Kaedah ini menggunakan prosedur lelaran untuk memulihkan GM yang telah dinyahkabur. Kemudian, satu kaedah pembinaan semula menggunakan nombor Stirling digunakan untuk memulihkan imej yang telah dinyahkabur dari GM yang dinyahkabur. Tiga eksperimen dijalankan untuk menunjukkan keberkesanan kaedah yang dicadangkan terhadap kualiti gambar yang telah dinyahkabur dengan mengambil kira kesan daripada penggunaan parameter variasi dan saiz kabur. Dalam kaedah kedua, masalah penganggaran PSF dimodelkan sebagai masalah regresi dan diselesaikan dengan menggunakan momen berwajaran geometri (WGM) dan Extreme Learning Machine (ELM). Khususnya, WGMs dirumuskan sebagai gabungan linear asas utama GM yang digunakan sebagai vektor ciri yang berkesan bagi menangkap kelakuan tepi di dalam imej yang tertakluk kepada kaburan Gaussian. Vektor ciri ini bersama-sama dengan ELM digunakan dalam penganggaran parameter kabur. Setelah kabur dianggar, pemulihan imej yang rosak dilakukan dalam domain masa. Ini diikuti dengan penggunaan litar penolak yang menggunakan penapis digital berlapis untuk melaksanakan tugas pembinaan semula imej dari domain momen ke domain ruang.

Di sini, untuk mengesahkan keberkesanan kedua-dua kaedah cadangan, kedua-dua eksperimen ini dijalankan ke atas enam pengkalan data umum untuk imej. Dalam eksperimen pertama, analisa silang antara pengkalan data untuk kaedah dicadangkan untuk menganggar PSF dijalankan dan keputusan eksperimen menunjukkan nilai PSF yang dianggarkan. Dalam eksperimen kedua, kaedah dicadangkan dibandingkan dengan tiga kaedah sedia ada dan kualiti imej yang terhasil dinilai menggunakan BRISQUE and SSIM. Keputusan eksperimen menunjukkan kaedah cadangan berfungsi dengan baik di setiap kes. Akhir sekali, detik-detik Tchebichef rendah dipilih sebagai ciri yang digunakan sebagai input kepada ELM untuk menganggar parameter kabur. Setelah PSF dianggar, kaedah pemulihan imej yang dicadangkan dijalankan menggunakan algoritma split Bregman. Analisis prestasi menggunakan kaedah cadangan bagi TM dibandingkan dengan tiga kaedah sedia ada. Berdasarkan pemerhatian dan nilai kualiti metric bagi imej, proses penghasilan semula imej menggunakan TM mempunyai prestasi yang lebih baik berbanding tiga kaedah sedia ada.

ACKNOWLEDGEMENTS

This thesis has benefited greatly from the support of many people, some of whom I would sincerely like to thank here: My thesis supervisor, Professor Dr. Raveendran Parames-ran for offering me such an interesting topic of investigation. His continuous guiding, constant support and invaluable suggestions and advices have given me the confidence to complete this research work.

Present and past members of the Center for Signal and Image Processing (CISIP) of Uni-versity of Malaya, for their directly and indirectly contribution to the constructive ideas and the time together in the lab. All the authors of the image databases and quality met-rics, to share their source codes and results of the tests. Without the unselfishness in sharing the knowledge and data, the comparison studies would have not been possible. All the people behind the offices of the Department of Electrical Engineering and Faculty of Engineering for giving me the opportunity to pursue my education. Last but not least, I would like to thank my parents and wife for their thorough support and encouragement throughout the difficult times.

TABLE OF CONTENTS

Abstract	iii
Abstrak	v
Acknowledgements	vii
Table of Contents	viii
List of Figures	xi
List of Tables	xiv
List of Symbols and Abbreviations	xvi
List of Appendices	xvii
 CHAPTER 1: INTRODUCTION	 1
1.1 Overview	1
1.2 Motivation	7
1.3 Objectives	7
1.4 Organization	9
1.5 Contributions	12
 CHAPTER 2: REVIEW OF GEOMETRIC AND TCHEBICHEF MOMENTS FOR IMAGE DEBLURRING	 14
2.1 Introduction	14
2.2 Geometric Moments	18
2.2.1 Relationship between blurred and original image in Geometric moments domain	19
2.2.2 Computation of Geometric moments using Feed-forward digital filter ..	22
2.2.3 Computation of Geometric moments using Feedback digital filter	25
2.2.4 Restoration of original image using Subtractor based digital filter	31
2.2.5 Restoration of original image using Stirling Numbers	40
2.3 Tchebichef Moments	41

2.4	Conclusion	44
CHAPTER 3: RESTORATION OF BLURRED IMAGES USING GEOMETRIC MOMENTS.....		45
3.1	Introduction.....	45
3.2	Non-Blind Deblurring in Moment Domain	51
3.2.1	Experimental Results.....	53
3.3	Restoration of motion blurred images degraded by motion blur	56
3.3.1	Comparative analysis on images degraded by motion blur.....	59
3.4	Blind deblurring in Moment domain	60
3.4.1	Selection Procedure for (λ).....	68
3.4.2	Experimental Results.....	69
3.5	Conclusion	78
CHAPTER 4: RESTORATION OF THE BLURRED IMAGES IN MOMENT DOMAIN VIA CASCADED DIGITAL FILTERS....		81
4.1	Introduction.....	81
4.2	Formulation of feature vectors using Weighted geometric moments (WGM)	89
4.2.1	Weighted Geometric Moments (WGM) as edge descriptors	91
4.2.2	Plain and Edge block classification	93
4.3	PSF parameters estimation via Extreme learning machine (ELM)	94
4.4	Experimental study for the estimated PSF parameters	99
4.4.1	Data description.....	99
4.4.2	Significance of using Edge blocks for PSF estimation.....	100
4.4.3	Cross-database validation	102
4.5	Proposed Image restoration in moment domain	104
4.6	Comparative analysis with the existing state of the art methods	109
4.6.1	Experiment 1	110
4.6.2	Experiment 2	111

4.6.3	Comparative Analysis using images from the published database images.....	112
4.7	Conclusion	115
CHAPTER 5: ESTIMATION OF THE PSF PARAMETERS USING TCHEBICHEF MOMENTS FOR IMAGE RESTORATION		117
5.1	Introduction.....	117
5.2	Formulation of feature vectors using Tchebichef moments (TM).....	122
5.2.1	Tchebichef moments as edge block descriptor.....	125
5.2.2	Plain and Edge block classification	127
5.3	PSF parameters estimation via Extreme learning machine (ELM)	128
5.4	Experimental study for the estimated PSF parameters	132
5.4.1	Data description.....	133
5.4.2	Significance of using edge blocks in PSF estimation.....	134
5.4.3	Training and testing using same databases.....	136
5.4.4	Cross-database validation.....	137
5.5	Image restoration using split Bregman algorithm.....	139
5.6	Comparative analysis with the existing state of the art methods	141
5.6.1	Experiment 1	142
5.6.2	Experiment 2	143
5.6.3	Comparative Analysis using images from the published database images.....	145
5.7	Conclusion	148
CHAPTER 6: CONCLUSION.....		149
6.1	Future Research Directions.....	151
	References.....	152
	List of Publications and Papers Presented	166
	Appendices.....	167

LIST OF FIGURES

Figure 1.1: Flowchart of thesis contents	9
Figure 2.1: Flowchart of the chapter content	22
Figure 2.2: 2D digital filter structure for calculating geometric moment up to 3 rd order.....	22
Figure 2.3: Feed-forward filter	23
Figure 2.4: One dimensional feed-forward filter structure	24
Figure 2.5: Feed-forward based structure (a) Row Filtering (b) Column Filtering Operation	26
Figure 2.6: Feedback filter	26
Figure 2.7: One dimensional feedback filter structure.....	27
Figure 2.8: Feedback based structure (a) Row Filtering (b) Column Filtering Operation	28
Figure 2.9: Proposed 2D digital filter structure for geometric moment calculation ...	30
Figure 2.10: Subtractor block diagram for N input sequence	32
Figure 2.11: Basic building block of horizontal subtractor digital filter (S_{fh}).....	33
Figure 2.12: Basic building block of vertical subtractor digital filter (S_{fv})	33
Figure 2.13: Subtractor matrix	35
Figure 2.14: 1D cascaded subtractor digital filter for 4-input sequence	35
Figure 2.15: Data flow in subtractor matrix shown in Fig 2.13	36
Figure 2.16: Block diagram of the 2D subtractor circuit for $N \times N$ image.....	37
Figure 2.17: Grayscale images used in the reconstruction process (a) 'blonde' image (b) 'pepper' image	39
Figure 2.18: Basis function of Tchebichef moments	43
Figure 3.1: (a) Estimated $m_1^{(\hat{f})}$ and (b) Estimated $m_2^{(\hat{f})}$	54
Figure 3.2: Error vs Iterations	54
Figure 3.3: Flowchart of the alternate minimization algorithm	63

Figure 3.4: Moments for different p (a) $m_p^{(\hat{h})}$ and (b) $m_p^{(\hat{f})}$.	66
Figure 3.5: L-curve for Optimal λ selection	69
Figure 3.6: Test Images	69
Figure 3.7: BRISQUE with regularization parameter λ	72
Figure 4.1: Flowchart of the proposed method	88
Figure 4.2: (a) Vertical (b) Horizontal and (c) Diagonal edge blocks of size 8×8 ; (d)-(f) \mathbf{F} variation with σ	92
Figure 4.3: (a-c) Images with varying edge content (d)-(f) Classification map showing edge patches	94
Figure 4.4: Proposed algorithm for PSF estimation	96
Figure 4.5: Effects of the initial parameter selection on training and testing performance for same database	97
Figure 4.6: Effects of the initial parameter selection on training and testing performance for cross database	98
Figure 4.7: (a) Training (b) Testing accuracy variation with respect to hidden neurons and initial parameters for same database	98
Figure 4.8: (a) Training (b) Testing accuracy variation with respect to hidden neurons and initial parameters for cross database	99
Figure 4.9: Moment transform (forward-inverse) via cascaded digital filters	107
Figure 4.10: 2D cascaded digital filter architecture	107
Figure 4.11: Comparison of the deblurring results on <i>woman</i> image blurred with Gaussian PSF of $(\sigma, w)=(1.5, 11)$	111
Figure 4.12: Comparison of the deblurring results on <i>house</i> image blurred with Gaussian PSF of $(\sigma, w)=(2.0, 15)$	112
Figure 4.13: Comparative analysis of SSIM score for the LIVE database images for different values of PSF parameters (σ, w) with existing methods namely: Kotera et al. (Kotera, Šroubek, & Milanfar, 2013), Goldstein et al. (A. Goldstein & Fattal, 2012) and (G. Liu, Chang, & Ma, 2014)	113
Figure 5.1: Flowchart of the proposed method	122
Figure 5.2: Basis function ϕ_{pq} for Tchebichef moment	125
Figure 5.3: (a) Vertical edge block of size 8×8 (b)-(d) T_{pq} variation with σ	126
Figure 5.4: (a) Horizontal edge block of size 8×8 (b)-(d) T_{pq} variation with σ	127

Figure 5.5: (a) Diagonal edge block of size 8×8 (b)-(d) T_{pq} variation with σ	127
Figure 5.6: Flowchart of the proposed algorithm	129
Figure 5.7: Effects of the initial parameter selection on training and testing performance for same database.	131
Figure 5.8: Effects of the initial parameter selection on training and testing performance for cross database.	131
Figure 5.9: (a) Training (b) Testing accuracy variation with respect to hidden neurons and initial parameters for same database	132
Figure 5.10: (a) Training (b) Testing accuracy variation with respect to hidden neurons and initial parameters for cross database	132
Figure 5.11: Test Images taken from LIVE database.....	134
Figure 5.12: Proposed restoration method	140
Figure 5.13: Comparison of the deblurring results on <i>parrots</i> image blurred with Gaussian PSF of $\sigma=2.5$ and $w = 13$	144
Figure 5.14: Zooming the details of <i>parrots</i> image for observing the local edge restoration	144
Figure 5.15: Comparison of the deblurring results on <i>building</i> image blurred with Gaussian PSF of $\sigma=1.5$ and $w = 11$	144
Figure 5.16: Zooming the details of <i>building2</i> image for observing the local edge restoration	145
Figure 5.17: Comparative analysis of SSIM score for the LIVE database images for different values of PSF parameters (σ, w) with existing methods namely: Kotera et al. (Kotera et al., 2013), Goldstein et al. (A. Goldstein & Fattal, 2012) and Liu et al.(G. Liu et al., 2014).....	146
Figure A.1: 29 Reference Images of LIVE Database	167
Figure A.2: 30 Reference Images of CSIQ Database	168
Figure A.3: 25 Reference Images of TID2008 Database.....	169
Figure A.4: 12 Reference Images of Caltech Database	170
Figure A.5: 12 Reference Images of Berkeley Database	170
Figure A.6: Sample of blurred images obtained from Caltech and Berkeley Databases	171
Figure A.7: Sample of blurred images obtained from LIVE, CSIQ and TID2008 Databases	171

LIST OF TABLES

Table 2.1: CPU time (sec) for grayscale images using Subtractor circuit	39
Table 3.1: Image deblurring using spatial and moment domain for a binary image of 32×32 with different Gaussian kernel, σ and mask size, w with their corresponding BRISQUE.....	56
Table 3.2: Image deblurring using spatial and moment domain approaches for test images of various sizes with different Gaussian kernel, σ and mask size, w and their corresponding BRISQUE.	57
Table 3.3: Comparative analysis of the different methods evaluated in terms of BRISQUE score	58
Table 3.4: PSNR, SSIM and BRISQUE results of the methods on four different images with motion blurred PSF	61
Table 3.5: Optimal values of λ	72
Table 3.6: BRISQUE scores variation with PSF size (w).....	73
Table 3.7: Image deblurring in Spatial and Moment domain.....	74
Table 3.8: Quality evaluation for Lena image	75
Table 3.9: Quality evaluation for Cameraman image	75
Table 3.10: Quality evaluation for Barbara image	75
Table 3.11: Full-Reference quality evaluation for Lena image.....	77
Table 3.12: Full-Reference quality evaluation for Cameraman image	77
Table 3.13: Full-Reference quality evaluation for Barbara image	77
Table 3.14: PSNR, SSIM and BRISQUE results of the methods on four different images with a Gaussian PSF of size 7×7 and $\sigma=1.2$	79
Table 3.15: PSNR, SSIM and BRISQUE results of the methods on four different images with an average PSF of size 9×9	80
Table 4.1: Classification of LIVE database images based on image content.....	94
Table 4.2: Details of testing and training samples for various databases.....	100
Table 4.3: Role of edge block in estimating the PSF sigma (σ).....	101
Table 4.4: Role of edge block in estimating the PSF size (w)	102

Table 4.5: Correlation coefficient (CC) values for cross database performance of estimating PSF's sigma (σ).....	103
Table 4.6: Correlation coefficient (CC) values for cross database performance of estimating PSF's size (w).	103
Table 4.7: Comparative analysis with the existing methods in terms of SSIM scores	110
Table 4.8: Quantitative comparison with the existing methods by using the images present in them.....	114
Table 4.9: Comparative analysis with the existing methods in terms of SSIM scores	115
Table 4.10: Quantitative comparison with the existing methods by using the images present in them	116
Table 5.1: Details of testing and training samples for various databases.....	133
Table 5.2: Classification of LIVE database images based on image content.....	134
Table 5.3: Role of edge block in estimating the PSF sigma (σ).....	135
Table 5.4: Role of edge block in estimating the PSF size (w)	136
Table 5.5: Estimation of PSF parameter σ for various databases	137
Table 5.6: Estimation of PSF parameter (w) for various databases.....	137
Table 5.7: Correlation coefficient (CC) values for cross database performance of estimating PSF's sigma (σ).....	138
Table 5.8: Correlation coefficient (CC) values for cross database performance of estimating PSF's size (w).	139
Table 5.9: Comparative analysis with the existing methods in terms of SSIM scores	142
Table 5.10: Quantitative comparison with the existing methods by using the images present in them	147
Table 5.11: Quantitative comparison with the existing methods by using the images present in them	148

LIST OF SYMBOLS AND ABBREVIATIONS

AM	: Alternate Minimization.
BID	: Blind Image Deblurring.
BOS-SB	: Bregmanized Operator Splitting- Split Bregman.
BRISQUE	: Blind/Reference-less Image Spatial Quality Evaluator.
BSS	: Blind Signal Separation.
CC	: Correlation Coefficient.
CT	: Computed Tomography.
DFT	: Discrete Fourier Transform.
ELM	: Extreme Learning Machine.
EM	: Expectation Maximization.
FR	: Full Reference.
FSIM	: Feature Similarity.
GM	: Geometric Moment.
HDTV	: High Definition Television.
HVS	: Human Visual System.
LSI	: Linear Shift Invariant.
MSE	: Mean Square Error.
NR	: No-Reference.
PDE	: Partial Differential Equation.
PET	: Positron Emission Tomography.
PPLN	: Projection Pursuit Learning Networks.
PSF	: Point Spread Function.
PSNR	: Peak Signal to Noise Ratio.
ROF	: Rudin Osher Fatemi.
RR	: Reduced-Reference.
RTFS	: Regression Tree Fields.
SSIM	: Structural Similarity Index.
TM	: Tchebichef Moment.
TV	: Total Variation.
VIF	: Visual Information Fidelity.
VLSI	: Very Large Scale Integration.
WGM	: Weighted Geometric Moment.

LIST OF APPENDICES

Appendix A: Database	167
Appendix B: Source Codes.....	172

University of Malaya

CHAPTER 1 : INTRODUCTION

1.1 Overview

Images are produced in order to record or display useful information. Due to imperfections in the electronic or photographic medium, however, the recorded image often represents a degraded version of the original scene. The degradations may have many causes, but two types of degradations are often dominant: blurring and noise. Blurring is a form of bandwidth reduction of the image due to the imperfect image formation process. It can be caused by relative motion between the camera and the original scene, or by an optical system which is out of focus. When aerial photographs are produced for remote sensing purposes, blurs are introduced by atmospheric turbulence, aberrations in the optical system, and relative motion between the camera and the ground. Such blurring is not confined to optical images, for example, electron micrographs are corrupted by spherical aberrations of the electron lenses and Computed Tomography (CT) scans suffer from X-ray scatter. In addition to these blurring effects, the recorded image is also corrupted by noises. These may be introduced by the transmission medium (e.g. a noisy channel), the recording medium (e.g. film grain noise), measurement errors due to the limited accuracy of the recording system, and quantization of the data for digital storage.

The term deblurring is commonly used to refer to restoration of images degraded by blur. Although the degradation process is in general nonlinear and space varying, a large number of problems could be addressed with a Linear Shift Invariants (LSIs) model. Because the output of an LSI system is the convolution of the true image with the impulse response of the system, the point spread function (PSF), image restoration in LSI systems is called image deconvolution. Typically, the phenomena of degradation is modeled as

$$g(x,y) = h(x,y) * f(x,y) + \eta(x,y) \quad (1.1)$$

where $g(x,y)$ is the observed degraded image, $f(x,y)$ is the original image, $h(x,y)$ is the PSF, $\eta(x,y)$ is the noise present in the system and $*$ is the convolution operator.

The degraded images, generally, masks and blurs important but subtle features in the images. Other sources of blur degradation are geometric and motion un-sharpness. The geometric un-sharpness owes its origin from the lack of beam collimation, statistical fluctuation due to low intensities or high background. Similarly, motion un-sharpness is due to object motion during exposure, and limitations in acquisition/processing systems. Hence, it becomes important to deblur the image which is subjected to aforementioned degradations. To overcome these problems, image restoration is a solution to reduce the blurring effects on the image. It tries to perform an inverse transformation of the observed blurred image to estimate the original one. Modeling the degradation is an essential part in performing the inverse transformation.

Image restoration is widely used in almost all technical areas involving images; astronomy, remote sensing, microscopy, medical imaging, photography, surveillance, and High Definition Televisions (HDTVs) systems are just a few. For example, license plates may appear illegible due to motion blur; photographs captured under low-light conditions may suffer from noise; out-of-focus photographs may look blurry; standard TV signals may not be sufficiently sharp for high definition TV sets; archived movies may be corrupted by artifacts and noise; atmospheric distortions may degrade the quality of images in remote sensing. In these examples and in many other scenarios, the importance of image restoration ranges from beneficial to essential.

Typically the research involved in the restoration of a degraded image can be classified as non-blind (G. Chantas, Galatsanos, Likas, & Saunders, 2008; Schuler, Burger, Harmeling, & Scholkopf, 2013; Schmidt, Rother, Nowozin, Jancsary, & Roth, 2013) and blind methods (Shi, Hong, Song, & Hua, 2015; Kundur & Hatzinakos, 1996; M. S. Almeida, Figueiredo, et al., 2013). In the case of non-blind method, the PSF $h(x,y)$ is assumed to

be known and a sharp image can be induced from both the blurry image and the PSF. Typical methods include the Richardson-Lucy method (Richardson, 1972) and Wiener filter (Wiener, 1949). Ill-posedness is the most severe problem in image deblurring. In the case of non-blind deblurring, the observed blurred image $g(x,y)$ does not uniquely determine the original image due to the ill-conditioned nature of the blur operator. This means that if there is a slight mismatch between the assumed PSF and the true PSF, or if the observed blurred image is corrupted by noise, the recovered image may be much worse than the underlying original image $f(x,y)$. Driven by these problems and the fact that in most situations of practical interest the PSF, is not known with good accuracy, blind methods for restoring the degraded image are introduced in the literature. The task of simultaneously estimating the PSF and deblurring an unknown image using partial or no information about the imaging system is known as blind image restoration. A good review article on blind deconvolution can be found in (Ruiz, Zhou, Mateos, Molina, & Katsaggelos, 2015; Banham & Katsaggelos, 1997). This problem is highly ill-posed since there are an infinite set of image-blur pairs that can synthesize the observed blurry image. In recent years, many novel approaches have been introduced to handle both the non-blind and blind deblurring problem, driven by a various motivations.

Researchers have been working to develop new models, or improving the efficiency of optimization methods, to deal with ill-posedness nature of deblurring. Most of the techniques related to developing of models can be grouped into the following categories: Bayesian inference framework (G. K. Chantas, Galatsanos, & Likas, 2006; Y. Zhang, Duijster, & Scheunders, 2012), variational methods (Vega, Mateos, Molina, & Katsaggelos, 2012; Danielyan, Katkovnik, & Egiazarian, 2012), and sparse representation-based methods (Amizic, Molina, & Katsaggelos, 2012; Dong, Zhang, Shi, & Li, 2013). In the Bayesian framework, the introduction of priors is included in order to impose uncertainty attributes on the unknown sharp image, the unknown PSF, or both. This operation is

intended to reduce the volume of the search space, where the problem's solution lies, to suppress the ill-posedness. Variational methods provide the unique and stable solution through the incorporation of regularization techniques whose role is similar to the prior's used in Bayesian inference. Sparse representation, in recent years, benefits from the fact that natural images are intrinsically sparse in some domains like Wavelets, Fourier etc. (Figueiredo & Nowak, 2003; Portilla, Strela, Wainwright, & Simoncelli, 2003). In (M. S. Almeida & Almeida, 2010) authors explored the sparsity of natural images to propose new regularizer that eventually helps in restoring the blurred images. In particular, their method is based on the facts that the leading edges in the natural images are sparse and the edges of the blurred image are less sparse when compared to the sharp image as they tend to occupy wide area. Based on these observations they proposed a prior that promotes sparsity of the edges which in turn will make the image sharp. Image restoration model based on a new wavelet frame is proposed that treats natural images as piecewise smooth functions (Cai, Dong, & Shen, 2016). The proposed model seeks piecewise smooth solutions to the linear inverse problem stated in (1.1). The proposed model estimates the restored image and its singularity. Further the method well protects the singularities which are important natural image features and simultaneously provides enough regularization in smooth regions. The proposed model is a combination of the advantages of partial differential equations and wavelets frames. (L. Ma & Zeng, 2016) proposed image deblurring based on the sparse representation of the natural images. In order to achieve this, instead of using over-complete dictionary which is not well structured and exhibits some shortcoming like instability due to approximations and high computational time; authors made use of structured sparse model selection. Recently, a new regularizer has been employed in the task of image deblurring known as overlapping group sparsity. (He, Fan, & Zheng, 2015) proposed an image restoration model by introducing a novel edge-continuous overlapping group sparsity regularizer. This regularizer is based

on the observation that the non-zero entries in an image gradient domain often distribute along its edges. Some researchers adopt a slightly different approach to image restoration problems. Here, the estimation of the PSF is performed independently using various techniques involving machine learning, polynomial transforms (Kayargadde, 1995), exploiting the sharpness of the natural images (Dijk, Van Ginkel, Van Asselt, Van Vliet, & Verbeek, 2003), using derivatives of the images (Bouma, Dijk, & van Eekeren, 2012). Once the PSF is estimated by these methods, the restoration is performed using non-blind method which can restore the image effectively as the PSF is already estimated.

The role of moments in image processing constitute an important feature extraction method which generates highly discriminative features, able to capture the particular characteristics of the described pattern, which distinguish it among similar or totally different objects. The ability of fully describing an image by encoding its features makes them suitable for image analysis (Sim, Kim, & Park, 2004), watermarking (G. Papakostas, Tsougenis, & Koulouriotis, 2010) and pattern recognition (G. A. Papakostas, Boutalis, Karras, & Mertzios, 2007; G. Papakostas, Boutalis, Karras, & Mertzios, 2009; Hosny, 2007). Work has also been done for the moment extraction through Very Large Scale Integrations (VLSIs) structures (Hatamian, 1986; Wong & Siu, 1999; Kotoulas & Andreadis, 2006, 2005, 2008). Recently, their use in the field of medical imaging is also explored (Maidment, Bakic, & Gavenonis, 2012; Meyer-Baese & Schmid, 2014; Hosny, Papakostas, & Koulouriotis, 2013). Moments of an image provide an efficient local descriptors and have been used extensively in image analysis. The features of an image like shape of boundary segments, can be described quantitatively by using statistical moments such as mean, variance and other higher order moments. These statistical measures provide many important image characteristics; for example the mean, $m = \sum z_k p_{zk}$, provides the measure of average intensity and the variance $\sigma^2 = \sum (z_k - m)^2 p_{zk}$ gives a measure of image contrast, where z_k and p_{zk} denote the intensity and the probability of a pixel. The

third moment measures the skewness of the histogram. A measure of its relativeness has been obtained by the fourth moment. The fifth and higher order moments are not so easily related to the histogram shape, but they do provide further discrimination of the texture content. Simple image properties are derived via raw moments, which include area, sum of gray levels, and centroid information. The main advantage of the moments is their ability to provide invariant measures of shape. The moments calculation for a digital image are important for its successful application in pattern recognition problems, aircraft identification, position detection, scene matching, image reconstruction, restoration, segmentation, color and character recognition (Ghorbel, Derrode, Dhahbi, Mezhoud, et al., 2005; P.-T. Yap & Raveendran, 2002; P. Yap, Raveendran, & Ong, 2001).

Hu(M.-K. Hu, 1962) used these moments for the first time in extracting the image features and employed them for image analysis and object representations. His Uniqueness Theorem states that, if an image $f(x,y)$ is piecewise continuous and has nonzero values only in the finite part of the (x,y) plane then geometric moments of all orders exist. It can then be shown that the moment set is uniquely determined by $f(x,y)$ and vice-versa. Some popular moments, which have been commonly used are like Geometric, Complex, Legendre, Zernike, Tchebichef, and Hann. Geometric moments (Martinez & Thomas, 2002) present low computational cost and are invariant under linear transformations. Complex moments (Teh & Chin, 1988) are invariant descriptors, but have drawback when dealing with noise and image reconstruction. Continuous orthogonal moments Legendre and Zernike (Abu-Mostafa & Psaltis, 1985; Mukundan, Ong, & Lee, 2001b) proposed by Teague (Teague, 1980) are less sensitive to noise, linear transformation invariant and can effectively be used for image reconstruction. However, their computational complexity is a major issue and their real-time implementation has not been reported.

1.2 Motivation

The field of image deblurring is explored in spatial and wavelet domains and no work has been reported in moment domain. This motivated to carry out this research in moment domain. In particular, images degraded due to Gaussian and motion blur are considered. The blur models are formulated in moment domain and the restoration is performed using geometric and Tchebichef moments. Moreover, the task of image reconstruction was not possible due to non-orthogonal nature of basis function of geometric moments. But this problem has been solved recently using Stirling numbers (Honarvar, Paramesran, & Lim, 2014) which paves the way to carry out image restoration in moment domain. Also, there exist a relationship between the blurred and original images in the moment domain which is one of the reasons to explore deblurring in moment domain.

1.3 Objectives

The main objectives of this thesis are

1. *To develop an algorithm in moment domain for restoring the blurred images via Euler Lagrange and Alternate Minimization techniques*

In this work the geometric moments are used for the implementation of deblurring algorithms. Geometric moment are projections of the image intensity function $f(x,y)$, onto monomials $x^p y^q$. The basis set $x^p y^q$ is not orthogonal. This non-orthogonality causes information redundancy in the extracted moments and it is more difficult to reconstruct images from the moments. Since, the problem of image reconstruction has been solved recently (Honarvar et al., 2014). This provides motivation and new opportunities to explore the potential of using geometric moments in the area of image restoration. In particular, restoration of images degraded due to Gaussian and motion blur is addressed. It involves the formulation of problem in moment domain and subsequent restoration of the blurred images using

variational method proposed in moment domain along with Euler Lagrange and alternate minimization techniques. This results in the formulation of the partial differential equations in moment domain which is iterated in time resulting in the solution that leads to the global minimum solution i.e. the restored image (Chapter 3).

2. *Blurred image restoration by estimating the PSF parameters using Weighted Geometric moments*

In the proposed method, Weighted geometric moments Weighted Geometric Moments (WGMs) which are linear combinations of the basic geometric moments are used for estimating the PSF. It has been observed that the WGM can effectively characterize the edges of the natural images subjected to varying degree of blur. Using this as the motivation, estimation of the PSF is performed using WGM as feature vectors along with the use of extreme learning machine. Once the PSF is estimated, a non-blind deblurring method in the geometric moment domain is proposed to estimate the moments of the restored image. In order to transform the restored image from moment domain to spatial domain, a fast architecture for doing this inverse transform is employed, which make use of digital filters operating as subtractors (Chapter 4).

3. *Blurred image restoration by estimating the PSF parameters using Tchebichef moments*

Generally, images are often blurred either due to Gaussian, motion or combination of both. This research is motivated by two facts: (1) Tchebichef moments (TM) are good edge descriptors. (2) Blurring of an image mostly affects the edges of an image leaving behind plains least affected. Using these two crucial observations the degraded image is restored by estimating the Gaussian blur parameters with the

use of Tchebichef Moments (TMs) and extreme learning machine (ELM). Once the blur is estimated, an existing non-blind deconvolution algorithm known as split Bregman is adopted to restore the degraded images (Chapter 5).

1.4 Organization

This thesis develops the image restoration algorithms using Geometric and Tchebichef moments. A block diagram representing the content of the thesis is shown in Figure 1.1.

The following is a summary of the content of the chapters

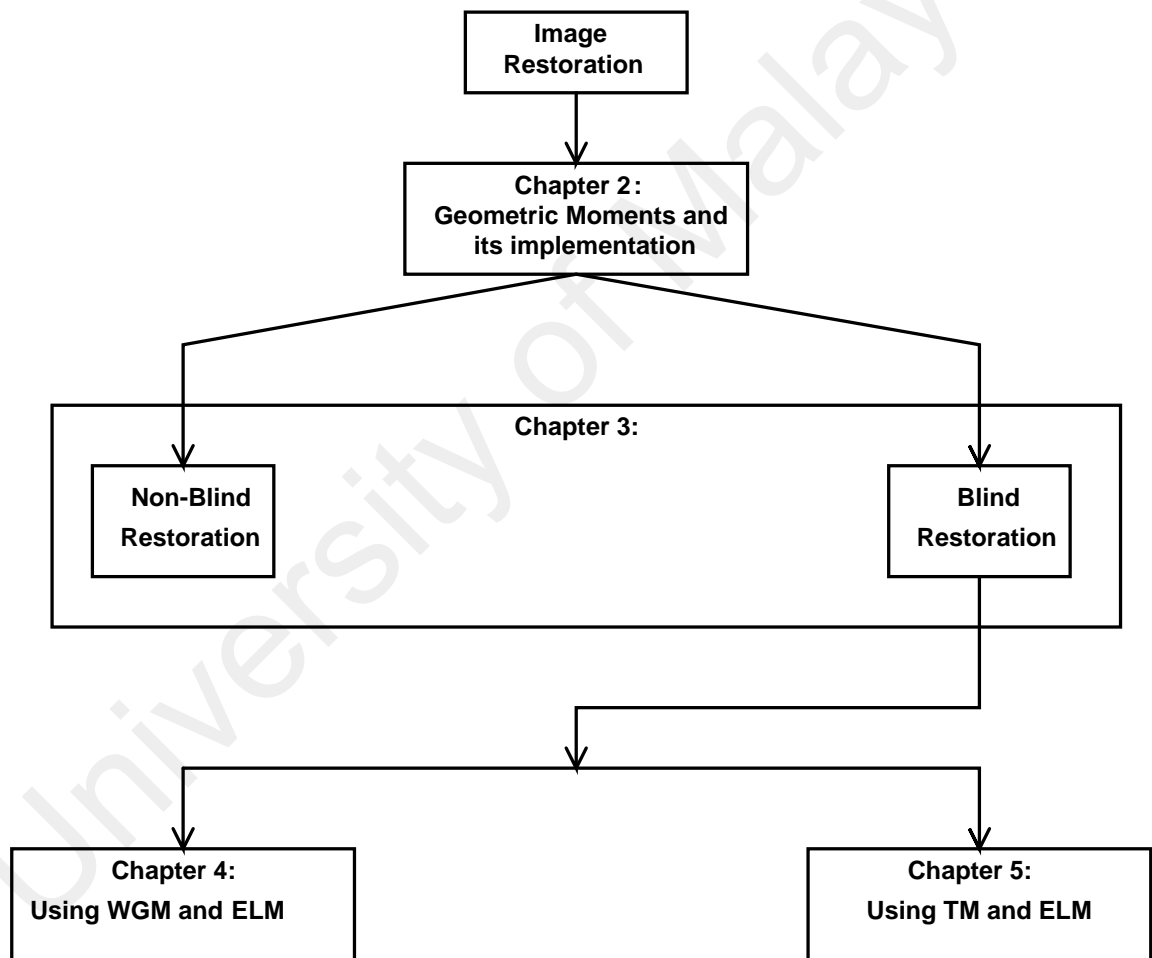


Figure 1.1: Flowchart of thesis contents

Chapter 2: Review of Geometric and Tchebichef moments for image deblurring. In this chapter a brief overview of geometric and Tchebichef moments is presented. This includes the mathematical preliminaries of both the moments and their role in the field of image processing. Further, some of the earlier work done with a specific target

of improving the computational speed of the Geometric and Tchebichef moments is discussed here. In particular, the computation of geometric moments via cascaded digital filters is explained in details. It has been shown that with the use of cascaded digital filters, geometric moments are calculated with fewer number of multiplication operations. A new cascaded digital filter structure with fast computation of geometric moments is also proposed.

Chapter 3: Restoration of blurred images using Geometric Moments. In this chapter, the use of geometric moments in the field of image deblurring is proposed. Using the developed relationship of geometric moments for original and blurred images, a mathematical formulation based on Euler-Lagrange identity and variational techniques is proposed. It uses an iterative procedure to deblur the image in moment domain. The theoretical framework is validated by a set of experiments. A comparative analysis of the results obtained using the spatial and moment domains are evaluated using a quality assessment method known as Blind/Reference-less Image Spatial Quality Evaluators (BRISQEs). The results show that the proposed method gives competitive results in terms of quality score when compared to the existing methods.

Chapter 4: Restoration of the blurred images in moment domain via Cascaded Digital Filters Image moments such as Zernike, Tchebichef and geometric moments have been widely used in image processing applications. They have useful properties to detect edges. In this chapter, one of the moment families, in particular Geometric moments (GM) can be utilized in estimating the (σ) and size (w) of the Gaussian point spread function (PSF) that degrades the images. With the knowledge of how edges vary in the presence of Gaussian blur, a method that uses low order geometric moments is proposed to estimate the PSF parameter. This is achieved by using the difference of the Weighted geometric moments (WGM) of the original and the reblurred images as feature vectors to train extreme learning machine (ELM) to estimate the PSF parameters respectively.

By estimating the PSF parameter along with using the mathematical relationship between the GMs of the original image, PSF and the blurred image a restoration method is used to obtain the deblurred GM of the sharp image. Further, a novel method that uses a cascaded digital filters operating as subtractor is used to transform the image from Geometric moment domain to spatial domain. The effectiveness of the proposed method for estimating the PSF parameters is examined using cross database validation. The results show that the proposed method in most of the cases performs better than the three existing methods when examined in terms of the visual quality evaluated using Structural Similarity Indexs (SSIMs) index.

Chapter 5: Estimation of the blur parameters using Tchebichef moments for image restoration. With the knowledge of how edges vary in the presence of the Gaussian blur, this chapter presents a method that uses low order Tchebichef moments is proposed to estimate the blur parameters: sigma (σ) and size (w). The difference of the Tchebichef moments of the original and the reblurred images is used as feature vectors to train extreme learning machine (ELM) for estimating the blur parameters (σ, w) respectively. The effectiveness of the proposed method to estimate the blur parameters is examined using cross database validation. The estimated blur parameters from the proposed method are used in the split Bregman based image restoration algorithm. A comparative analysis of the proposed method with three existing methods using all the images from the LIVE database is carried out. The results show that the proposed method in most of the cases performs better than the three existing methods in terms of the visual quality evaluated using structural similarity (SSIM) index.

Chapter 6: Conclusion. The thesis is concluded by summarizing the contents of the thesis and discussing the possibilities of future work in this direction.

1.5 Contributions

One of the contributions of this thesis is that it proposes a method of deblurring the images in the geometric moment domain. In particular, images degraded due to Gaussian and motion blur are analyzed. A mathematical framework is established in the form of partial differential equations that results in the optimum solution i.e. the deblurred image. A comparative analysis of the proposed method with the recent state of the art methods is performed. In order to provide a fair comparison with the proposed method, images are selected from the existing methods. Next contribution of this thesis is to propose a new feature vector known as Weighted geometric moment (WGM) that has an ability to describe the behavior of the edges of an image in the presence of blur. WGM along with ELM is used for the estimation of blur parameters followed by restoration performed in moment domain. Further, an inverse transform that uses cascaded digital filter as subtractor is employed to convert the image from moment domain back to the spatial domain for quality evaluations. For evaluating the performance of the proposed approach, standard databases such as LIVE, Berkeley, CSIQ, CDIQ, TID2008, VCL and Caltech are used. Using these databases provides a fair comparison with the proposed method as the images used are the same across the platform. Lastly, the use of Tchebichef moments (TM) for the estimation of blur parameters by using it as the feature vectors along with the use of extreme learning machine (ELM) is proposed. Once the PSF is estimated, widely used non-blind algorithm known as split Bregman is used to restore the original image. The performance of the proposed TM based estimation method is evaluated using the aforementioned six databases along with the images used in the existing methods with which the comparison is performed. Lastly, the quality of the restored images in all these contributions is evaluated using no-reference and full-reference image quality metrics such as SSIM, BRISQUE, VIF and FSIM available in the literature. The proposed contributions

provide competitive quality scores when compared with the other existing methods.

University of Malaya

CHAPTER 2 : REVIEW OF GEOMETRIC AND TCHEBICHEF MOMENTS FOR IMAGE DEBLURRING

This chapter starts with a brief overview of the theory of geometric and Tchebichef moments. A relationship of the geometric moments of the original image, blurred image and the point spread function (PSF) is discussed. The relationship along with the developed reconstruction algorithms to restore the deblurred image from its geometric moments are elaborated. The basic kernel of the geometric moment can be represented as an impulse response and hence digital filters can be designed to compute the geometric moments. The advantage of using cascaded digital filters is to avoid the use of multiplication algorithms which reduces the computational time of calculating the geometric moments of higher order. These digital filters can also be used to restore the image from its geometric moments.

2.1 Introduction

The task of image restoration involves deblurring of images degraded due to various distortions present in nature such as atmospheric turbulence etc. Typical restoration algorithms available in the literature are implemented in spatial, wavelet and sparse domains. Wavelet analysis turns out to be invaluable in dealing with a wide class of images with spatially localized features (Daubechies et al., 1992; Mallat, 1999). They are designed to capture most of the energy of the images and signal using fewer number of coefficients. This property turns out to be effectively used in various estimation algorithms that are based on shrinking of the wavelet coefficients (Chambolle, De Vore, Lee, & Lucier, 1998; Coifman & Donoho, 1995). Its discrete version allows for fast implementation of linear wavelet methods. The ability to capture sharp signal discontinuity with sparse coefficients makes them the suitable choice. The WV (wavelet-vaguelette) decomposition is proposed

for image deblurring problem in (Donoho, 1995b). Here, the signal obtained as an output of the deconvolution is expanded using wavelet basis. A scale-dependent shrinkage is used to estimate the signal wavelet coefficients. Further improvement in the wavelet domain is proposed in (Candes & Donoho, 2002) with a new decomposition which is better adapted to edge-dominated objects and simultaneously providing a diagonal representation of the Radon operator. It is achieved by proposing a new basis functions, such as curvelets. These are smooth, anisotropic directional functions suited for curved edge detections. An effective hybrid wavelet based deconvolution algorithm was developed in (Neelamani, Choi, & Baraniuk, 2004). It consist of Fourier-domain regularized inversion followed by noise suppression in wavelet domain. The developed algorithm performs better compared to the conventional Wiener filters and other wavelet-based image restoration algorithms. Moreover, the expectation-maximization (EM) algorithm used for image deblurring based on penalizing the maximum likelihood setting is proposed in (Figueiredo & Nowak, 2003). The algorithm implements the EM method in the wavelet domain. It is shown that the algorithm is very efficient and competitive.

The success of image restoration depends on the selection of good image prior models. In wavelet based image deblurring (Donoho, 1995a), researchers have found that the sparsity of wavelet coefficients can serve as good prior. This provides an important fact about the images, that it can be sparsely represented using a dictionary, such as DCT or wavelet bases. Nowadays, sparse representation is successfully applied in various image processing applications (Elad, Figueiredo, & Ma, 2010; Fadili & Starck, 2006). An important issue in sparse representation modeling is the proper selection of dictionary . Typically designed dictionaries, such as DCT and wavelet, has the advantages of fast implementation. However, their adaptivity to the local structures of an image is lacking. Now days, much emphasis is laid on designing the dictionaries learned from the image patches (Elad et al., 2010; Mairal, Elad, & Sapiro, 2008; Mairal, Bach, Ponce, Sapiro, &

Zisserman, 2009). This leads to state-of-the-art image deblurring.

This work addresses the task of image deblurring in moment domain. Therefore, it becomes essential to discuss their role in image processing. Geometric moments (GM) were first introduced by (M.-K. Hu, 1962). The typical application of geometric moments is in pattern recognition problems, aircraft identification, position detection, scene matching, image reconstruction, restoration, segmentation, color and character recognition (P.-T. Yap & Raveendran, 2002; P. Yap et al., 2001). Hu employed the geometric moments for the calculation of moment invariants which are invariant to translation, rotation and scale change. These invariant features are derived using non linear combination of geometric moments which are projection of the image intensity $f(x,y)$ onto the monomial $x^p y^q$. Due to this the computational cost of evaluating the geometric moments is exhaustive because of the presence of an exponential factor in its definition.

Most of the work related to the geometric moment calculations have been done using software and there are very few digital filter implementations for deriving geometric moments. In order to perform real time image processing, it is desirable to have a digital filter implementation for geometric moment calculation. Li et al. (B. Li, 1993) proposed the method of Pascal-triangle-transform to compute the geometric moments. The advantage of the approach is that it doesn't involve any multiplications.

Recently moments have gained importance in the field of image processing for performing various image related tasks like object recognition, feature extraction (Flusser, Zitova, & Suk, 2009) etc. The basic monomial kernel (x^p) of the geometric moment can be expressed as an impulse response. Hence, this paves the way for the implementation of using digital filters to compute geometric moments. An early effort was made by Mehdi Hatamian (Hatamian, 1986) who proposed a cascaded 2D digital filter structure for implementation for geometric moments for real time image processing in which the basic building block used is a single pole feed forward digital filter. The filters have been

cascaded in such a way that the structure looks quite symmetrical and thus suitable for very large scale integration. Li (B. Li, 1993) employs a fast moment calculation based on Hatamian filter method. However, the setback of Hatamian's digital filter structure is the delay involved in the feed-forward path. Wong and Siu (Wong & Siu, 1999) improves the filter configuration by employing a feedback digital filter structure with reduced computational time for calculating geometric moments.

Similarly, Tchebichef moments are also implemented using cascaded digital filters. Kotoulas et al. (Kotoulas & Andreadis, 2006) proposed an efficient method for calculating Tchebichef moments from cascaded digital filters. The fast architecture was based on Hatamian structure which calculates Tchebichef moments from geometric moments. They showed that the elapsed CPU time decreased to as low as 99.81% when compared to the recursive method proposed by Wang et al. (G. Wang & Wang, 2006). Further, Kotoulas et al. proposed a FPGA implementation for fast generation of moments (Kotoulas & Andreadis, 2008). Hardware implementation of the Tchebichef moments is also proposed by Chang et al. (K.-H. Chang, Paramesran, Asli, & Lim, 2012).

The rest of the chapter is organized as follows. Section 2.2 gives a theoretical background of Geometric moments which reviews the feed-forward and feedback based 2D digital filter method for computing the geometric moments. Further, the proposed architecture is presented for fast implementation of Geometric moments. Experimental results of the proposed architecture with the existing architectures is presented. Next, techniques for obtaining the original image back from the geometric moments are discussed. Section 2.3 presents the brief overview of Tchebichef moments and its computational aspects. Finally, Section 2.4 concludes the chapter.

2.2 Geometric Moments

For an image function $f(x,y)$, a two dimensional geometric moment m_{pq} of order $(p+q)$ is defined as

$$m_{pq} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} x^p y^q f(x,y) \quad (2.1)$$

where N, M are the number of rows and column of an image matrix $f(x,y)$.

The moment m_{00} represents the total intensity in an image. The first order moments, m_{01} and m_{10} about the intensity of the image about x-axis and y-axis respectively. The centroid of the image is defined as

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad (2.2)$$

$$\bar{y} = \frac{m_{01}}{m_{00}} \quad (2.3)$$

Central moments are translation invariants and are defined as

$$\mu_{pq} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (x - \bar{x})^p (y - \bar{y})^q f(x,y) \quad (2.4)$$

The second order central moments μ_{02} and μ_{20} provide the variance about the mean.

The central moment μ_{11} provides the covariance measure. The central moments can be used to provide translation and scale invariant moments given as

$$\eta_{pq} = \frac{(\mu_{00})^{\frac{(p+q+2)}{2}}}{(\mu_{20})^{\frac{(p+1)}{2}} (\mu_{02})^{\frac{(q+1)}{2}}} \mu_{pq} \quad (2.5)$$

where η_{pq} is invariant to scaling.

The most commonly used moment invariants proposed by Hu are given as

$$I_1 = \eta_{20} + \eta_{02} \quad (2.6)$$

$$I_2 = (\eta_{20} - \eta_{02})^2 + 2(\eta_{11})^2 \quad (2.7)$$

$$I_3 = (3\eta_{30} - \eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (2.8)$$

$$I_4 = (3\eta_{30} + \eta_{12})^2 + (3\eta_{21} + \eta_{03})^2 \quad (2.9)$$

$$I_5 = (3\eta_{30} - \eta_{12})(3\eta_{30} + \eta_{12}) [(3\eta_{30} + \eta_{12})^2 - 3(3\eta_{21} - \eta_{03})^2] + \quad (2.10)$$

$$(3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) [(3\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$I_6 = (\eta_{20} - \eta_{02}); [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + \quad (2.11)$$

$$4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$I_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \quad (2.12)$$

$$-(\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

I_1 denotes the moment of inertia around the centroid of the image while I_7 is skew invariant.

2.2.1 Relationship between blurred and original image in Geometric moments domain

The mathematical concept of moments has been around for many years and has been used in many diverse fields ranging from mechanics, statistics, pattern recognition and feature extraction. Describing images with moments instead of other more commonly used image features means that global properties of the image are used rather than local properties. In particular, the shape of boundary segments can be described quantitatively by using statistical moments such as mean, variance and other higher order moments. These statistical measures provide an important image characteristic.

In this thesis, the restoration of blurred images using moments via variational methods is presented. Geometric moments are taken for this task as they have a representative equation like (1.1) in spatial domain which relates the blurred image with the original

image and the system PSF. Therefore, to develop a similar relationship in moments domain the following derivation is performed. Geometric moments of an image $f(x,y)$ in continuous domain are defined as

$$m_{p,q}^{(f)} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y) dx dy \quad (2.13)$$

Geometric moments of blurred image $g(x,y)$ and the PSF $h(x,y)$ are defined using (2.13) as

$$m_{p,q}^{(g)} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q g(x,y) dx dy \quad (2.14)$$

$$m_{p,q}^{(h)} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q h(x,y) dx dy \quad (2.15)$$

Substituting $g(x,y) = h(x,y) * f(x,y)$ in (2.14), one can obtain

$$m_{p,q}^{(g)} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q \left\{ \int \int_{R^2} h(a,b) f(x-a, y-b) da db \right\} dx dy \quad (2.16)$$

As shown in (Flusser et al., 2009), by changing the order of integration of (2.16) results in

$$m_{p,q}^{(g)} = \int \int_{R^2} h(a,b) \left\{ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x-a, y-b) dx dy \right\} da db \quad (2.17)$$

Using a change of variable $x' = x - a$ and $y' = y - b$ in (2.17) gives

$$m_{p,q}^{(g)} = \int \int_{R^2} h(a,b) \left\{ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x' + a)^p (y' + b)^q f(x', y') dx' dy' \right\} da db \quad (2.18)$$

Substituting back $x' = x$ and $y' = y$ and binomial expanding (2.18) results in

$$m_{p,q}^{(g)} = \int \int_{R^2} h(a,b) \times \left\{ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sum_{i=0}^p \binom{p}{i} x^i a^{p-i} \sum_{j=0}^q \binom{q}{j} y^j b^{q-j} f(x,y) dx dy \right\} da db \quad (2.19)$$

Bringing the summations outside integration of (2.19) gives

$$m_{p,q}^{(g)} = \sum_{i=0}^p \binom{p}{i} \sum_{j=0}^q \binom{q}{j} \left\{ \int \int_{R^2} h(a,b) a^{p-i} b^{q-j} da db \right\} \times \left\{ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^i y^j f(x,y) dx dy \right\} \quad (2.20)$$

Finally, (2.20) is simplified using the definitions given in (2.13) and (2.15) as

$$m_{p,q}^{(g)} = \left\{ \sum_{i=0}^p \sum_{j=0}^q \binom{p}{i} \binom{q}{j} m_{p-i,q-j}^{(h)} m_{i,j}^{(f)} \right\} \quad (2.21)$$

The relationship between original image moments $m_{p,q}^{(f)}$ and blurred image moments $m_{p,q}^{(g)}$ is established using (2.21) which provides motivation for performing restoration using geometric moments. Blurring affects the edges present in an image which is quantified in moments domain as given in (2.21). For a 2D Gaussian PSF $h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$, Liu et al. (J. Liu & Zhang, 2005) derived an explicit formula for the GMs of the Gaussian PSF which is simplified as

$$m_{pq}^{(h)} = \begin{cases} 2^{-\frac{p+q}{2}} \frac{p!q!}{(\frac{p}{2})!(\frac{q}{2})!} \sigma^{p+q}, & \text{if } p \text{ and } q \text{ are even.} \\ 0, & \text{otherwise.} \end{cases} \quad (2.22)$$

Substituting (2.22) into (2.21) and taking inverse transform to derive the expression for original image GMs in terms of the blurred image GMs (Honarvar et al., 2014) as

$$m_{pq}^{(f)} = \sum_{k=0}^p \sum_{l=0}^q \binom{p}{k} \binom{q}{l} (-2)^{-\frac{k+l}{2}} \frac{k!l!}{(\frac{k}{2})!(\frac{l}{2})!} \sigma^{k+l} m_{p-k,q-l}^{(g)} \quad (2.23)$$

The restoration of deblurred geometric moments cannot be achieved easily due to its non-orthogonal basis functions $x^p y^q$. Recently, (Honarvar et al., 2014) solved this problem by using Stirling numbers. However, this method is computationally intensive. Based on the methods which use cascaded digital filters operating as an adder to compute geometric moments, a technique that uses these filters as subtractor is discussed. These subtractor based cascaded digital filters can restore the image from its geometric moments faster.

Figure 2.1 shows the flowchart of the chapter content. It consist of two paths, namely forward path and inverse path. The forward path calculates the geometric moments m_{pq}^f from the image $f(x,y)$ via the digital filter output Y . It consist of cascaded digital filters and Geometric moment calculation blocks. This will be discussed in detail in Sections 2.2.2 and 2.2.3 . While the inverse path is used to restore back the image $f(x,y)$ from its geometric moments m_{pq}^f via digital filter output Y . It consists of an inverse geometric moment calculation followed by the use of subtractor based cascaded digital filter blocks. The detail description of these blocks will be carried out in Sections 2.2.4 and 2.2.5

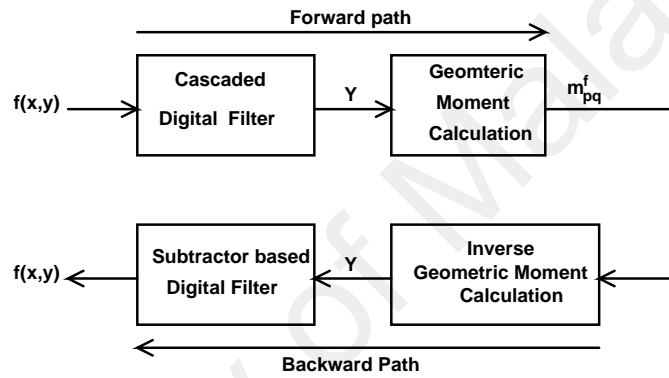


Figure 2.1: Flowchart of the chapter content

2.2.2 Computation of Geometric moments using Feed-forward digital filter

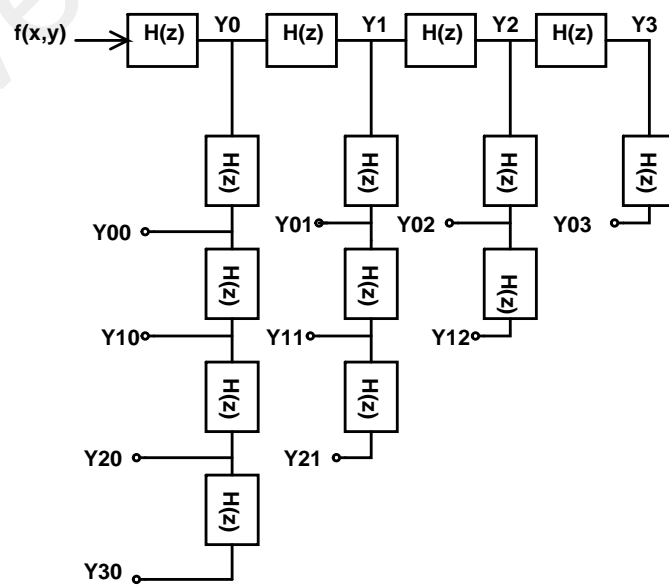


Figure 2.2: 2D digital filter structure for calculating geometric moment up to 3rd order

The geometric moments given in Eq. 2.1 can be implemented using cascaded digital filters and the resultant structure has been shown in Figure 2.2. Here, the implementation is restricted to third order moments for a two dimensional image $f(x,y)$, which can be extended easily for higher order moments. These cascaded digital filters operate as adders. These filters are of feed-forward and feedback filters. In this section the theory about feed-forward filters is discussed, followed by feedback filters in the next section.

The feed-forward filter structure has been shown in Figure 2.3. Its input-output relationship is given as

$$y(n) - y(n-1) = x(n-1) \quad (2.24)$$

Taking Z-transform of Eq. 2.24 results in

$$H_1(z) = \frac{Y(z)}{X(z)} = \frac{z^{-1}}{1 - z^{-1}} \quad (2.25)$$

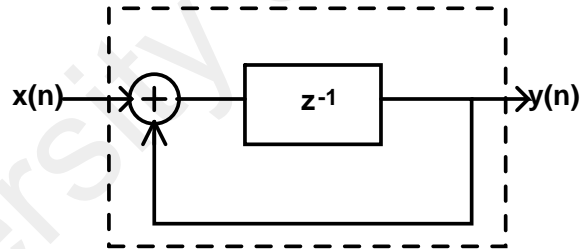


Figure 2.3: Feed-forward filter

The basic kernel of geometric moments can be represented in terms of a filter whose impulse response can be given by

$$h(n) = n^p u(n) \quad (2.26)$$

where p is the order of the required moment and $u(n)$ is the unit step function. The higher order filter structures for calculating the moments can be implemented by cascading the single pole digital filters to form a multi-pole filter with a transfer function as

$$h_p(z) = 1/(z-1)^{p+1} \quad (2.27)$$

Thus, one can use filter banks to calculate various orders of moments. In general, for each value of p , $h_p(z)$ has poles at $z = 1$ and no zeros. For one dimensional sequence $(a_1, a_2, a_3 \dots a_N)$, the output of the digital filters (Hatamian, 1986) can be evaluated at point $n = N$ using convolution theorem as

$$y(N) = \sum_{i=1}^N x(i)(N-i)^p \quad (2.28)$$

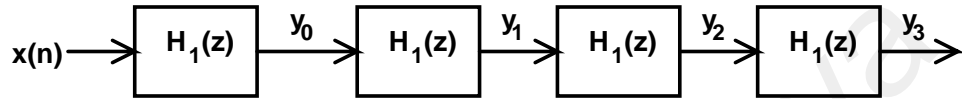


Figure 2.4: One dimensional feed-forward filter structure

Consider a one-dimensional input of length 4-input sequence i.e. $N = 4$, and $p = 0, 1, 2, 3$. The corresponding impulse response for the transfer function given in (2.26) for the filter structure shown in Figure 2.4 is given as

$$h_0(n) = u(n-1) \quad (2.29)$$

$$h_1(n) = (n-2)u(n-2) + u(n-2) \quad (2.30)$$

$$h_2(n) = \frac{1}{2}(n-3)^2 u(n-3) + \frac{3}{2}(n-3)u(n-3) + u(n-3) \quad (2.31)$$

$$h_3(n) = \frac{1}{6}(n-4)^3 u(n-4) + (n-4)^2 u(n-4) + \frac{11}{6}(n-4)u(n-4) + u(n-4) \quad (2.32)$$

It can be observed from (2.29)-(2.32) that the impulse response of the cascaded feed-forward filter are a linear combination of the delayed version of the desired impulse response given in (2.26).

Now, the working principle of the cascaded digital filter structure (Figure 2.2) proposed by Hatamian which used feed-forward digital filter $H_1(z)$ given in (2.25) in terms of

their architecture and clocking schemes is discussed. In order to understand the working principle of this structure, the following test matrix is used

$$\mathbf{A} = \begin{bmatrix} 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} \quad (2.33)$$

as an input to the system. In this operation, the data $\mathbf{A}(3,4)$ enters the filter at the first clock pulse. This is followed by $\mathbf{A}(3,3)$, $\mathbf{A}(3,2)$ and $\mathbf{A}(3,1)$. Then the data $\mathbf{A}(2,4)$, $\mathbf{A}(2,3)$, $\mathbf{A}(2,2)$, $\mathbf{A}(2,1)$, $\mathbf{A}(1,4)$, $\mathbf{A}(1,3)$, $\mathbf{A}(1,2)$, and $\mathbf{A}(1,1)$ will be the inputs to the filter. The row outputs Y0 and Y1 after every clock pulse are given in Figure 2.5a, where Δ is the delay in resetting the system to allow new set of inputs. These outputs are applied to column filters to give the column output Y00, Y01, Y10 and Y11. The outputs of the column filter after 5^{th} , 11^{th} , 17^{th} , 23^{rd} , and 29^{th} clock period is shown in Figure 2.5b. The column outputs Y00, Y01, Y10, Y11 after 23^{rd} , 24^{th} , 29^{th} and 30^{th} clock pulse are 126, 284, 330 and 740 respectively. The clocking scheme used in feed-forward structure is very simple and the details of this clocking scheme can be seen in reference(Hatamian, 1986).

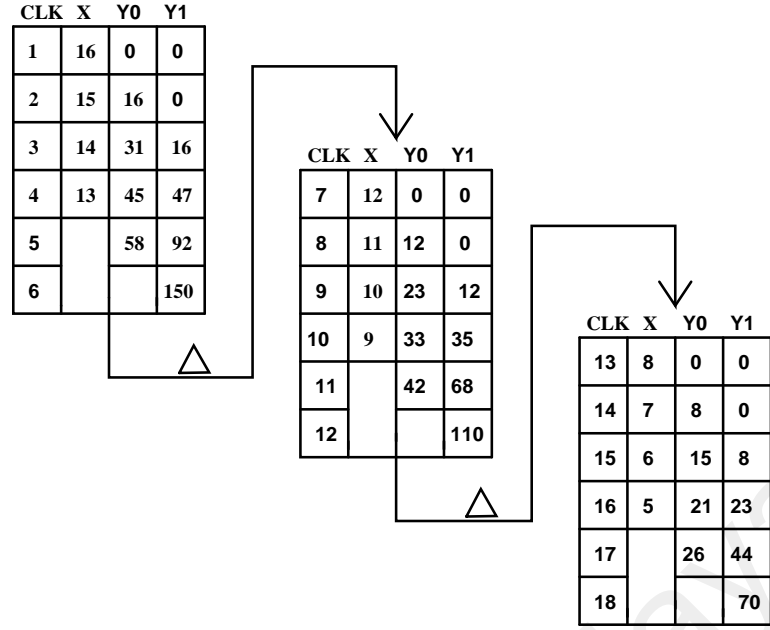
2.2.3 Computation of Geometric moments using Feedback digital filter

Another popular filter is the feedback type structure, shown in Figure 2.6, whose input-output relationship is

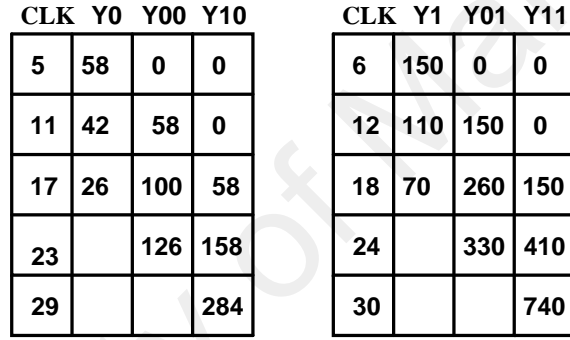
$$y(n) - y(n-1) = x(n) \quad (2.34)$$

Taking Z-transform of Eq. 2.34 results in

$$H_2(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - z^{-1}} \quad (2.35)$$



(a)



(b)

Figure 2.5: Feed-forward based structure (a) Row Filtering (b) Column Filtering Operation

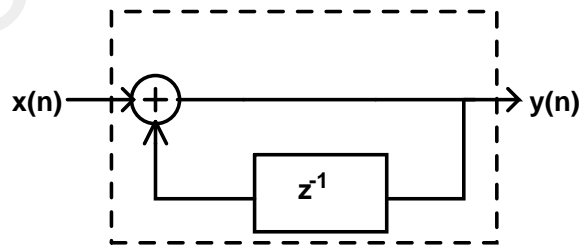


Figure 2.6: Feedback filter

To generate higher order moments for 1D case, the transfer function of the cascaded digital feedback filter is given as

$$h_p(z) = \left(\frac{1}{1 - z^{-1}} \right)^{p+1} \quad (2.36)$$

where p is the order of moment. The MacLaurian expansion of (2.36) is given as

$$1 + \binom{p+1}{1} z^{-1} + \binom{p+2}{2} (z^{-1})^{-2} + \binom{p+3}{3} (z^{-1})^{-3} + \dots + \binom{p+k}{k} (z^{-1})^{-k} \quad (2.37)$$

The impulse response of the feedback filter can be found by taking inverse of the Maclaurin expansion in (2.37) and is given as

$$h_p(n) = \binom{p+n}{n} u(n) \quad (2.38)$$

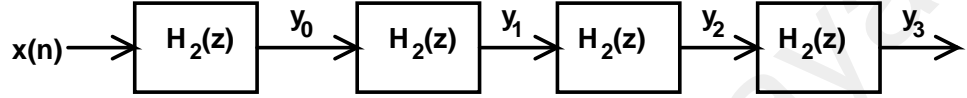


Figure 2.7: One dimensional feedback filter structure

Consider a one-dimensional input of length 4-input sequence i.e. $N = 4$, and $p = 0, 1, 2, 3$. The corresponding impulse response for the transfer function given in (2.38) for the filter structure shown in Figure 2.7 is given as

$$h_0(n) = u(n) \quad (2.39)$$

$$h_1(n) = (n+1)u(n) \quad (2.40)$$

$$h_2(n) = \frac{1}{2}[(n+1)^2 + (n+1)]u(n) \quad (2.41)$$

$$h_3(n) = \frac{1}{6}[(n+1)^3 + 3(n+1)^2 + 2(n+1)]u(n) \quad (2.42)$$

It can be observed from (2.39)-(2.42) that the impulse response in case of feedback filter starts at $n = 0$, which means that the response time of the feedback filters is not affected by the delay element of the filter.

The cascaded digital filter structure (Figure 2.2) for moment extraction proposed by Wong (Wong & Siu, 1999) used feedback filter shown in Figure 2.6. In order to understand its working, same test matrix \mathbf{A} has been taken as an input. In this operation, the data $\mathbf{A}(3,4)$ enters the filter at first clock pulse. This is followed by $\mathbf{A}(3,3)$, $\mathbf{A}(3,2)$

and $A(3,1)$. Then the data $A(2,4)$, $A(2,3)$, $A(2,2)$, $A(2,1)$, $A(2,1)$, $A(1,4)$, $A(1,3)$, $A(1,2)$, $A(1,1)$ will be input to the filter. The row outputs $Y0$, $Y1$ and $Y2$ after every clock pulse are given in Figure 2.8a. Here Δ represents the delay in resetting the system to allow new set of inputs. These outputs are applied to the column filters to give column output $Y00$, $Y01$, $Y02$, $Y03$, $Y10$, $Y11$, $Y12$, $Y20$, $Y21$, and $Y30$. The outputs obtained from the column filters after 4^{th} , 10^{th} , 16^{th} clock period have been shown in Figure 2.8b. The column outputs $Y00$, $Y01$, $Y02$, $Y03$, $Y10$, $Y11$, $Y12$, $Y20$, $Y21$, and $Y30$ after 16^{th} clock pulse have been found to be 126, 330, 675, 1197, 284, 740, 1510, 500, 1300 and 774, respectively.

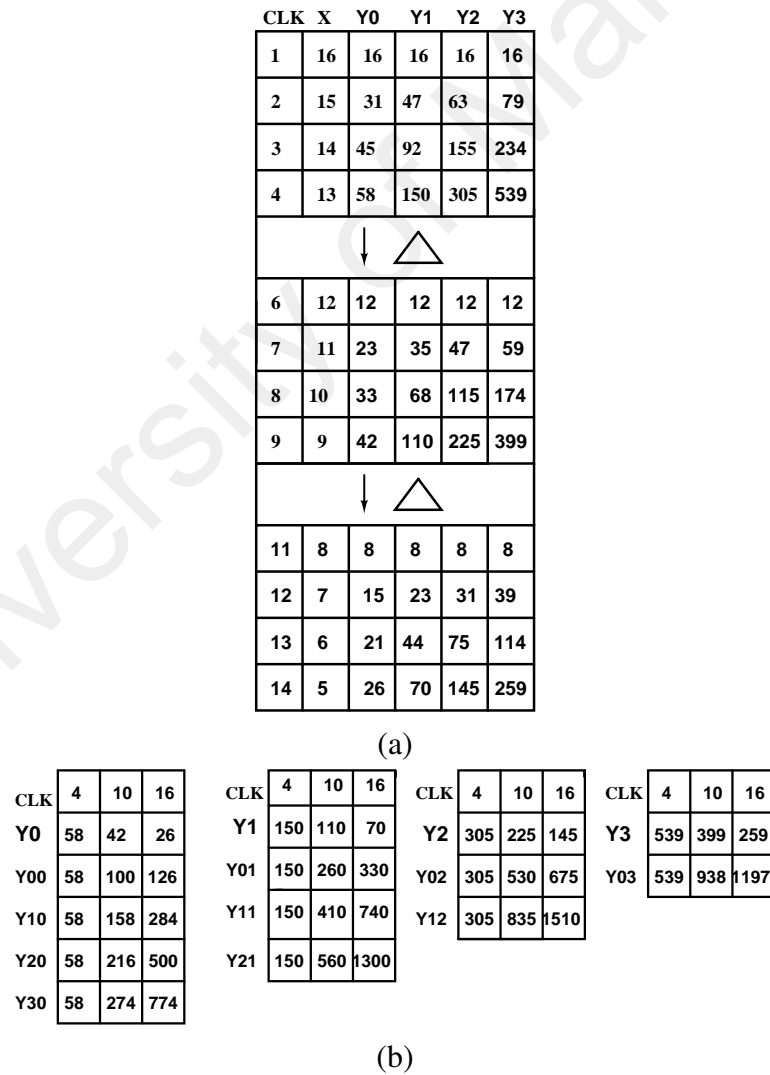


Figure 2.8: Feedback based structure (a) Row Filtering (b) Column Filtering Operation

From Figs. 2.5 and 2.8, two significant differences in the working of both the structures is observed

1. In feed-forward structure row filter outputs (Y_0 and Y_1) and column filter outputs (Y_{00} and Y_{10}) occurs with a delay of one clock cycle (shown by the presence of zeros in Figure 2.5). However, in the case of feedback structure (Figure 2.8) there is no such delay.
2. The final column outputs obtained using feedback structure occur earlier as compared to feed-forward. For example, in the case of feedback structure, output Y_{00} is obtained at 16^{th} clock cycle while in feed-forward structure it is obtained at 23^{rd} clock cycle.

These differences are the use of feedback filters in Wong's system which increase the speed and simplify the clocking.

Though, the Wong structure is faster (Wong & Siu, 1999) as compared to Hatamian but it was implemented only in software. The architecture implementation of this system is not available. Figure 2.9 shows the proposed 2D digital filter structure based on feedback filters (Figure 2.6) to obtain output Y 's for 3^{rd} order Geometric moments calculations.

To speed up the Wong's feedback structure shown in Figure 2.9 further, an architecture by making use of dual edge triggered flip flops (D-type) (Afghahi & Yuan, 1991) in feedback filters is proposed. However, the core structure of the 2D digital filter remains the same as shown in Figure 2.9. Details of the work can be found in (Kumar & Paramesran, 2014).

The outputs Y 's from the cascaded digital filter (Figure 2.9) are to be processed further in order to calculate the geometric moments matrix, \mathbf{M} . Hence, the output of

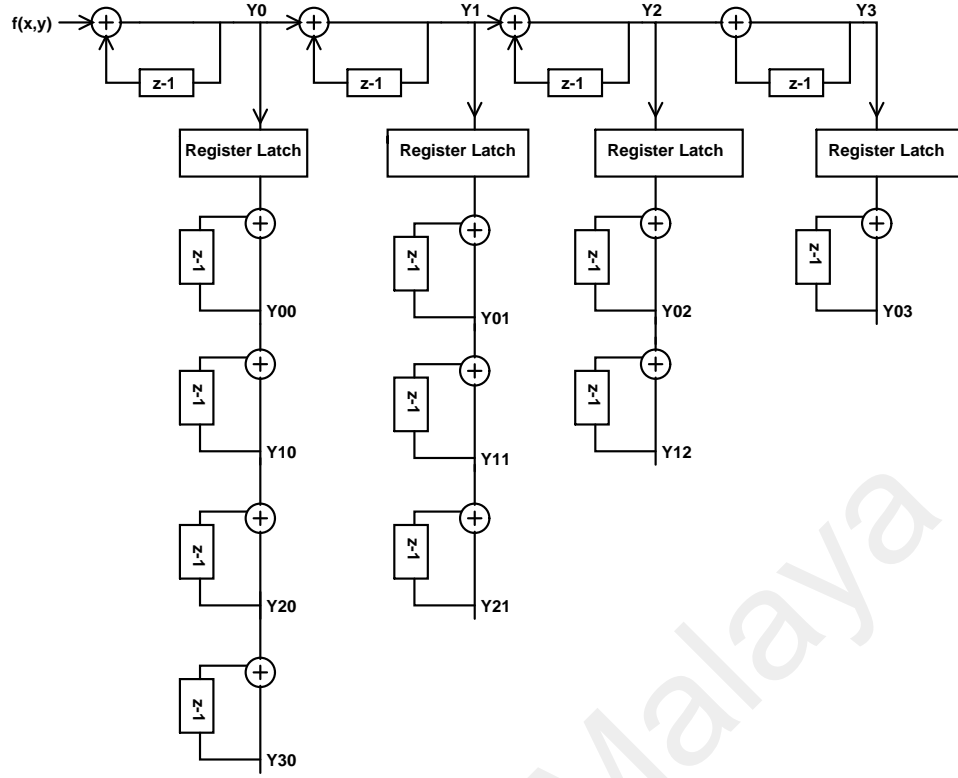


Figure 2.9: Proposed 2D digital filter structure for geometric moment calculation

digital filters passes through an appropriate digital block which implements following transformation

$$\mathbf{M} = \mathbf{C}\mathbf{Y}\mathbf{C}^T \quad (2.43)$$

where, \mathbf{M} is the geometric moments matrix, \mathbf{Y} is the 2D digital filter output matrix defined as

$$\mathbf{Y} = \begin{bmatrix} Y_{00} & Y_{01} & Y_{02} & Y_{03} & \dots & Y_{0N} \\ Y_{10} & Y_{11} & Y_{12} & Y_{13} & \dots & Y_{1N} \\ Y_{20} & Y_{21} & Y_{22} & Y_{23} & \dots & Y_{2N} \\ Y_{30} & Y_{31} & Y_{32} & Y_{33} & \dots & Y_{3N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Y_{N0} & Y_{N1} & Y_{N2} & Y_{N3} & \dots & Y_{NN} \end{bmatrix} \quad (2.44)$$

The coefficients of matrix C in case of Hatamian's approach is defined as (Al-Rawi, 2008)

$$C_{p,r} = \begin{cases} 0 & (p > 0, r = 0) \text{ or } (r > p) \\ 1 & p = 0, r = 0 \\ r(C_{p-1,r-1} - C_{p-1,r}) & \text{otherwise} \end{cases} \quad (2.45)$$

while for Wong's approach is given as (Al-Rawi & Jie, 2002)

$$C_{p,r} = \begin{cases} 0 & r > p \\ (-1)^p & r = 0, p \geq 0 \\ rC_{p-1,r-1} - (r+1)C_{p-1,r} & r > 0, p > 0 \end{cases} \quad (2.46)$$

2.2.4 Restoration of original image using Subtractor based digital filter

Here, the restoration of original image from its geometric moments using cascaded digital filters is discussed. Once the digital filter output Y is obtained from cascaded digital filters (Figure 2.9), Geometric moments are obtained using

$$m_{p,q} = \sum_{r=0}^p \sum_{s=0}^q C_{pr} C_{qs} Y_{rs} \quad (2.47)$$

where, C is the transformation matrix given as (2.45) for feed-forward architecture and (2.46) for feedback architecture (Al-Rawi & Jie, 2002). From the GMs of the original image given in (2.47), main focus is to restore the original image from its geometric moments by using an inverse moment transform based on the feedback architecture proposed by Wong et al. (Wong & Siu, 1999). For this, there is a need to calculate the inverse of the matrix C given in (2.46) as

$$E = C^{-1} \quad (2.48)$$

Once the inverse is calculated, the digital filter output Y can be calculated from the re-stored GMs of the sharp image as

$$Y_{p,q} = \sum_{r=0}^p \sum_{s=0}^q E_{pr} E_{qs} m_{rs} \quad (2.49)$$

Using the digital filter output Y given in (2.49), cascaded subtractor circuit to transform the digital filter outputs back into the image $f(x,y)$ in the spatial domain is discussed. To explain the inverse transform process, 1D signal sequence (Y_1, Y_2, \dots, Y_N) of size $1 \times N$ is taken for illustration which then generalized to 2D images later. For a 1D sequence, a complete cascaded subtractor digital filter is shown in Figure 2.10.

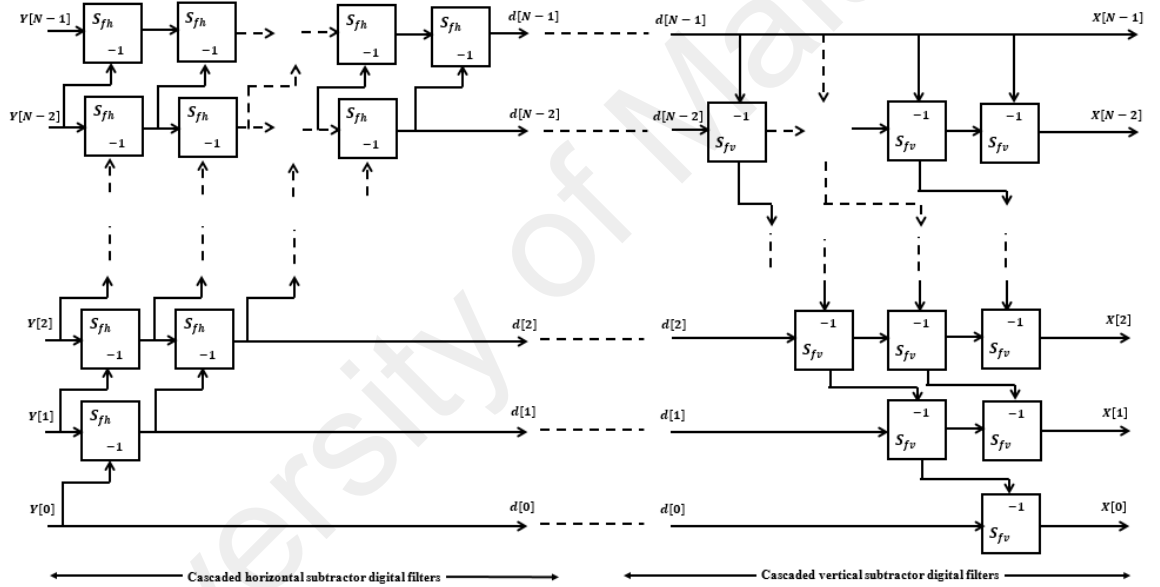


Figure 2.10: Subtractor block diagram for N input sequence

It comprises of horizontal subtractor filters (S_{hv}) and vertical filters (S_{fv}). The output of the subtractor is the reconstructed input data $(X_0, X_1, X_2, \dots, X_N)$. Next, the details of both the horizontal and vertical subtractors is discussed in detail as follows

2.2.4.1 Horizontal Cascaded Subtractor Digital Filters

The basic building block of a horizontal subtractor digital filter is shown in Figure 2.11.

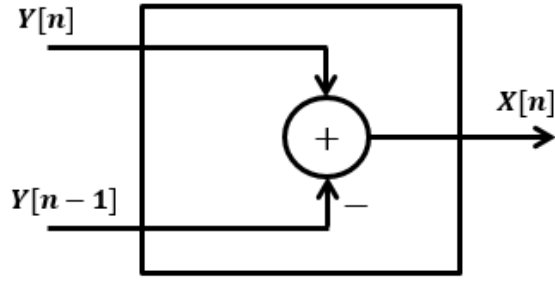


Figure 2.11: Basic building block of horizontal subtractor digital filter (S_{fh})

Its transfer function $H_{h0}(z)$ is expressed as

$$H_{h0}(z) = 1 - z^{-1} \quad (2.50)$$

while for the series of cascaded horizontal subtractor digital filters for N input sequence is given as

$$H_{hN}(z) = (1 - z^{-1})^{N+1} \quad (2.51)$$

2.2.4.2 Vertical Cascaded Subtractor Digital Filters

Once all the diagonal values are obtained, the values are inserted into the vertical subtractor digital filters. Here, the subtraction process is performed until all the image pixel intensity values are computed. The image pixel intensity values are denoted as X_0 to X_{N-1} in Fig 2.10.

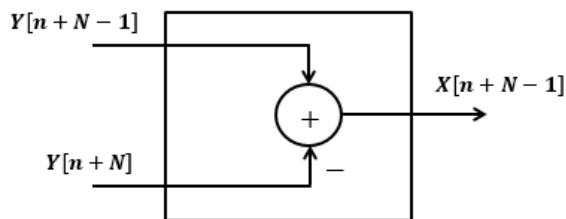


Figure 2.12: Basic building block of vertical subtractor digital filter (S_{fv})

The basic building block of vertical subtractor digital filter is shown in Figure 2.12 and the transfer function in the Z-transform domain is given as

$$H_{v0}(z) = 1 - z \quad (2.52)$$

For N input sequence, to compute x_n of the image pixel intensity value where $0 \leq n < N - 1$, there is a need of $n + 1$ vertical subtractor digital filters connected in cascade. The transfer function of the cascaded vertical subtractor digital filters of N input sequence can be expressed generally as

$$H_{vn}(z) = (1 - z)^{n+1} \quad \text{for } 0 \leq n < N - 1 \quad (2.53)$$

2.2.4.3 Working Principle of the 1D Subtractor Circuit

The working principle of the complete subtraction process can be explained using subtractor matrix. In general, the size of the subtractor matrix depends on the image size. For $N \times N$ image, the size of the subtractor matrix is $(N + 1) \times (N + 1)$ and S_{ij} represents the elements of the subtractor matrix where $i = 0, 1, \dots, N - 1$ and $j = 0, 1, \dots, N$. Initially, all the elements in the subtractor matrix are set to zero.

Thus, for an image of 4×4 , the subtractor matrix is a 5×5 as shown in Figure 2.13. The red triangle represents the working principle of the horizontal subtractor digital filters, the blue rectangle represents the intersection process between the horizontal and vertical subtractor digital filters and the green triangle represents the working principle of the vertical subtractor digital filters.

As an example, consider an arbitrary 1-dimensional digital filter the output of $(y_0, y_1, y_2, y_3) = (10, 27, 55, 97)$ are inserted into the subtractor circuit which consist of 12 cascaded subtractor digital filters as shown in Figure 2.14. This process can be represented using the process shown in Figure 2.15(a). Next, the horizontal subtraction process, $S_{ij} = S_{i-1,j} - S_{i-1,j+1}$ is performed to the input until all the diagonal values are computed as shown in Figure 2.15(b). For simplicity, the zero values are removed from the subtractor matrix in this example. These diagonal values $(3, 11, 17, 10)$ are denoted as d_0 to d_3 respectively in Figure 2.14.

The last row of the subtractor matrix has the same value as the last diagonal value,

$y_3(n)$	$y_2(n)$	$y_1(n)$	$y_0(n)$	$x(n)$
S_{00}	S_{01}	S_{02}	S_{03}	S_{04}
S_{10}	S_{11}	S_{12}	S_{13}	S_{14}
S_{20}	S_{21}	S_{22}	S_{23}	S_{24}
S_{30}	S_{31}	S_{32}	S_{33}	S_{34}

Figure 2.13: Subtractor matrix

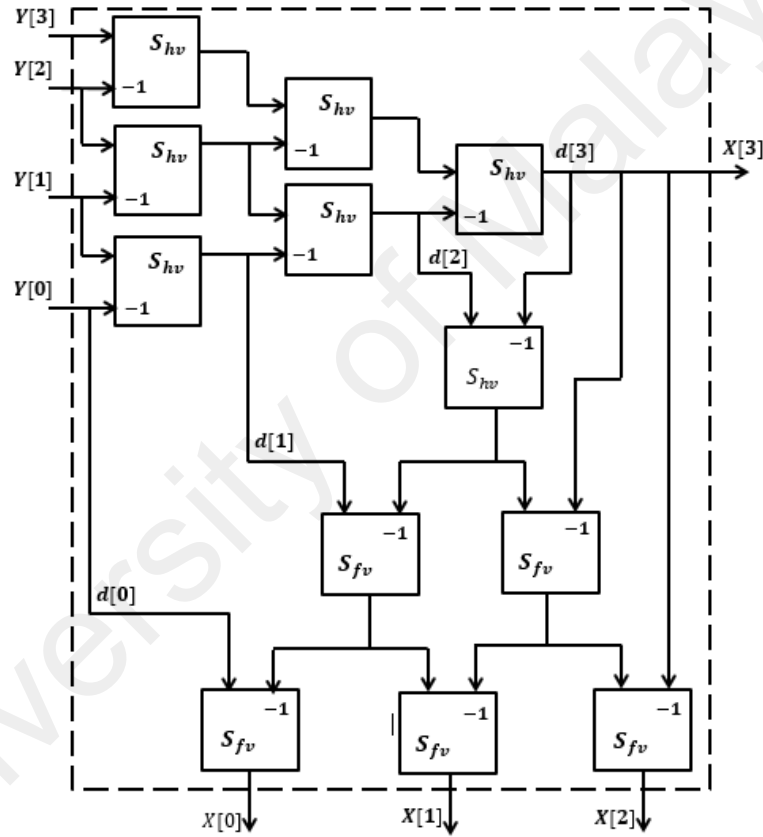


Figure 2.14: 1D cascaded subtractor digital filter for 4-input sequence

d_3 as shown in Figure 2.15(c). This is the intersection process between the horizontal and vertical subtractor digital filters. Finally, the vertical subtraction process, $S_{ij} = S_{i,j-1} - S_{i+1,j+1}$ is performed to complete the subtractor matrix as shown in Figure 2.15(d). The image pixel intensity can be found in the last column of the subtractor matrix. For this example, the image pixel intensity values are $(x_0, x_1, x_2, x_3) = (1, 4, 2, 3)$.

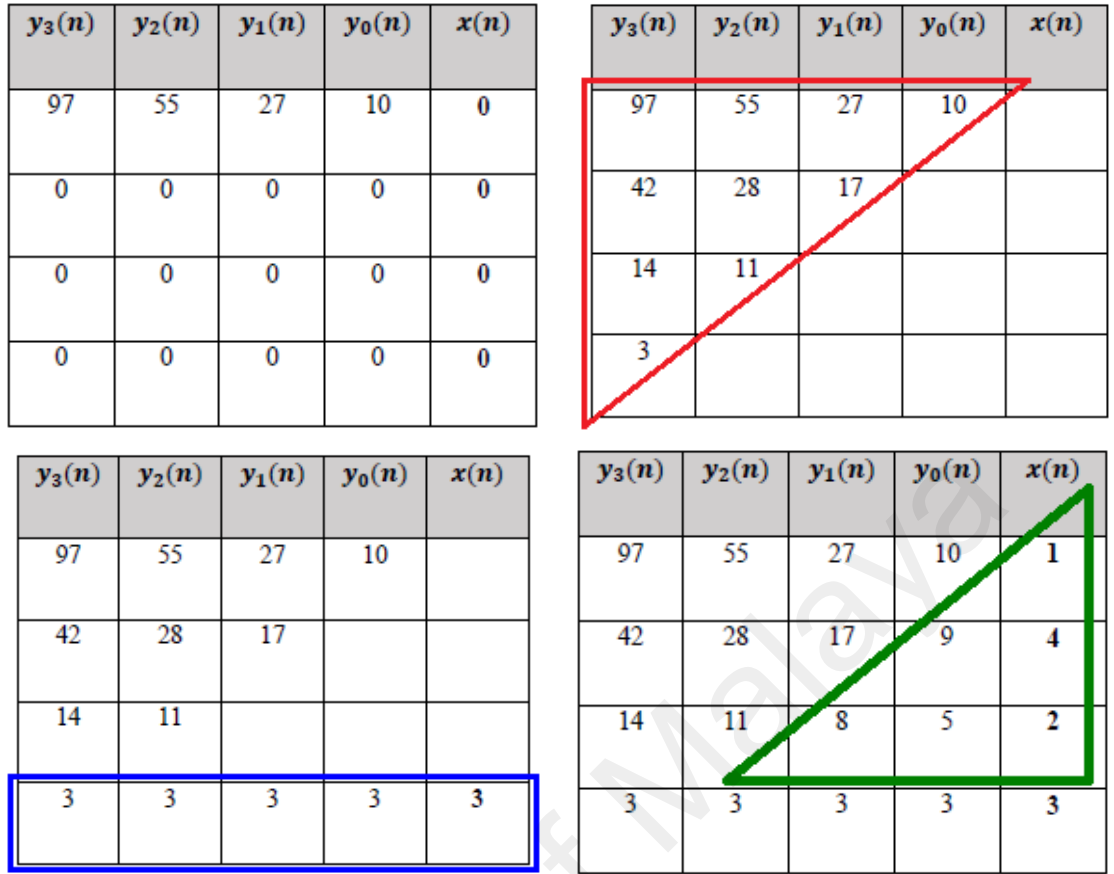


Figure 2.15: Data flow in subtractor matrix shown in Fig 2.13

2.2.4.4 2D Subtractor Circuit

For 2-dimensional case, the 2D subtractor circuit is formed by arranging multiple 1D subtractor circuits horizontally and vertically. The number of 1D subtractor circuit required are depending on the size $N \times N$ of the square image used, i.e., a 4×4 image reconstruction requires 4 vertically arranged 1D subtractor circuits, S_v and 4 horizontally arranged 1D subtractor circuits, S_h . Both S_v and S_h in this case are similar to the 1D cascaded digital filters as shown in Figure 2.14.

In general, the number of subtractor digital filters for the 2D subtractor circuit is represented by $2N^2(N-1)$ and the block diagram is shown in Figure 2.16. Consider a 4

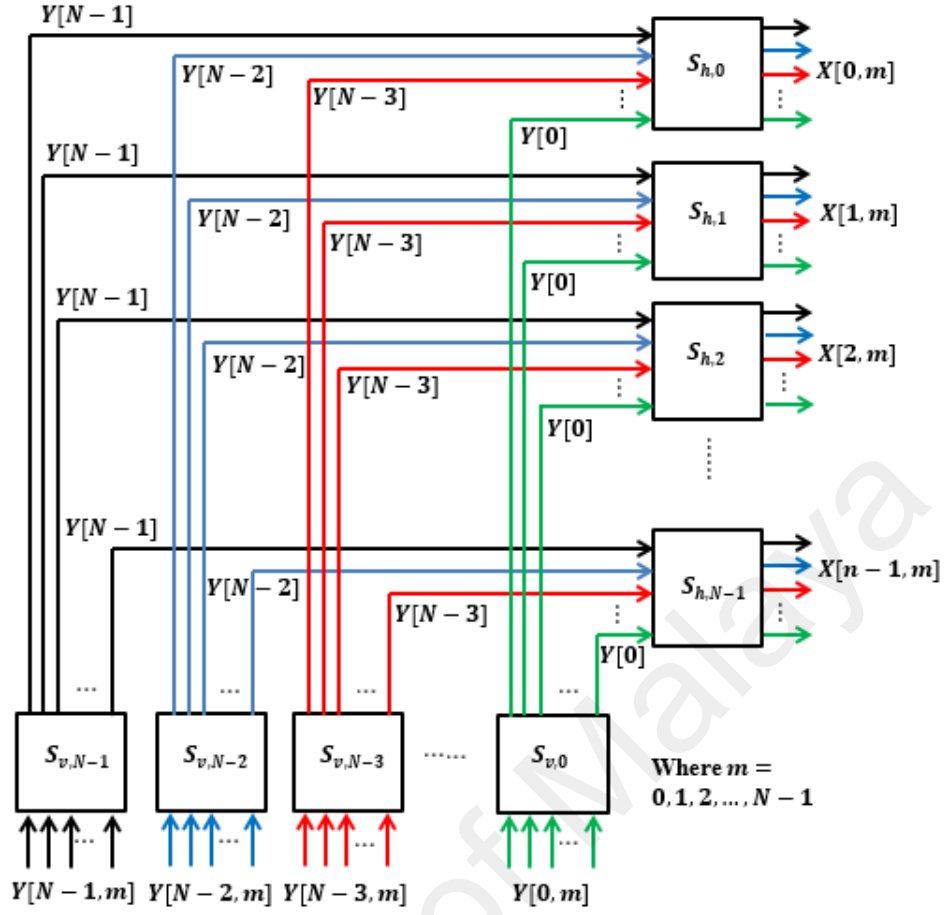


Figure 2.16: Block diagram of the 2D subtractor circuit for $N \times N$ image

$\times 4$ digital filters output as shown below.

$$y(m,n) = \begin{bmatrix} y_{00} & y_{01} & y_{02} & y_{03} \\ y_{10} & y_{11} & y_{12} & y_{13} \\ y_{20} & y_{21} & y_{22} & y_{23} \\ y_{30} & y_{31} & y_{32} & y_{33} \end{bmatrix} \quad (2.54)$$

In the 2D case, the digital filter outputs are inserted row by row into the 1D vertical subtractor circuits, $S_{v,N-1}$, i.e., for row 0, y_{00} to y_{03} are inserted into $S_{v,0}$; for row 1, y_{10} to y_{13} are inserted into $S_{v,1}$ and so on. Each row will undergo 1D subtractor processes as explained in the previous section. Then, the outputs of the 2D vertical subtractor circuit are inserted accordingly into the 1D horizontal subtractor circuits, S_h . The 1D subtractor process is repeated at this stage. The restored image is reconstructed using the final outputs of the 2D subtractor circuit.

To understand more about the process, consider an artificial test image with the following the digital filter outputs matrix as follows:

$$y(m,n) = \begin{bmatrix} 304 & 750 & 1490 & 2597 \\ 692 & 1710 & 3400 & 5929 \\ 1316 & 3250 & 6460 & 11263 \\ 2226 & 5490 & 10905 & 19005 \end{bmatrix}$$

Using the 2D subtractor circuit as shown in Figure 2.16, the first row values of the digital filter output matrix are inserted into the first 1D vertical subtractor digital circuit $S_{v,0}$, the second row values of the digital filter output matrix are inserted into the second 1D vertical subtractor digital circuit $S_{v,1}$ and so on. All these values will undergo the subtraction process and the resulting values will then be inserted into the horizontal subtractor circuit.

The first output values from each 1D vertical subtractor digital circuit will then be inserted into the first 1D horizontal subtractor digital filter $S_{h,0}$, the second output values into the second 1D horizontal subtractor digital filter and so on. The final values obtained from this process are the recovered artificial test image pixel intensity, $f(m,n)$ and is given as follows:

$$f(m,n) = \begin{bmatrix} 27 & 26 & 25 & 24 \\ 18 & 17 & 16 & 15 \\ 20 & 21 & 22 & 23 \\ 14 & 13 & 12 & 11 \end{bmatrix}$$

2.2.4.5 Experimental studies

In this section, subtractor method is validated and evaluated through a series of experiments. Here, the reconstruction process is explored where grayscale images are tested. The CPU elapsed time of the reconstruction of images is evaluated. In this experiment,

the reconstruction process to two grayscale images as shown in Figure 2.17 with the resolution of 64×64 , 128×128 and 256×256 is performed. The reconstruction process was performed using the complete set of the geometric moments. For this, geometric moments of the images are computed and used as the input to the reconstruction process.

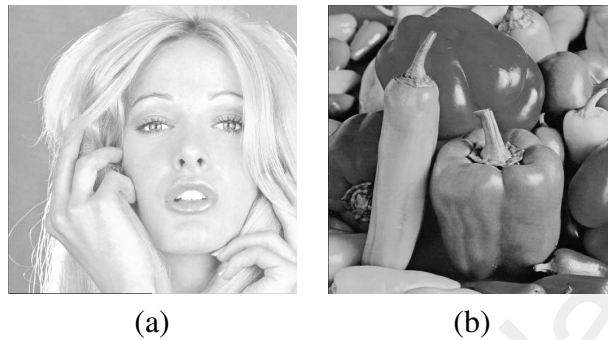




Figure 2.17: Grayscale images used in the reconstruction process (a) 'blonde' image (b) 'pepper' image

The CPU elapsed time is computed for this experiment and the experiment results are shown in Table 2.1. Using the subtractor method, the digital filter outputs is computed directly using the geometric moments and the inverse coefficient matrix. Moreover, subtraction operation used in the second process reduced larger values to smaller values. Thus, the computational complexity is reduced since both processes use direct and simple computations.

Table 2.1: CPU time (sec) for grayscale images using Subtractor circuit

Original Image	Size: 64×64	Size: 128×128	Size: 256×256
	10.23	25.12	35.65
	10.15	25.71	35.56

2.2.5 Restoration of original image using Stirling Numbers

Images can be easily constructed from the finite sets of orthogonal moments. But, geometric and complex moments basis function does not satisfy the orthogonal criterion, as a result reconstruction from these moments become difficult. Recently (Honarvar et al., 2014) accomplished the task of image reconstruction by using the relationship between geometric moments and Stirling numbers of the second kind. The Stirling number of second kind $S_2(p, k)$ is given as

$$x^p = \sum_{k=0}^p k! S_2(p, k) \binom{x}{k} \quad (2.55)$$

where x^p is the 1D monomial kernel of geometric moment. By substituting (2.55) into 1D form of (2.1) results in

$$m_p = \sum_{x=1}^N \sum_{k=0}^p k! f(x) S_2(p, k) \binom{x}{k} \quad (2.56)$$

By using the relationship between the Stirling numbers of first and second kinds, the original sequence $f(x)$ can be restored from its geometric moments and is given as

$$f(x) = \sum_{p=0}^{N-1} \sum_{i=0}^{N-1} \frac{(-1)^{i-x}}{i!} \binom{i}{x} S_1(i+1, p+1) m_p \quad (2.57)$$

Generalizing (2.57) for 2D images of size $N \times N$ as

$$f(x, y) = \sum_{p=0}^{N-1} \sum_{q=0}^{M-1} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \frac{(-1)^{i+j-x-y}}{i! j!} \binom{i}{x} \binom{j}{y} S_1(i+1, p+1) S_1(j+1, q+1) m_{pq} \quad (2.58)$$

Since geometric moments are related to Stirling numbers and an inverse exist between Stirling numbers of the first and second kind, Eq. 2.58 provides a a mathematical relationship to obtain the original image from its geometric moments.

2.3 Tchebichef Moments

The Tchebichef moment of order $(p + q)$ for an image with intensity function $f(x, y)$ has been defined as

$$T_{p,q} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \tilde{t}_p(x; N) \tilde{t}_q(y; N) f(x, y) \quad (2.59)$$

where, the image is of size $N \times N$ and $\tilde{t}_p(x; N)$ and $\tilde{t}_q(y; N)$ are the normalized Tchebichef polynomials defined as (Mukundan et al., 2001b)

$$\tilde{t}_p(x; N) = \frac{t_p(x; N)}{\sqrt{\rho(p, N)}}, \quad \tilde{t}_q(y; N) = \frac{t_q(y; N)}{\sqrt{\rho(q, N)}} \quad (2.60)$$

and $t_n(x; N)$ is the n th-order N -point Tchebichef polynomial defined as

$$t_n(x; N) = n! \sum_{k=0}^n (-1)^{n-k} \binom{N-1-k}{n-k} \binom{n+k}{n} \binom{x}{k} \quad (2.61)$$

For simplicity, the convention $\tilde{t}_n(x)$ is adopted to represent $\tilde{t}_n(x; N)$. The orthonormal version of Tchebichef polynomial $\tilde{t}_n(x)$ is calculated using the recurrence relation as

$$\begin{aligned} \tilde{t}_n(x) &= \alpha_1(2x + 1 - N)\tilde{t}_{n-1}(x) + \alpha_2\tilde{t}_{n-2}(x) \\ n &= 2, 3, \dots, N-1; x = 0, 1, \dots, N-1 \end{aligned} \quad (2.62)$$

where

$$\alpha_1 = \frac{1}{n} \sqrt{\frac{4n^2 - 1}{N^2 - n^2}} \quad (2.63)$$

$$\alpha_2 = \frac{1-n}{n} \sqrt{\frac{2n+1}{2n-3}} \sqrt{\frac{N^2 - (n-1)^2}{N^2 - n^2}} \quad (2.64)$$

The initial conditions for the recurrence relation given in (2.60) is given as

$$\tilde{t}_0(x) = \frac{1}{\sqrt{(N)}} \quad (2.65)$$

$$\tilde{t}_1(x) = (2x + 1 - N) \sqrt{\frac{3}{N(N^2 - 1)}} \quad (2.66)$$

Tchebichef moments are selected as feature vectors in this thesis due to the following desirable properties

1. Due to its discrete and orthogonal nature, these moments can be directly applied to digital images without incurring any discretization errors and co-ordinate mapping.
2. The feature vectors of Tchebichef moments are efficient image descriptors which can represent the information present in an image without any redundancy and can detect any small variation in the pixel intensities.
3. It can effectively extract the statistical and structural content of the image and hence these features can be widely used in various applications of image processing.

Tchebichef moments in (2.59) can be represented in matrix form which is easily implemented in MATLAB. The Tchebichef moments for an image $\mathbf{X} = f(x, y)|_{x,y=0}^{N-1}$ up to the order $(p + q)$ can be implemented as

$$\mathbf{T} = \mathbf{P}\mathbf{X}\mathbf{Q}^T \quad (2.67)$$

where \mathbf{P} and \mathbf{Q} are the Tchebichef polynomials defined in matrix form as

$$\mathbf{P} = \begin{bmatrix} K_0(0) & \dots & K_0(N-1) \\ \vdots & & \vdots \\ \vdots & & \vdots \\ K_n(0) & \dots & K_n(N-1) \end{bmatrix} \quad (2.68)$$

$$\mathbf{Q} = \begin{bmatrix} K_0(0) & \dots & K_0(N-1) \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ K_m(0) & \dots & K_m(N-1) \end{bmatrix} \quad (2.69)$$

Tchebichef moments of an image \mathbf{X} are basically the projections of the image on to the basis function ϕ_{pq} , which is the product of \tilde{t}_p and \tilde{t}_q defined as

$$\tilde{\mathbf{t}}_p = [\tilde{t}_p(0)\tilde{t}_p(1)\dots\tilde{t}_p(N-1)] \quad (2.70)$$

$$\tilde{\mathbf{t}}_q = [\tilde{t}_q(0)\tilde{t}_q(1)\dots\tilde{t}_q(N-1)] \quad (2.71)$$

$$\phi_{pq} = [\tilde{\mathbf{t}}_p]^T \tilde{\mathbf{t}}_q \quad (2.72)$$

Figure 2.18 shows the plots of the basis function of Tchebichef moments.

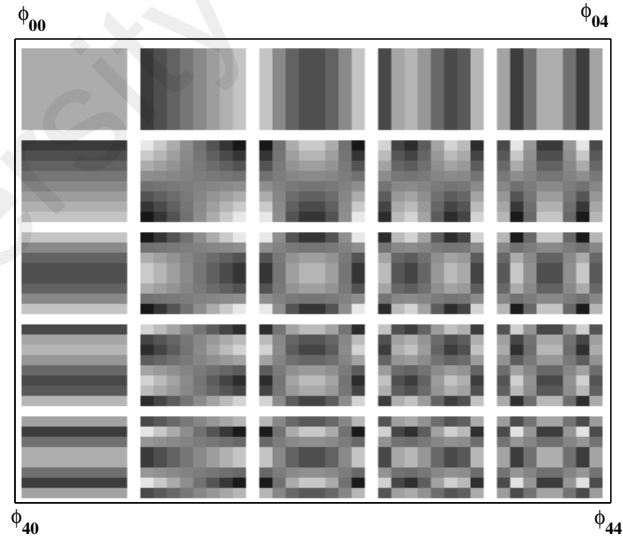


Figure 2.18: Basis function of Tchebichef moments

It can be observed from these basis function plot that for ϕ_{pq} , $\forall p = 0; q = 0, 1, 2, 3, 4$ consist of mostly the vertical edges while for ϕ_{pq} , $\forall p = 0, 1, 2, 3, 4; q = 0$ are mostly horizontal edges. If these basis functions are applied to images of varying edge contents, it will certainly extract vertical and horizontal edge information present. This information

can be used as edge features which will be later in this thesis along with learning machines is used to estimate the Gaussian blur present in an image.

2.4 Conclusion

In this chapter, an overview of the use of Geometric and Tchebichef moments in the field of image analysis is presented. In particular, their computational aspects are discussed. Here, the details of the two different 2D digital filter structures for implementing geometric moments is discussed. All the structures (Wong and Hatamian) presented will generate the same values of geometric moments, however the novelty of the structure lie in the fact that how fast the execution of moments will be performed. Further, two methods used for image reconstruction from its geometric moments which are used later in the study are also explored.

CHAPTER 3 : RESTORATION OF BLURRED IMAGES USING GEOMETRIC MOMENTS

In this chapter a mathematical framework for restoring the degraded image is proposed in moment domain for both the Gaussian and the motion blurred images. This is achieved using the relationship of geometric moments for original and blurred images along with Euler-Lagrange identity and variational framework. The proposed method is formulated as a Partial Differential Equations (PDEs) where the solution to the global minimum i.e restored image is achieved using iterative computation. In order to perform a fair comparison with the existing methods, same images used in the existing methods are taken for comparison. Experimental results are conducted to validate the proposed theoretical framework using various quality assessment methods such as SSIM, BRISQUE and VIF. It has been observed that the proposed method gives competitive results in terms of quality score when compared to the existing methods.

3.1 Introduction

Images are produced to record or display useful information. However, due to imperfections in the imaging and capturing processes, the observed image invariably represents a degraded version of the original image. To recover these imperfections is crucial in various image processing tasks. There are various types of degradations affecting the image such as noise, illumination, color contrast and blur. Here the focus is primarily on degradations due to the Gaussian blur. Typically, the degradation model of an image is given as

$$g(x,y) = h(x,y) * f(x,y) + n(x,y) \quad (3.1)$$

where, $f(x,y)$ is the true image that one would like to recover from the degraded image $g(x,y)$. Here, $*$ denotes the convolution operator, $h(x,y)$ is the Point Spread Functions

(PSFs) of the imaging system, and $n(x,y)$ is the additive noise. This degradation model can also be represented in matrix-vector form as

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n} \quad (3.2)$$

For a $N \times M$ true image $f(x,y)$, \mathbf{f} , \mathbf{g} and \mathbf{n} are the vectors of size $NM \times 1$ and \mathbf{H} is the two-dimensional degradation matrix of size $NM \times NM$.

The research involved in deblurring of an image can be classified as blind and non-blind problems. In the case of non-blind, the PSF $h(x,y)$ is assumed to be known. However, in case of blind deblurring, both of the original image and the PSF are unknown. Despite of the narrower applicability of non-blind deblurring approach, it is already a challenging problem as the convolution operators of interest are typically ill-posed. As a result of this lot of research has been still going on (J. Chen, Dong, Feng, Xu, & Li, 2013; M. Almeida & Figueiredo, 2013; Tao, Dong, Feng, Xu, & Li, 2013; Tang, Gong, Li, & Wang, 2014; Vera, Vega, Molina, & Katsaggelos, 2013b; Stoker, Wedd, Lavelle, & van der Laan, 2013). A typical application of image deblurring lies in various areas of astronomy, optics and surveillance.

In many scientific applications, the PSF is known. For example, in computational photography systems (Fergus, Singh, Hertzmann, Roweis, & Freeman, 2006), the PSF is known up to a scale. Also the PSF due to the camera motion can be effectively estimated from a single image, a sensor image or through an accelerometer (Ben-Ezra & Nayar, 2003; Yuan, Sun, Quan, & Shum, 2007). Estimation of the PSF due to one-dimensional motion, affine transformation can be estimated automatically or through iterations (Raskar, Agrawal, & Tumblin, 2006; Levin, 2006; Jia, 2007).

To solve blind deblurring problem, implementation in various domains has been proposed. One of the solutions to restore the original image in the spatial domain is by using the iterative regularization algorithm proposed based on the use of Bregman distances

(Osher, Burger, Goldfarb, Xu, & Yin, 2005). Recently, a normalized regularization function L_1 - L_2 has been proposed in (Krishnan, Tay, & Fergus, 2011). The L_1 - L_2 function is scale invariant and the gradient information obtained can be used to provide a tractable optimization algorithm. However, due to the non-convex nature of the L_1 - L_2 function, multiple local minima are produced from this process. Further, a novel image restoration process using principal components analysis (PCA) can be applied to multi-channel and single-channel deconvolution cases as introduced by Li et al. in (D.Li & Simske, 2007). This method was later modified by Nakamura et al. in (Nakamura, Mitsukura, & Hamada, 2013) to improve the quality of the restoration process in single-channel cases. The majority of the solutions in blind deblurring estimate the blur operator before restoring the image using a non-blind restoration process (Schelten, Nowozin, Jancsary, Rother, & Roth, 2015). Here, using Regression Tree Fields (RTFSs), an alternate process between estimating the blur operator and discriminative deblurring is used for blind deblurring. In this method, the errors produced by the estimation process are learned during training and the errors are compensated for in the image restoration method. Xin (X. Li, 2011) used projection based image deblurring which helped to achieve an improved tradeoff between image recovery and noise suppression. In (Bioucas-Dias, Figueiredo, & Oliveira, 2006), a new TV-based image deblurring algorithm under the assumptions of linear observations is proposed which used a majorization-minimization approach to solve image deblurring problem which replaces a difficult optimization problem by a sequence of simpler ones.

Masschaele et al. (Masschaele et al., 2005) developed an iterative, accelerated, damped algorithm to enhance the Neutron Computed Tomography images by iterative deblurring of neutron transmission projections. Gou et.al (Guo, Lee, & Teo, 1997) used iterative Constrained Least Squares algorithm for both blur identification and image restoration in blind restoration of images degraded by space-variant blurs. They extended the Expectation Maximizations (EMs) algorithm and combined it with the region adaptive

technique to handle the problem of identifying the spatially variant blurs. A variational approach in a Bayesian framework using the expectation maximization (EM) is successfully applied to blind restoration problems (Likas & Galatsanos, 2004). A combination of Haar wavelet and Lucy Richardson algorithm has been used to restore Positron Emission Tomographs (PETs) images (Tam, Lee, Hu, Liu, & Chen, 2011).

Ziya et al. (Telatar, 2005) introduces a novel method to enhance the quality of blurred images. The proposed method is based on the estimation of the multi-criteria information about degraded images. The multi-criteria information is extracted from the distribution model. Langer et al. (Langer, Osher, & Schönlieb, 2013) discusses a fast computational algorithm to solve domain decomposition for total variation minimization. An accelerated computation of the sub-problems is achieved by nested Bregman iterations. Further, a Bregmanized Operator Splitting- Split Bregmans (BOS-SBs) algorithm is proposed, which enforces the restriction onto the respective sub-domain by a Bregman iteration that is subsequently solved by a Split Bregman strategy. Xian et al. (Xiang, Meng, Wang, Pan, & Zhang, 2012) presents a supervised learning algorithm for image deblurring using the conceptual framework of matrix regression and gradient evolution. For a pair of blurred image patches and their corresponding sharp images, an optimization framework of matrix regression is formulated to learn matrix mapping. Pang et al. (Pang & Yang, 2011) proposed a projected gradient algorithm for image restoration and texture extraction based on the augmented Lagrangian strategy. It is based on the basis of a mixed model which combines the Rudin Osher Fatemi (ROFs) model with the Lysaker-Lundevold-Tai (LLT) model to reduce the staircase effect and blur phenomenon. Su et al. (M. Su & Basu, 2002) proposed a 3-stage hybrid learning system with unsupervised learning as an application for image deblurring. Also Kennedy et al. (Kennedy & Basu, 2000) used the Projection pursuit learning networks Projection Pursuit Learning Networks (PPLNs) to be used in the field of image processing. They showed how

PPLNs are effectively used to connect edges and produce continuous boundaries. Further, an application of PPLN to deblurring a degraded image when little or no a priori information about the blur is available is also demonstrated. Other solutions to deblurring in the spatial domain can be found in (Katsaggelos, 1989; Campisi & Egiazarian, 2007; Money & Kang, 2008; Ceccarelli, 2007; Shi & Yang, 2010). Vairy et al. (Vairy & Venkatesh, 1995) propose a multi-scale inversion method based on wavelet arrays. It is applicable to a wide class of images, and show that the inversion is stable with respect to noise both in the blurred signal and in the blur variance. Dobrosotskaya et al. (Dobrosotskaya & Bertozzi, 2008) proposed a new variational method for blind deconvolution of images utilizing PDE-based techniques involving the Ginzburg-Landau functional along with wavelet-based methods. Other restoration methods that utilized wavelet domain for image deblurring are reported in (Figueiredo & Nowak, 2003; Bioucas-Dias et al., 2006; Duijster, Scheunders, & De Backer, 2009).

In image restoration an important measure to be considered is the quality of the restored image. In deciding the quality of an image, subjective human perception has been, and still is, considered the benchmark of quality. This is because the end user (the human being) and his perception of quality are deemed critical. Based on the characteristics of the human visual system (HVS), a number of subjective metrics have been introduced (Z. Wang, Bovik, Sheikh, & Simoncelli, 2004a; Xue, Zhang, Mou, & Bovik, 2014). There are three categories in image quality assessment: Full References (FRs), Reduced-References (RRs) and No-References (NRs) which are explained as follows:

1. Full-Reference (FR): This metric requires the availability of the original images (L. Zhang, Zhang, & Mou, 2011; Sheikh & Bovik, 2006). These images are assumed to be of perfect quality. The distorted images are compared with the reference images to construct an objective metric.

2. Reduced Reference (RR): It provides score to the distorted images when only partial information of the original image is available (L. Ma, Li, Zhang, & Ngan, 2011).
3. No-Reference (NR): This metric provides score to distorted image when the users do not have access to the original image (Mittal, Moorthy, & Bovik, 2012; Sheikh, Bovik, & Cormack, 2005).

These methods compute an absolute value based on the given image rather than a metric that is relative to the reference image. This scenario is arguably the most common and therefore a lot of research attention has been on the production of metrics that can be used in such situations. Here, one can use both NR and FR quality metrics to justify the effectiveness of the proposed approach.

The role of moments in image processing constitute an important feature extraction method which generates highly discriminative features, able to capture the particular characteristics of the described pattern, which distinguish it among similar or totally different objects. Work has also been done for the moment extraction through VLSI structures (Hatamian, 1986; Kumar & Paramesran, 2014). Recently, their use in the field of medical imaging is also explored (Maidment et al., 2012; Meyer-Baese & Schmid, 2014; Hosny et al., 2013).

Here, a blind restoration technique for blurred images in the moment domain is proposed. The study is based on the alternate minimization principle which applies to restoration techniques in spatial domain (Chan & Wong, 1998; Sroubek & Flusser, 2003). Geometric moments are projections of the image intensity function $f(x, y)$, onto monomials $x^p y^q$. The basis set $x^p y^q$ is not orthogonal. This non-orthogonality causes information redundancy in the extracted moments and it is more difficult to reconstruct images from the moments (Teague, 1980). However, the problem of image reconstruction has been solved recently (Honarvar et al., 2014). This provides motivation and new opportunities

to explore the potential of using geometric moments in the area of image restoration.

The rest of the chapter is organized as follows. Section 3.2 gives theoretical and experimental results for the image deblurring performed using non-blind algorithm. The applicability of the proposed method in restoring the blurred image degraded due to motion blur is studied in Section 3.3. Here, the mathematical framework is formulated in moment domain for the motion blur case. Moreover, comparative analysis is performed with the recently introduced methods (Deshpande & Patnaik, 2014; Dash & Majhi, 2014). Section 3.4 describes the details of blind algorithm for image deblurring. Section 3.5 concludes the chapter.

3.2 Non-Blind Deblurring in Moment Domain

In this section, a deblurring algorithm in moment domain that uses Euler Lagrange and variational technique is discussed. A blurred image possesses higher energy and therefore seeks out a lower energy state. Thus, the goal of the variational approach is to construct an energy that describes the quality of the image and then minimize that energy (Chan & Shen, 2005). For this a well established theory of partial differential equations (PDE) has been used. PDE approach treats an image as a function of space and time which evolves gradually. Finally, an estimation of the original image can be obtained by iterating the PDE for a fixed number of iterations.

Hence, one can begin by using an established relationship between geometric moments for original and blurred image. Here, the formulation has been discussed for 1D and then extended to 2D. For a 1D N -length signal, $f(x)$ the geometric moment defined in (2.1) is given here as

$$m_p^{(f)} = \sum_{x=1}^N x^p f(x) \quad (3.3)$$

and for a 2D images as

$$m_{p,q}^{(f)} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} x^p y^q f(x,y) \quad (3.4)$$

where $m_p^{(f)}$ is the geometric moment of signal $f(x)$ of order p while $m_{p,q}^{(f)}$ is the geometric moment of an image $f(x,y)$ of order p, q .

As shown in (Flusser et al., 2009), the relationship between the degraded signal moments with original signal moments in 1D is given by

$$m_p^{(g)} = \sum_{i=0}^p \binom{p}{i} m_i^{(f)} m_{p-i}^{(h)} \quad (3.5)$$

where $m_p^{(g)}$, $m_p^{(f)}$, $m_p^{(h)}$ are the moments of the degraded signal, original signal, PSF respectively and p is the order of moment. Using (3.5), one can now write the energy function, E_p for a specific order, p as

$$E_p = \left(\sum_{i=0}^p \binom{p}{i} m_i^{(\hat{f})} m_{p-i}^{(h)} - m_p^{(g)} \right)^2 \quad (3.6)$$

where $m_p^{(\hat{f})}$ is the estimated original signal. Taking derivative of (3.6) with respect to $m_p^{(\hat{f})}$, one can obtain

$$\frac{\partial E_p}{\partial m_p^{(\hat{f})}} = 2m_0^{(h)} \left(\sum_{i=0}^p \binom{p}{i} m_i^{(\hat{f})} m_{p-i}^{(h)} - m_p^{(g)} \right) \quad (3.7)$$

Using Euler Lagrange identity, the partial differential equation (PDE) can be modeled as

$$\frac{\partial m_p^{(\hat{f})}}{\partial t} = - \frac{\partial E_p}{\partial m_p^{(\hat{f})}} \quad (3.8)$$

This variational procedure consists of iteratively updating the PDE given in (3.8) which is consistent with gradient descent approach of E_p . Substituting (3.7) in (3.8) yields,

$$\frac{\partial m_p^{(\hat{f})}}{\partial t} = -2m_0^{(h)} \left(\sum_{i=0}^p \binom{p}{i} m_i^{(\hat{f})} m_{p-i}^{(h)} - m_p^{(g)} \right) \quad (3.9)$$

Finally, by discretization of (3.9), one can get

$$m_p^{(\hat{f})}[n+1] = m_p^{(\hat{f})}[n] - 2m_0^{(h)} \left(\sum_{i=0}^p \binom{p}{i} m_i^{(\hat{f})}[n] m_{p-i}^{(h)} - m_p^{(g)} \right) \quad (3.10)$$

where n is the number of iterations performed.

Similarly for 2D image, (3.10) can be generalized as

$$m_{p,q}^{(\hat{f})}[n+1] = m_{p,q}^{(\hat{f})}[n] - 2m_{0,0}^{(h)} \left(\sum_{k=0}^p \sum_{j=0}^q \binom{p}{k} \binom{q}{j} m_{k,j}^{(\hat{f})}[n] m_{p-k,q-j}^{(h)} - m_{p,q}^{(g)} \right) \quad (3.11)$$

where p and q are the order of moments.

For a fixed number of iterations or when the error criterion is met, (3.11) is iterated.

However, the proposed method needs an additional task of reconstructing the image from $m_{p,q}^{(\hat{f})}$, which can be performed using subtractor technique discussed in Chapter 2 Section 2.2.5 (Honarvar et al., 2014).

3.2.1 Experimental Results

In this section, the validity of the proposed method for 1D signals is shown followed by a comparative analysis with spatial domain. Lastly, experiments have been performed on real astronomical images and their perceptual quality has been evaluated through the use of recently introduced quality metric (Mittal et al., 2012) .

3.2.1.1 Deblurring of 1D Signals

In order to verify the mathematical formulation provided in (3.10), an example of a 1D signal given as $f(x) = \{2, 1, 4, 3, 5, 1\}$ and the PSF as $h(x) = \{0.2, 0.2, 0.2, 0.2, 0.2\}$ is taken. The moments of the original image, PSF and blurred image are obtained using (3.3) as

$$m_p^{(f)} = \{16, 43, 149, 559, 2213, 9103\} \quad (3.12)$$

$$m_p^{(h)} = \{1, 2, 6, 20, 70.8\} \quad (3.13)$$

$$m_p^{(g)} = \{16, 75, 417, 25470, 16622, 113960\} \quad (3.14)$$

where p varies from 0 to 5.

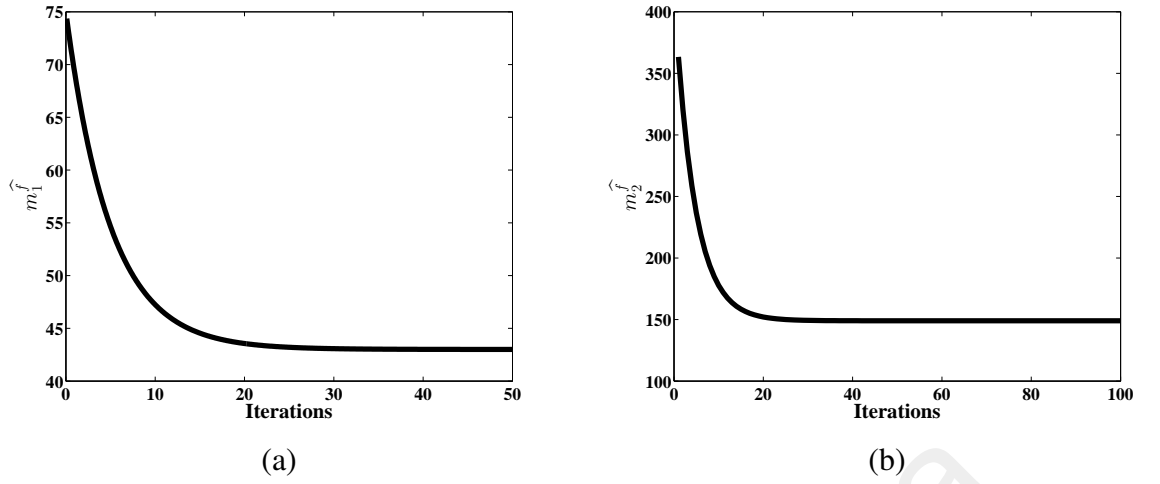


Figure 3.1: (a) Estimated $m_1^{(f)}$ and (b) Estimated $m_2^{(f)}$

By making use of observed blurred image moments, $m_p^{(g)}$ (3.14) and the moments of the PSF, $m_p^{(h)}$ (3.13), estimated original image moments, $m_p^{(f)}$ can be calculated using (3.10). From Figure 3.1 it can be observed that after 25 and 20 iterations the estimated value of $m_1^{(f)}$ and $m_2^{(f)}$ comes close to the original image moment 43 and 149 as calculated in (3.12) for $p = 1, 2$. Hence, the simulation results confirm the validity of the proposed approach in (3.10).

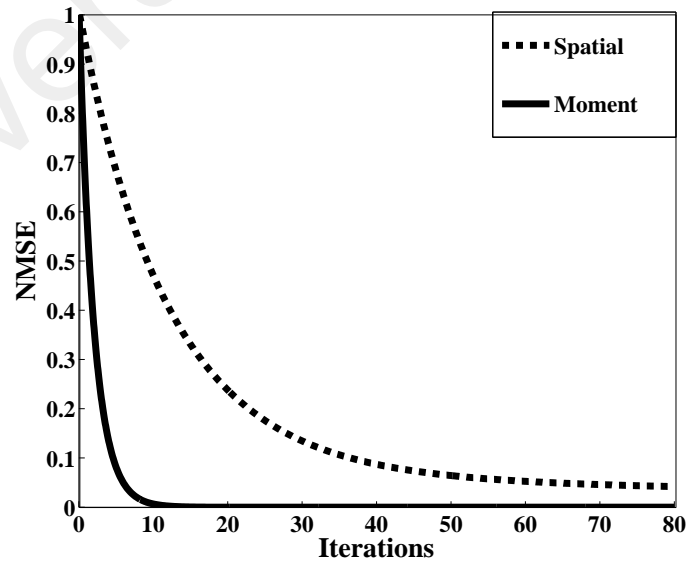


Figure 3.2: Error vs Iterations

3.2.1.2 Deblurring of 2D Images

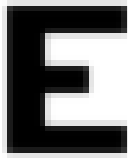



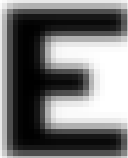

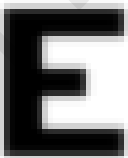
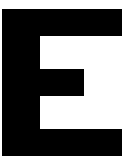
Further carrying the discussion to 2-D image processing, a comparative performance of the proposed work in moment domain with spatial domain is carried out. In this experiment, the original image was blurred using Gaussian PSF. Here, the image size is 32×32 and $p, q = 32$. First, a comparison is performed in terms of how fast the error in spatial domain $|\hat{f}(x, y) - f(x, y)|$ and $|m_{p,q}^{(\hat{f})} - m_{p,q}^{(f)}|$ in moments domain converges. From Figure 3.2 it can be observed that error drops rapidly in the case of the moment domain when compared to spatial domain for a fixed number of iterations. In this work, the number of iterations performed is 80.

3.2.1.3 Comparative Analysis

Comparative study of the image obtained through both the domains is carried out. In this work image quality assessment is used to predict perceptual image quality scores without access to reference images. One such technique is known as BRISQUE (Mittal et al., 2012) which is used to quantify the quality of an image and gives the objective score. The lower score indicates a higher quality image. Table.3.1 shows the 32×32 binary image of letter E and its blurred images. Using this original image, two sets of blurred images were created. In both the cases, the σ and mask size w are ($\sigma = 0.5$, $w = 5 \times 5$) and ($\sigma = 1.167$, $w = 7 \times 7$). By applying BRISQUE to each set of blurred images it can be observed that the objective score obtained using moment domain is less than the spatial domain. Hence, it shows that the perceptual quality of an image is better in the case of moment domain. In the next experiment astronomical images of size 64×64 and 128×128 is used. Two data sets of blurred images are created with ($\sigma = 0.833$, $w = 5 \times 5$), ($\sigma = 2.167$, $w = 13 \times 13$) for 64×64 image and ($\sigma = 1.667$, $w = 7 \times 7$), ($\sigma = 3.167$, $w = 19 \times 19$) for 128×128 . Values of mask size(w) and sigma (σ) are selected on the basis of (<http://classes.soe.ucsc.edu/ee264/Winter09/SECURE/5->

LocalOperations.pdf”, n.d.). Table. 3.2 shows the results of the BRISQUE score for both the spatial and moment domain. It can be seen that the BRISQUE score in case of moment domain is less than the spatial domain. Further, Table. 3.3 shows the comparative analysis with the existing methods. It can be observed that the proposed method provides good quality images as evaluated in terms of the BRISQUE scores.

Table 3.1: Image deblurring using spatial and moment domain for a binary image of 32×32 with different Gaussian kernel, σ and mask size, w with their corresponding BRISQUE

Blur Image	Deblurred Images		Original Image
	Spatial Domain	Moment Domain	
			
$\sigma = 0.5, w = 5 \times 5$			
BRISQUE	16.45	13.21	
			
$\sigma_1 = 1.167, w = 7 \times 7$			
BRISQUE	18.23	15.41	

3.3 Restoration of motion blurred images degraded by motion blur

In this section, the applicability of the proposed algorithm for restoring the images degraded using motion blur is explored. Mathematically, a uniform motion blur is characterized as

$$h(x, y) = \begin{cases} \frac{1}{L}, & \text{if } \sqrt{x^2 + y^2} \leq \frac{L}{2}, \frac{y}{x} = \tan \theta. \\ 0, & \text{otherwise.} \end{cases} \quad (3.15)$$

where L and θ are the length and angle parameters of the motion blur PSF.

In order to restore the blurred image, an estimation of the original moments $m_{p,q}^{(f)}$ is needed. This can be done with the use of (2.21) provided an approximate modeling of the $m_{p,q}^{(h)}$ is known. Here, we propose a mathematical model of $m_{p,q}^{(h)}$ for motion blur as

Table 3.2: Image deblurring using spatial and moment domain approaches for test images of various sizes with different Gaussian kernel, σ and mask size, w and their corresponding BRISQUE.



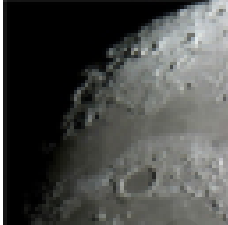



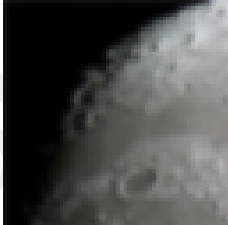
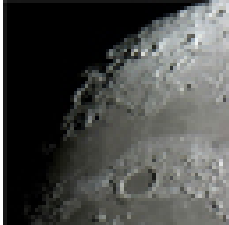












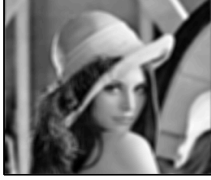

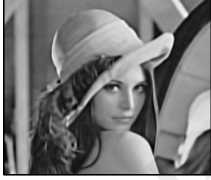

Blur Image	Deblurred Images		Original Image
	Spatial domain	Moment domain	
			
$\sigma = 0.833, w = 5$ BRISQUE	15.34	12.21	Size=64 × 64
			
$\sigma = 2.167, w = 13$ BRISQUE	18.76	16.37	Size=64 × 64
			
$\sigma = 1.667, w = 7$ BRISQUE	16.35	14.27	Size = 128 × 128
			
$\sigma = 3.167, w = 19$ BRISQUE	19.45	17.67	Size = 128 × 128

Table 3.3: Comparative analysis of the different methods evaluated in terms of BRISQUE score

Blurred Image	(Tao et al., 2013)	Deblurred Image (J. Chen et al., 2013)	Proposed
			
BRISQUE	14.56	17.23	9.12
			
BRISQUE	15.47	16.21	10.13

follows: The PSF of the motion blurred image is given as (Stern, Kruchakov, Yoavi, & Kopeika, 2002)

$$h(x, y) = \frac{1}{t_a} \int_0^{t_a} \delta(x - v_x(t), y - v_y(t)) dt \quad (3.16)$$

Here $v_x(t)$ and $v_y(t)$ are the velocities in x and y direction and t_a is the aperture time. The geometric moments of the PSF can be formulated using (3.16) as

$$\begin{aligned} m_{p,q}^{(h)} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q h(x, y) dx dy \\ &= \frac{1}{t_a} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q \int_0^{t_a} \delta(x - v_x(t), y - v_y(t)) dt dx dy \\ &= \frac{1}{t_a} \int_0^{t_a} v_x^p(t) v_y^q(t) dt \end{aligned} \quad (3.17)$$

Substituting $v_x(t) = v(t) \cos\theta$ and $v_y(t) = v(t) \sin\theta$, (3.17) becomes

$$m_{p,q}^{(h)} = \left(\frac{\cos^p \theta \sin^q \theta}{t_a} \right) \int_0^{t_a} v^p(t) v^q(t) dt \quad (3.18)$$

By assuming constant velocity $v(t)=v$, (3.18) can be approximated as

$$m_{p,q}^{(h)} = v^{(p+q)} \cos^p \theta \sin^q \theta \quad (3.19)$$

(3.19) is the proposed modeling of the PSF in moment domain represented as $m_{pq}^{(h)}$. Substituting it in (2.21) results in

$$m_{p,q}^{(g)} = v^{(p+q)} (\cos^p \theta \sin^q \theta) \sum_{k=0}^p \sum_{l=0}^q \binom{p}{k} \binom{q}{l} m_{p-k,q-l}^{(f)} \quad (3.20)$$

The estimates of the original image moments $m_{p,q}^{(\hat{f})}$ can be obtained using the variational model in moment domain discussed in Sections 3.2-3.4. Using the gradient descent technique with the help of Euler-Lagrange equation, the estimates of the original image moments $m_{p,q}^{(\hat{f})}$ can be obtained by substituting (3.19) into a 2D version of (3.9) as

$$\begin{aligned} \frac{\partial m_{p,q}^{(\hat{f})}}{\partial t} = & -2v^{(p+q)} \cos^p \theta \sin^q \theta \times \\ & \left(\sum_{k=0}^p \sum_{l=0}^q \binom{p}{k} \binom{q}{l} \left(\frac{m_{kl}^{(\hat{f})}}{v^{(k+l)} \cos^k \theta \sin^l \theta} \right) - m_{p,q}^{(g)} \right) \end{aligned} \quad (3.21)$$

By discretization of (3.21), we obtain the partial differential equation (PDE) used for estimating the moments of original image as

$$\begin{aligned} m_{p,q}^{(\hat{f})}[n+1] = & m_{p,q}^{(\hat{f})}[n] - 2v^{(p+q)} \cos^p \theta \sin^q \theta \times \\ & \left(\sum_{k=0}^p \sum_{l=0}^q \binom{p}{k} \binom{q}{l} \frac{m_{kl}^{(\hat{f})}[n]}{v^{(k+l)} \cos^k \theta \sin^l \theta} - m_{p,q}^{(g)} \right) \end{aligned} \quad (3.22)$$

Once the estimates of the original moments are obtained using (3.22), the image can be reconstructed from its geometric moments using Stirling numbers as discussed in Section 2.2.5.

3.3.1 Comparative analysis on images degraded by motion blur

The restored images are compared using the existing methods proposed by (Deshpande & Patnaik, 2014) and (Dash & Majhi, 2014). For fair comparison, the images used in these papers are taken to evaluate the restoration performance of the proposed method. Table 3.4 presents the comparative analysis of the proposed work with the existing methods

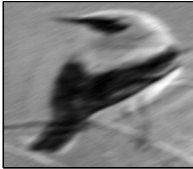
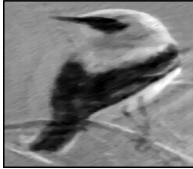











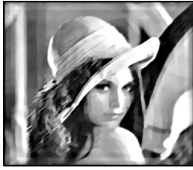






evaluated in terms of PSNR, SSIM and BRISQUE scores. Specifically, the first three figures are taken from (Deshpande & Patnaik, 2014) are blurred with $(L = 35 \text{ pixels}, \theta = 15^\circ)$, $(L = 10 \text{ pixels}, \theta = 25^\circ)$ and $(L = 25 \text{ pixels}, \theta = 10^\circ)$ respectively. It can be observed that the proposed method performs less when compared to (Deshpande & Patnaik, 2014). It can be observed that there is significant ringing at the borders of the restored image in our method. As a result the performance score in terms of SSIM, PSNR and BRISQUE degrades for our method as compared to the existing methods. But nevertheless, the overall quality of the restored image is still better. Further, the last two images of Table 3.4 are taken from (Dash & Majhi, 2014) for fair comparison and are blurred using $(L = 10 \text{ pixels}, \theta = 45^\circ)$ and $(L = 15 \text{ pixels}, \theta = 30^\circ)$ respectively. For these two images it can be observed that the proposed method provides competitive scores with the existing methods. In this case noticeable artifacts are observed in the restored image obtained through the methods by (Deshpande & Patnaik, 2014) and (Dash & Majhi, 2014). In all the experiments performed the average time elapsed in evaluating the results is around 1 minute which is slightly higher. This can be reduced to a larger extent by vectorization of the codes and use of parallel computation.

3.4 Blind deblurring in Moment domain

The deblurring of an image from the Bayesian perspective has been discussed which results into the idea of restoring the image in spatial and moment domains. Bayesian estimation provides an elegant statistical perspective to the image restoration problem. The maximum a posteriori (MAP) estimator maximizes $p(f|g)$. Using the Bayes rule, the MAP estimator can be written in terms of the conditional probability density function $p(g|f)$ and the prior probability of $p(f)$ as

$$\hat{f} = \underset{f}{\operatorname{argmax}} \frac{p(g|f)p(f)}{p(g)} = \underset{f}{\operatorname{argmax}} p(g|f)p(f) \quad (3.23)$$

Table 3.4: PSNR, SSIM and BRISQUE results of the methods on four different images with motion blurred PSF

Blurred Image	Deblurred Image (Deshpande & Patnaik, 2014) (Dash & Majhi, 2014) Proposed		
			
PSNR	28.19	26.23	20.67
SSIM	0.84	0.81	0.70
BRISQUE	17.67	18.54	25.87
			
PSNR	29.23	27.79	28.12
SSIM	0.89	0.82	0.86
BRISQUE	11.69	15.12	12.84
			
PSNR	28.89	26.51	21.79
SSIM	0.84	0.82	0.73
BRISQUE	13.43	15.73	23.12
			
PSNR	27.24	26.12	28.56
SSIM	0.84	0.81	0.88
BRISQUE	13.23	15.37	10.84
			
PSNR	27.12	28.3	29.13
SSIM	0.82	0.89	0.91
BRISQUE	15.31	14.63	11.21

where $p(g)$ has been dropped in the last term, being constant with respect to the argument f . And the last term reveals that the MAP solution indeed reduces to the maximum likelihood (ML) solution when image prior does not favor a specific solution, in other words, when $p(f)$ is of uniform distribution. In most of the image restoration problems, the error $e(x,y)$ between the blurred and original image has been modeled to be a zero-mean independent and identically distributed (iid) Gaussian random variable with probability density function given as

$$p(e(x,y)) = \frac{1}{\sqrt{2\pi}\sigma_e} \exp\left(-\frac{(e(x,y))^2}{2\sigma_e^2}\right) \quad (3.24)$$

where $e(x,y)=g(x,y) - h(x,y) * f(x,y)$. Using this relationship along with (3.24), the conditional probability density function $p(g|f)$ is written as

$$p(g|f) = \prod_{x,y} \frac{1}{(\sqrt{2\pi}\sigma_e)} \exp\left(-\frac{\|g(x,y) - h(x,y) * f(x,y)\|_2^2}{2\sigma_e^2}\right) \quad (3.25)$$

$$= \frac{1}{(\sqrt{2\pi}\sigma_e)^M} \exp\left(-\frac{1}{2\sigma_e^2} \sum_{x,y} \|g(x,y) - h(x,y) * f(x,y)\|_2^2\right) \quad (3.26)$$

(3.26) can be written in matrix form and hence the conditional probability density function $p(g|f)$ of the observed image is written as (Gunturk & Li, 2012)

$$p(g|f) = \frac{1}{(\sqrt{2\pi}\sigma_e)^M} \exp\left(-\frac{\|\mathbf{g} - \mathbf{H}\mathbf{f}\|_2^2}{2\sigma_e^2}\right) \quad (3.27)$$

where σ_e is the standard deviation of the error and M is the total number of pixels in the observed image. Substituting (3.27) into (3.23) after taking the logarithm of the argument and neglecting the irrelevant terms, one can obtain

$$\hat{f} = \operatorname{argmax}_f (\log p(g|f) + \log p(f)) = \operatorname{argmax}_f \left\{ -\frac{\|\mathbf{g} - \mathbf{H}\mathbf{f}\|_2^2}{2\sigma_e^2} + \log p(f) \right\} \quad (3.28)$$

When the prior probability density function $p(f)$ has an exponential form given as $p(f) \propto \exp(-\|\mathbf{L}f\|^2)$, where \mathbf{L} is the operator acting on the image, then

$$\hat{f} = \operatorname{argmax}_f \left\{ -\frac{\|\mathbf{g} - \mathbf{H}\mathbf{f}\|_2^2}{2\sigma_e^2} - \|\mathbf{L}f\|^2 \right\} \quad (3.29)$$

Here, the first term is known as data fidelity term and second term is known as regularization prior. The solution to (3.29) can be obtained using energy minimization technique, which can be given as

$$E(\hat{f}) = \underset{f}{\operatorname{argmin}} \left\{ \frac{\|\mathbf{g} - \mathbf{H}f\|_2^2}{2\sigma_e^2} + \|\mathbf{L}f\|^2 \right\} \quad (3.30)$$

(3.30) provides the solution to the Bayesian approach with the help of energy minimization technique. Normally, the energy minimization is performed using alternate minimization (AM) or expectation maximization (EM) algorithms (Sroubek & Flusser, 2003; Likas & Galatsanos, 2004). In this chapter AM algorithm is used for minimizing the energy (3.30) proposed in moment domain. But before this, a brief discussion of the AM algorithm is explained, followed by its implementation in spatial domain. The flowchart of the alternate minimization algorithm is shown in Figure 3.3. Here, the image deblur-

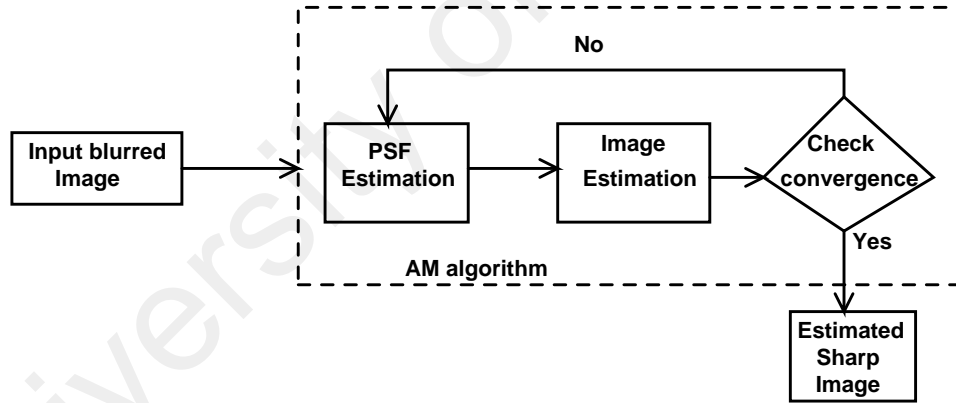


Figure 3.3: Flowchart of the alternate minimization algorithm

ring problem is split into two sub-problems, namely the PSF estimation and the image estimation. These sub-problems alternate between each other till the convergence criteria is achieved. Once the convergence is satisfied, the output of the AM algorithm is the sharp image. In this work, a blind restoration technique for blurred images in moment domain is proposed. It is based on the alternate minimization principle which is applied to restoration techniques in spatial domain (Chan & Wong, 1998; Sroubek & Flusser, 2003).

The basis sets $x^p y^q$ (3.4) in case of 2D images is not orthogonal. This non-orthogonality causes information redundancy in the extracted moments and it is more difficult to reconstruct images from the moments (Teague, 1980). However, the problem of 2D image reconstruction has been solved recently (Honarvar et al., 2014). This provides new opportunities to explore the potential of using geometric moments in the area of image restoration. Hence, it motivated to formulate a similar variational model in moment domain for solving the blind restoration problem.

Using the reference in (Chan & Wong, 1998), the energy functional E can be expressed in moment domain as follows

$$E(m_p^{(\hat{f})}) = \underset{m_p^{(\hat{f})}}{\operatorname{argmin}} \left\{ \underbrace{\left(\sum_{i=0}^p \binom{p}{i} m_i^{(\hat{f})} m_{p-i}^{(h)} - m_p^{(g)} \right)^2}_{1^{st} \text{ Term}} + \lambda \underbrace{(\nabla m_p^{(\hat{f})})^2}_{2^{nd} \text{ Term}} \right\} \quad (3.31)$$

Here, the first term is the data fidelity term and the second term is the regularization prior, $m_p^{(\hat{f})}$ is the estimated image, λ is the regularization parameter and $\nabla m_p^{(\hat{f})} = m_{p+1}^{(\hat{f})} - m_p^{(\hat{f})}$. With the help of Euler-Lagrange identity, the energy functional in (3.31) can be minimized. The Euler-Lagrange identity in moment domain can be expressed as

$$\frac{dE(m_p^{(\hat{f})})}{dm_p^{(\hat{f})}} = \left(\frac{\partial L(m_p^{(\hat{f})}, m_p^{(\hat{f})'})}{\partial m_p^{(\hat{f})}} - \nabla \left(\frac{\partial L(m_p^{(\hat{f})}, m_p^{(\hat{f})'})}{\partial m_p^{(\hat{f})'}} \right) \right) \quad (3.32)$$

where, $m_p^{(\hat{f})'} = \nabla m_p^{(\hat{f})}$ and

$$L(m_p^{(\hat{f})}, m_p^{(\hat{f})'}) = \left(\sum_{i=0}^p \binom{p}{i} m_i^{(\hat{f})} m_{p-i}^{(h)} - m_p^{(g)} \right)^2 + \lambda (\nabla m_p^{(\hat{f})})^2 \quad (3.33)$$

Evaluating (3.32) using (3.33) results in

$$\frac{dE(m_p^{(\hat{f})})}{dm_p^{(\hat{f})}} = 2 \left(m_0^{(h)} \left(\sum_{i=0}^p \binom{p}{i} m_i^{(\hat{f})} m_{p-i}^{(h)} - m_p^{(g)} \right) - \lambda (\Delta m_p^{(\hat{f})}) \right) \quad (3.34)$$

where $\Delta m_p^{(\hat{f})} = m_{p+1}^{(\hat{f})} - 2m_p^{(\hat{f})} + m_{p-1}^{(\hat{f})}$. The relationship of Euler-Lagrange with gradient descent is given as

$$\frac{\partial m_p^{(\hat{f})}(t)}{\partial t} = - \frac{dE(m_p^{(\hat{f})})}{dm_p^{(\hat{f})}} \quad (3.35)$$

Substituting (3.34) in (3.35), one can obtain

$$\frac{\partial m_p^{(\hat{f})}(t)}{\partial t} = -2 \left(m_0^{(h)} \left(\sum_{i=0}^p \binom{p}{i} m_i^{(\hat{f})} m_{p-i}^{(h)} - m_p^{(g)} \right) - \lambda (\Delta m_p^{(\hat{f})}) \right) \quad (3.36)$$

Notice the constant zero order moment ($m_0^{(h)}$) in (3.36) acts as an accelerator during the gradient descent. Also can be observed that it is a partial differential equation (PDE) which needs to be iterated with time. Numerical implementation of (3.36) results in

$$m_p^{(\hat{f})}[n+1] = m_p^{(\hat{f})}[n] - 2 \left(m_0^{(h)} \left(\sum_{i=0}^p \binom{p}{i} m_i^{(\hat{f})}[n] m_{p-i}^{(h)} - m_p^{(g)} \right) - \lambda (\Delta m_p^{(\hat{f})}[n]) \right) \quad (3.37)$$

where, n is the number of iterations performed. Using a similar argument, one can write the discrete formulation of PSF as follows: The energy functional is given as

$$E(m_p^{(\hat{h})}) = \underset{m_p^{(\hat{h})}}{\operatorname{argmin}} \left\{ \left(\sum_{i=0}^p \binom{p}{i} m_i^{(f)} m_{p-i}^{(\hat{h})} - m_p^{(g)} \right)^2 + \lambda (\nabla m_p^{(\hat{h})})^2 \right\} \quad (3.38)$$

where, $m_p^{(\hat{h})}$ is the estimated kernel moment. Using the relationship of Euler-Lagrange and gradient descent, the PDE can be modeled as

$$\frac{\partial m_p^{(\hat{h})}(t)}{\partial t} = -2 \left(m_0^{(f)} \left(\sum_{i=0}^p \binom{p}{i} m_i^{(f)} m_{p-i}^{(\hat{h})} - m_p^{(g)} \right) - \lambda (\Delta m_p^{(\hat{h})}) \right) \quad (3.39)$$

Numerical implementation of (3.39) is given as

$$m_p^{(\hat{h})}[n+1] = m_p^{(\hat{h})}[n] - 2 \left(m_0^{(f)} \left(\sum_{i=0}^p \binom{p}{i} m_i^{(\hat{h})}[n] m_i^{(f)} - m_p^{(g)} \right) - \lambda (\Delta m_p^{(\hat{h})}[n]) \right) \quad (3.40)$$

Here, $m_p^{(\hat{h})}$ is the estimated PSF moment and λ is the regularization parameter. (3.37) and (3.40) are iterated using alternate minimization algorithm (Chan & Wong, 1998). In order to verify the mathematical formulations derived for (3.37) and (3.40), an example of a 1D signal given as $f(x) = \{2, 1, 4, 2, 4\}$ and the PSF as $h(x) = \{0, 0, 1, 0, 0\}$ is taken. The geometric moments of order p are calculated for original signal, PSF and blurred signal using (3.3). The computed moment values are as follows where p varies from 0 to 4.

$$m_p^{(f)} = \{13, 31, 99, 343, 1251\} \quad (3.41)$$

$$m_p^{(h)} = \{1, 2, 4, 8, 16\} \quad (3.42)$$

$$m_p^{(g)} = \{13, 57, 275, 1413, 7571\} \quad (3.43)$$

The estimated original image moment, $m_p^{(\hat{f})}$ and PSF moments $m_p^{(\hat{h})}$, are calculated by iterating (3.37) and (3.40), respectively as shown in Figure 3.4. The values of $m_p^{(\hat{h})}$ and $m_p^{(\hat{f})}$ for 80 and 290 iterations respectively are as follows

$$m_p^{(\hat{f})} = \{12.99, 30.99, 98.98, 342.99, 1250.99\} \quad (3.44)$$

$$m_p^{(\hat{h})} = \{0.98, 1.99, 3.98, 7.99, 15.98\} \quad (3.45)$$

It can be observed that the calculated values of $m_p^{(\hat{h})}$ and $m_p^{(\hat{f})}$ are close to $m_p^{(f)}$ and $m_p^{(h)}$ respectively. The proposed approach can be extended to 2D image for orders p and

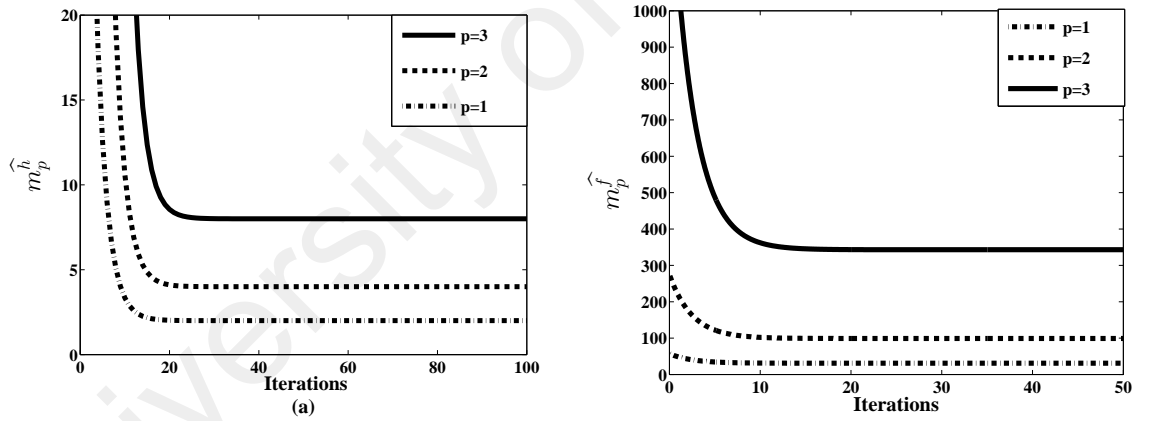


Figure 3.4: Moments for different p (a) $m_p^{(\hat{h})}$ and (b) $m_p^{(\hat{f})}$.

q as follows

$$m_{p,q}^{(\hat{h})}[n+1] = m_{p,q}^{(\hat{h})}[n] - 2 \left(m_{0,0}^{(f)} \left(\sum_{i=0}^p \sum_{j=0}^q \binom{p}{i} \binom{q}{j} m_{p-i,q-j}^{(\hat{h})}[n] m_{i,j}^{(f)} - m_{p,q}^{(g)} \right) - \lambda (\Delta m_{p,q}^{(\hat{h})}[n]) \right) \quad (3.46)$$

$$m_{p,q}^{(\hat{f})}[n+1] = m_{p,q}^{(\hat{f})}[n] - 2 \left(m_{0,0}^{(h)} \left(\sum_{i=0}^p \sum_{j=0}^q \binom{p}{i} \binom{q}{j} m_{p-i,q-j}^{(h)} m_{i,j}^{(\hat{f})}[n] - m_{p,q}^{(g)} \right) - \lambda (\Delta m_{p,q}^{(\hat{f})}[n]) \right) \quad (3.47)$$

Notice the constant zero order moment $m_{0,0}^{(f)}$ and $m_{0,0}^{(h)}$ in (3.46)-(3.47) respectively which acts as an accelerator during the gradient descent. The image can be reconstructed from $m_{p,q}^{(\hat{f})}$ (Honarvar et al., 2014) using

$$f(x,y) = \sum_{p=0}^{N-1} \sum_{q=0}^{M-1} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \frac{(-1)^{i+j-x-y}}{i!j!} \binom{i}{x} \binom{j}{y} S_1(i+1, p+1) S_1(j+1, q+1) m_{pq}^{(\hat{f})} \quad (3.48)$$

where $f(x,y)$ is the reconstructed image and $S_1(i,j)$ is the Stirling number of first kind which can be obtained recursively using

$$S_1(n+1, k) = -nS_1(n, k) + S_1(n, k-1) \quad (3.49)$$

with $S_1(0,0) = 1$, $S_1(n,0) = S_1(0,n) = 0$ and $k = 0, 1, 2, \dots, n$. The details are discussed in Chapter 2 Section 2.2.5.

The convex nature of the problem i.e. (3.31) and (3.38) ensures convergence. However, the ill-composed nature of matrix \mathbf{H} that arises in a typical inverse problem has been avoided as follows

1. The value of regularization parameter λ needs to be adjusted and selected using L-curve so that the damped least squares problem is well conditioned (Fieguth, 2010; Hansen, 2010).
2. There is a possibility that the determinant of matrix \mathbf{H} going to zero during inversion of the matrix \mathbf{H} to get the solution vector \mathbf{f} (3.2). However, the proposed method is based on an iterative scheme that converges to the final solution with time index (n) and there is no inversion of the matrix \mathbf{H} involved (Gunturk & Li, 2012).
3. The algorithms are incorporated using the multiple precision arithmetic library to increase the precision limits of the arithmetic operators so that system remains well conditioned.

3.4.1 Selection Procedure for (λ)

The regularization parameter λ is very important for obtaining a good restoration result, since it controls the trade-off between data fidelity and smoothness of the solution. If λ is too small, the small features in the regularized solution is over-smooth. On the other-hand if λ is too large, the regularized solution is under-smooth. Hence, there is a need to select λ which balances both smoothness and data fidelity in an optimal way. There are various methods for obtaining the optimal selection of the regularization parameter (λ) . Some of them are L-curve (Hansen, 1992), Discrepancy Principle (Vainikko, 1982) and Generalized Cross Validation (Golub, Heath, & Wahba, 1979). Here, the L-curve method is used. It is a log-log plot of the data fidelity against the smoothness term for a range of values of regularization parameters. To illustrate how the regularization parameter is obtained for both the moment and spatial domains, the standard test image of Lena is blurred using PSF of size 11×11 and $\sigma=1.53$. In the case of the moment domain, the data fidelity term $\|\mathbf{g} - \mathbf{H}\mathbf{f}\|_2^2$ and the smoothness term $\|\mathbf{L}\mathbf{f}\|_2^2$ are equated as

$$\|\mathbf{g} - \mathbf{H}\mathbf{f}\|_2^2 = \left(\sum_{i=0}^p \binom{p}{i} m_i^{(f)} m_{p-i}^{(\hat{h})} - m_p^{(g)} \right)^2 \quad (3.50)$$

$$\|\mathbf{L}\mathbf{f}\|_2^2 = (\nabla m_p^{(\hat{f})})^2 \quad (3.51)$$

From the L-curves shown in Figure 3.5, it can be seen how the regularization parameter λ increases and eventually settles to a value at the corners of the L-curve which balances both the data fidelity and smoothness term. The optimal values of the regularization parameters λ obtained from the L-curve as shown in Figure 3.5 for both the moment and spatial domains are 0.0045 and 0.0041 respectively.

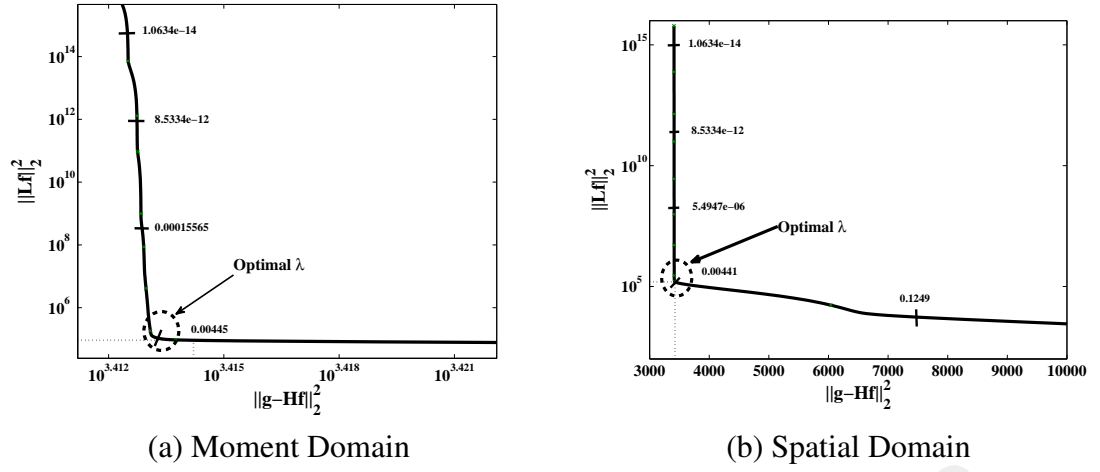


Figure 3.5: L-curve for Optimal λ selection

3.4.2 Experimental Results

To establish the theoretical analysis, various experiments have been carried out. For all the experiments access to noise free original images as shown in Figure 3.6 is there. The proposed algorithm presented here for deblurring is processed on a notebook computer with an Intel Core(TM) i7 CPU, 8GB RAM, scripted in C++ and MATLAB.



Figure 3.6: Test Images

3.4.2.1 Performance metrics

The proposed method has been evaluated using both simulated open access and real databases. The quality of the restored image has been measured using recently introduced quality metrics such as BRISQUE (Mittal et al., 2012), SSIM (Z. Wang et al., 2004a)

and PSNR. Generally, a deblurring technique is a problem of finding a reasonably good estimate $\hat{f}(x,y)$ of its blur-free version $f(x,y)$ from the corresponding degraded version $g(x,y)$. One of the common and simplest performance metric is the Mean Square Errors (MSEs), defined as

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(f(i,j) - \hat{f}(i,j) \right)^2 \quad (3.52)$$

where $N \times M$ is the size of an image.

Another most commonly used performance metric is Peak Signal to Noise Ratios (PSNRs), which is defined in decibels (dB) for 8-bit grayscale images as PSNR measures the extent to which noise has been suppressed and has been defined as

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \quad (3.53)$$

The MSE and PSNR are appealing performance metrics as they can be calculated mathematically and have clear physical meaning. But, these are not very well matched to visual quality, i.e. they do not measure the visual features of the resulting image directly. Apart from this, the images with large differences in their psycho-visual quality can have similar MSE or PSNR values. To resolve this issue BRISQUE and SSIM have also been used.

BRISQUE is a no-reference image quality assessment model based on natural scene statistic and operates in the spatial domain. It uses features from scene statistics of locally mean subtracted contrast normalized (MSCN) luminance coefficients and the pairwise products of MSCN. The features are then fed into support vector machine regressor (SVR) to quantify possible losses of naturalness in the image due to the presence of different type of distortions. The lower BRISQUE score indicates a good quality image.

The Structural SIMilarity (SSIM) index is a method for measuring the similarity between two images. The SSIM index can be viewed as a quality measure of one of the

images being compared, provided the other image is regarded as of perfect quality. SSIM for the two images x and y is defined as

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (3.54)$$

where μ_x, μ_y are the average of x and y respectively, σ_x^2, σ_y^2 are the variances of x and y , σ_{xy}^2 is the covariance of x and y while c_1, c_2 are the two variables to stabilize the division with weak denominator defined as $(k_1a)^2$ and $(k_2a)^2$ where the default value of $k_1 = 0.01$ and $k_2 = 0.03$ and the dynamic range of pixel as a , typically for 8 bits pixel is given as 255. The SSIM score is less than or equal to 1 and it is maximal when two images are coinciding. Furthermore, the full reference quality metrics FSIM (L. Zhang et al., 2011), VIF (Sheikh & Bovik, 2006) and SSIM (Z. Wang et al., 2004a) are also used to verify the proposed approach. The Structural SIMilarity (SSIM) index is a method for measuring the similarity between two images. The SSIM index can be viewed as a quality measure of one of the images being compared, provided the other image is regarded as of perfect quality. Feature-similarity (FSIM) index is an image quality assessment method based on the fact that Human Visual Systems (HVSs) understands an image mainly according to its low-level features whereas Visual Information Fidelity measure (VIF) provides the scores by combining an image information measure that quantifies the information that is present in the reference image and how much of this reference information can be extracted from the distorted image. Combining these two quantities, a score is provided. For all the FR metrics considered here, a higher score indicates a good quality image.

In this study, three experiments are carried out to demonstrate the effectiveness of the proposed method on the quality of the restored images by considering the effects of the regularization parameter and PSF size. The quality of the restored images from the proposed method is compared with the restored image obtained using the alternate

minimization method in spatial domain. In all the experiments, images shown in Figure 3.6 are used, where they are subjected to various degrees of Gaussian blur parameters.

3.4.2.2 Variation of Regularization Parameter (λ) and Quality scores using No Reference

In this section the significance of regularization parameter λ on the quality of the restored image is studied. In order to carry out the experimental work images shown in Figure 3.6 are taken into consideration. These test images are blurred using different PSF's. For example Barbara, Lena and Cameraman images are blurred using PSF's of $(\sigma = 0.76, w = 7 \times 7)$, $(\sigma = 1.53, w = 11 \times 11)$, $(\sigma = 3.26, w = 13 \times 13)$ respectively. The optimal values of λ shown in Table 3.5 for the test images of Figure 3.6 for both spatial and moment domains are obtained using the L-curve method.

Table 3.5: Optimal values of λ

Images	Spatial Domain	Moment Domain
Lena	0.0041	0.0045
Cameraman	0.0059	0.0062
Barbara	0.0071	0.0073

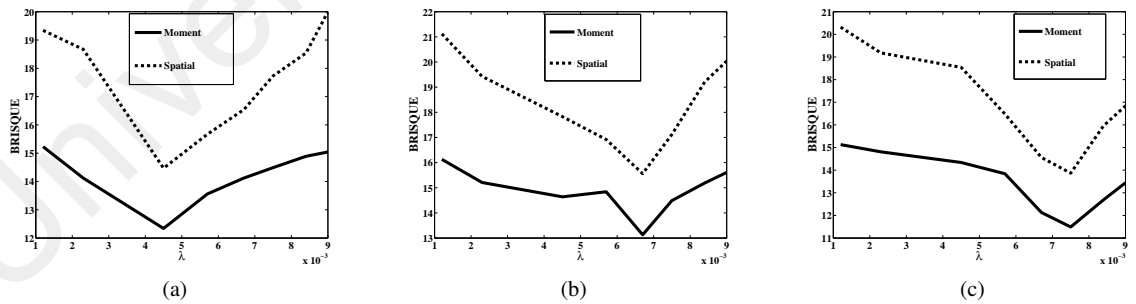


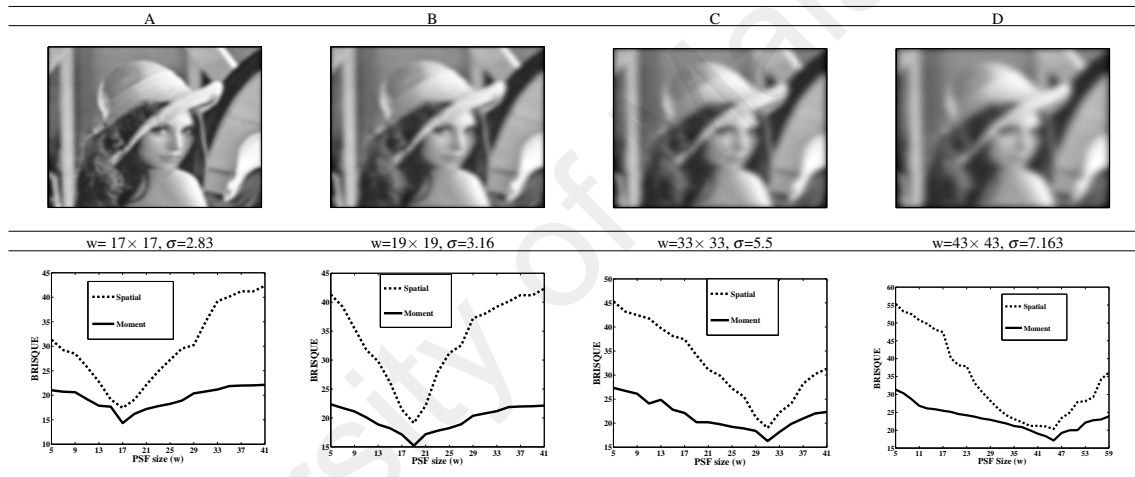
Figure 3.7: BRISQUE with regularization parameter λ

Henceforth, these optimal values of λ are used in the experimental work to determine whether they gave the best scores of the restored images for both the proposed and spatial domains. Here, the regularization parameter λ is varied from 0.0010 to 0.0009 and the effect of the regularization parameter on the BRISQUE scores is shown in Figure 3.7. It

can be seen that the best quality scores for both the spatial and moment domains for all the test images are obtained when λ is used from the L-curve as shown in Table 3.5.

Additionally, it shows the quality scores for all the three experiments using the proposed method is better than the alternate method in spatial domain for all the values of λ . In all the experimental results shown in Figure 3.7, the PSF size (w) is fixed. But in most of the applications, the PSF size is unknown. In the next section one wants to study the effects of varying PSF size on quality scores using the optimal regularization parameter as shown in Table 3.5.





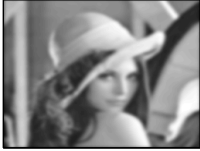






Table 3.6: BRISQUE scores variation with PSF size (w)



3.4.2.3 Effect of varying PSF Size (w) on Image Quality

In this study, the original image shown in Figure 3.6(a) is blurred using various values of Gaussian blur parameters σ and w . The columns of Table 3.6 show the blurred images with their corresponding Gaussian blur parameters. Here, the significance of the PSF size in relation to the quality scores of the restored images using BRISQUE is shown. The variation of the PSF size (w) for a fixed optimal $\lambda = 0.0045$ and the obtained BRISQUE scores of the restored images are also shown in Table 3.6. It can be seen that there is a variation in the BRISQUE scores of the restored images (L. Chen & Yap, 2003; Fahmy, Raheem, Mohamed, & Fahmy, 2012). For instance, the first column in Table 3.6, the best

Table 3.7: Image deblurring in Spatial and Moment domain

Blur	Deblurred Images		Original
Image	Spatial domain	Moment domain	Image
			
$\sigma_1 = 0.76, w = 7 \times 7$ BRISQUE	17.1421	9.97	6.71
			
$\sigma = 1.53, w = 11 \times 11$ BRISQUE	18.4590	12.0411	9.28
			
$\sigma_1 = 3.26, w = 13 \times 13$ BRISQUE	19.2305	11.35	8.23

quality scores are obtained for both the spatial and proposed method when $w=17 \times 17$. The selection of wrong PSF size results in an increase of the BRISQUE scores when compared with the exact PSF size with which the image was originally blurred. A minimum BRISQUE scores is registered for both the proposed method and spatial domain if the PSF size is the same with which the image is originally blurred. Additionally, it can be seen that the variations in the BRISQUE scores for the proposed method is lower as compared to that method used in the spatial domain.

To show the quality of the restored images, an experiment is carried with two additional images together with the Lena image as shown in Table 3.7. It can be seen that the BRISQUE scores in case of moment domain are lower than the spatial domain. This observation shows that there is an increase the quality of the restored image in case of the moment domain when compared to the image restored in spatial domain.

Table 3.8: Quality evaluation for Lena image

$\lambda \backslash w$	13	19*	23
0.0035	17.23	18.23	19.34
0.0045*	16.23	15.23	17.21
0.0055	18.21	19.23	20.47

Table 3.9: Quality evaluation for Cameraman image

$\lambda \backslash w$	17	21*	27
0.0052	17.76	19.45	21.34
0.0062*	16.23	15.31	18.72
0.0072	18.87	20.76	22.65

Table 3.10: Quality evaluation for Barbara image

$\lambda \backslash w$	13	17*	23
0.0063	17.12	18.65	19.23
0.0073*	16.42	14.65	16.43
0.0083	19.87	20.76	22.34

3.4.2.4 Effect of varying regularization parameter and PSF size

In this section, one explores the possibility of obtaining better quality scores when the regularization parameter λ is other than the optimal value used and the PSF size is not the exact one. The images shown in Figure 3.6 are used and the results are shown in Tables 3.8-3.10 for the proposed method. It can be observed that a minimum BRISQUE is registered when both λ and w are optimal as indicated by λ^* and w^* respectively. However, the BRISQUE scores are higher when either of the two parameter deviates from the optimal value. For instance, in Table 3.8 for the optimal values of the parameters $w=19$ and $\lambda=0.0045$, a minimum BRISQUE score of 15.23 is registered. However, the value of BRISQUE scores are higher if the parameters deviate from the optimal values. Similar observations are seen from Tables 3.9-3.10 respectively. Hence, this experiment shows that proper selection of parameters λ and w are essential for determining the quality of the restored image.

3.4.2.5 Quality Analysis using Full Reference

In the previous section, a no-reference quality metric was used to evaluate the restored images. However, in order to prove the robustness of the proposed approach, in this section the quality of the restored images is evaluated using Full Reference quality metrics. This experimental work is similar to the one shown in Figure 3.7 with a difference that now the quality scores are evaluated using FSIM (L. Zhang et al., 2011) and VIF (Sheikh & Bovik, 2006). It is performed to observe whether FSIM and VIF also provide the best quality scores at optimal λ or not.

Table 3.11: Full-Reference quality evaluation for Lena image

Regularization Parameter (λ)	VIF		FSIM	
	Spatial domain	Moment domain	Spatial domain	Moment domain
0.0025	0.7912	0.8931	0.7943	0.9131
0.0035	0.8216	0.9112	0.8146	0.9212
0.0045*	0.8521	0.9412	0.8372	0.9361
0.0055	0.8148	0.9381	0.8281	0.9321
0.0065	0.7859	0.9246	0.8164	0.9012

Table 3.12: Full-Reference quality evaluation for Cameraman image

Regularization Parameter (λ)	VIF		FSIM	
	Spatial domain	Moment domain	Spatial domain	Moment domain
0.0042	0.8271	0.9012	0.8371	0.8862
0.0052	0.8532	0.9276	0.8451	0.9021
0.0062*	0.8681	0.9331	0.8546	0.9152
0.0072	0.8273	0.9123	0.8363	0.8991
0.0082	0.8043	0.8921	0.8217	0.8831

Table 3.13: Full-Reference quality evaluation for Barbara image

Regularization Parameter (λ)	VIF		FSIM	
	Spatial domain	Moment domain	Spatial domain	Moment domain
0.0053	0.8187	0.9064	0.7981	0.9089
0.0063	0.8354	0.9217	0.8148	0.9151
0.0073*	0.8413	0.9318	0.8243	0.9298
0.0083	0.8371	0.9112	0.8171	0.9189
0.0093	0.8065	0.8949	0.8041	0.8971

Tables 3.11-3.13 provide the variations of FSIM and VIF quality scores for a range of λ values that include the optimal value λ^* for all the images shown in Figure 3.6. For instance, the optimal regularization parameter $\lambda=0.0045$ shown in Table 3.11, one can obtain the best VIF and FSIM scores for both the spatial and moment domains as highlighted in boldface. The spatial and moment domain registered quality scores (VIF, FSIM) as (0.8521, 0.9412) and (0.8372, 0.9361) respectively. However, as the regularization parameter deviates from its optimal value, the quality of a restored image is degraded as indicated by its lower quality scores. Similar observations can be seen from Tables 3.12-3.13 respectively. By observing the quality scores in the Tables it can be seen that the moment domain gives better quality scores when compared to spatial domain.





















3.4.2.6 Comparative analysis performed on images degraded due to Gaussian blur PSF

The experiments are performed on four of the classical gray scale images shown in Figure 3.6 to demonstrate viability and efficiency of the proposed method. The results obtained are compared with the three state-of-the-art image deblurring methods: wavelet (Dobrosotskaya & Bertozzi, 2008), majorization-minimization (Bioucas-Dias et al., 2006) and multiplicative-iterative (J. Zhang, Zhang, & He, 2008) based methods. Here, the test images are blurred using a 7×7 Gaussian blur PSF with standard deviation 1.2 and a 9×9 average blur PSF. The blurred images obtained as a result of the Gaussian and average blur are shown in the first column of Tables 3.14 and 3.15 respectively. Comparative analysis of the methods is evaluated in terms of PSNR, SSIM and BRISQUE scores. It can be observed from Table 3.14 that the evaluated scores of the restored images using the proposed method are comparable to the state-of-the-art methods. To summarize the comparative results, it can be concluded from Tables 3.14 and 3.15 that the proposed technique works fairly well along with other methods, hence paving the way for using geometric moments in the field of image restoration.

3.5 Conclusion





















In this chapter, restoration of blurred images using geometric moments is proposed. A mathematical formulation for restoration is proposed using an Euler-Lagrange identity together with the Alternate Minimizations (AMs) algorithm set up in a variational framework in the moment domain. In the restoration of the blurred images, selection of the regularization parameter λ plays an important role in restoring the quality of the images. Therefore, the L-curve is adopted to select an optimal value of the regularization parameter. Further, the experimental results show the evidence of how the quality of an image is affected by the variation of the regularization parameter and the PSF size (w) on both the proposed and the spatial domain methods. Image quality is measured using the state-of-

Table 3.14: PSNR, SSIM and BRISQUE results of the methods on four different images with a Gaussian PSF of size 7x7 and $\sigma=1.2$

Blurred Image	Deblurred Image (Dobrosotskaya & Bertozzi, 2008) (Bioucas-Dias et al., 2006) (J. Zhang et al., 2008) Proposed			
				
PSNR	28.83	27.12	28.3	28.13
SSIM	0.87	0.82	0.86	0.89
BRISQUE	12.32	15.73	13.23	11.21
				
PSNR	27.32	27.12	29.3	29.32
SSIM	0.88	0.79	0.89	0.91
BRISQUE	13.24	14.63	12.31	12.21
				
PSNR	26.12	25.23	27.13	27.14
SSIM	0.82	0.76	0.85	0.84
BRISQUE	13.32	14.42	13.23	13.13
				
PSNR	27.89	26.65	27.93	27.32
SSIM	0.86	0.84	0.85	0.88
BRISQUE	13.86	14.19	13.63	13.84

the-art quality metrics like BRISQUE, Feature Similarity (FSIMs), Visual Information Fidelity (VIFs) and SSIMs. It has been observed that the variation of the image quality scores with respect to λ and w is less in the case of the proposed method when compared to the spatial domain method. This is justified in terms of the image quality scores. Also the proposed method exhibits good restored image quality when compared to the existing method. Further, the applicability of the proposed method is explored in the case of motion blur and it has been observed that the proposed method gives satisfactory results when compared to the existing methods.

Table 3.15: PSNR, SSIM and BRISQUE results of the methods on four different images with an average PSF of size 9x9

Blurred Image	Deblurred Image (Dobrosotskaya & Bertozzi, 2008) (Bioucas-Dias et al., 2006) (J. Zhang et al., 2008) Proposed			
				
PSNR	29.3	26.12	27.5	29.86
SSIM	0.91	0.88	0.89	0.92
BRISQUE	11.73	13.73	12.65	11.76
				
PSNR	28.34	26.24	27.54	28.56
SSIM	0.87	0.79	0.82	0.89
BRISQUE	12.46	13.43	11.87	12.45
				
PSNR	28.98	27.86	27.73	28.43
SSIM	0.87	0.81	0.79	0.88
BRISQUE	12.53	13.12	13.73	12.87
				
PSNR	29.89	27.51	28.79	29.23
SSIM	0.88	0.86	0.87	0.89
BRISQUE	11.86	12.69	12.12	11.84

CHAPTER 4 : RESTORATION OF THE BLURRED IMAGES IN MOMENT DOMAIN VIA CASCADED DIGITAL FILTERS

In this chapter, the estimation of the Gaussian PSF is carried out using the Weighted Geometric moments (WGM). It works as an effective edge descriptor which together with extreme learning machine (ELM) is used for the estimation of PSF parameters. The estimation performance of the ELM using cross-database analysis performed on six different standard databases such as LIVE, CSIQ, CIDIQ, TID2008, Caltech and Berkeley. This is followed by restoration in moments domain and reconstructing the restored image using 2D cascaded digital filters operating as subtractors. To validate, the proposed method experimental work has been carried out on the aforementioned databases and results is compared with the existing methods. The results obtained using the proposed method demonstrates effectiveness of the method compared to the existing methods.

4.1 Introduction

A typical image restoration task can be casted as a linear inverse problem as

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n} \quad (4.1)$$

where $\mathbf{H} \in \mathbb{R}^{n^2 \times n^2}$ is a two dimensional blurring matrix where the elements are taken from the point spread function (PSF) $h(x, y)$, $\mathbf{f} \in \mathbb{R}^{n^2}$ is an original image of size $n \times n$, $\mathbf{g} \in \mathbb{R}^{n^2}$ is the degraded image and \mathbf{n} is usually additive Gaussian white noise. When \mathbf{H} is an identity matrix, the problem is define as image denoising; when \mathbf{H} is a diagonal matrix with elements 1 or 0, the problem is casted as image inpainting; when \mathbf{H} is a Gaussian PSF, the problem becomes image deblurring. In this chapter, focus is on problem of image deblurring. Image deblurring is an inverse problem where the objective is to recover a sharp image from its degraded version affected due to various distortions such as atmospheric turbulence, optical aberrations, sensor or motion blur etc. This deblurring

task is widely used in various areas of remote sensing, medical imaging, astronomy and surveillance systems.

With the applications in wide areas, the research on deblurring is divided into two categories, namely non-blind image deblurring, in which the PSF is assumed to be known, and more realistically the blind image deconvolution, in which both the image and PSF are unknown. The focus of this chapter is on Blind Image Deblurrings (BIDs) subjected to Gaussian PSF degradation. The problem of BID is an ill-posed problem since there are infinitely large number of solutions, i.e. pair of image and PSF estimates, which are compatible with the blurred image. A typical minimization problem to achieve the task of BID is given as

$$\underset{\mathbf{f}}{\operatorname{argmin}} \left\{ \|\mathbf{g} - \mathbf{H}\mathbf{f}\|_2^2 + \lambda \phi(\mathbf{H}) \right\} \quad (4.2)$$

where the first term is the data fidelity term, second is the regularization term denoting the PSF prior whereas λ is the regularization term. In order to obtain the reasonable approximation of the solution given in (4.1), most of the BID methods restricts the class of PSF using various types of regularizers/priors or parametric models. For instance, in (Carasso, 2003), use the parametric model in which the proposed method performs the blind restoration of the blurred images with the restricted class of shift-invariant PSF, which are expressed as finite convolution products of two-dimensional radially symmetric Levy stable probability density functions. This constraint on PSF generalizes the Gaussian and Lorentzian densities but does not include the defocus and motion blurs. Chang et. al. (M. S. Chang, Yun, & Park, 2007) restricts the PSF to the dual-exposure type and proposes an effective PSF search algorithm, by incorporating the concept that the gain of PSF has a relatively small influence over the quality of the restored image. Krylov et al. (Krylov & Nasonov, 2009) proposed the method of splitting the image into low and high frequency components. Further, use the regularization-based sharpening to low fre-

quency image component and combine the sharpened low frequency image with the high frequency image. In (Oliveira, Figueiredo, & Bioucas-Dias, 2007) the estimates of the PSF parameters are obtained using the spectral behavior of the natural images and Radon transform. Yin (Yin & Hussain, 2006) proposed a blind image restoration technique based on the Blind Signal Separations (BSSs) technique in conjugation with the genetic algorithm for PSF parameters estimation. Amizic et al. (Amizic et al., 2012) used a Total Variations (TVs) based prior for the PSF. Further, the concept of Bayesian inference is carried out for both the image and PSF priors by making use of majorization-minimization technique. Fergus et al. (Fergus et al., 2006) employs the heavy tailed distribution over gradient magnitudes along with a zero mean mixture-of-Gaussian models to estimate the PSF and the image. Krishnan et al. (Krishnan et al., 2011) proposed the use of L1/L2 norm for the image and PSF estimations. Other methods using a similar approach to PSF estimation, but using different priors can be found in (Levin, Weiss, Durand, & Freeman, 2009; Likas & Galatsanos, 2004; Shan, Jia, & Agarwala, 2008; Money & Kang, 2008). (Vera, Vega, Molina, & Katsaggelos, 2013a) proposed an algorithm for image restoration based on fusion of non-stationary and edge preserving priors. A Bayesian model is developed followed by an approximated inference approach for deriving the proposed restoration method. With a series of approximations, the implementation of the proposed restoration algorithm is iterative based and takes advantage of the Fourier domain. (Yan, Fang, & Zhong, 2012) used a blind deconvolution algorithm with spatially adaptive regularization. The information about the different region is incorporated in to the regularizer by using edge indicator. The proposed algorithm can effectively preserve the edges in the restored image along with detailed information. Further, it is robust to the change in the regularization parameter.

There are essentially two different approaches to solve blind image deblurring problem: (i) to estimate the image and the PSF simultaneously using optimization techniques

(Bar, Sochen, & Kiryati, 2004; Chan & Wong, 1998; Šroubek & Milanfar, 2012; Osher et al., 2005); (ii) perform the step of PSF estimation first and then use this PSF estimate into an existing non-blind image deblurring algorithms. Here, a PSF estimation method to be used in an approach of the type (ii) is proposed. Similar type of approach had been adopted in (Rooms, Pizurica, & Philips, 2002). Here, the authors used the wavelet domain to estimate the PSF parameters. Next, Moghaddam (Moghaddam, 2008) proposed the estimation of PSF parameters using genetic algorithms. Also Aizenberg et al. (Aizenberg et al., 2002) proposed a neural network based framework for PSF parameters estimation and used this information in restoring the degraded image using several kinds of restoration methods. Motivated by these contributions, we adopted the method of training the Extreme Learning Machines (ELMs) (Huang, Zhu, & Siew, 2006) using the images with geometric moments (GM) as features in order to estimate the PSF parameters of a Gaussian blurred images. Once the PSF is estimated, the restoration together with reconstruction is performed using geometric moments.

Unlike other domains such as Wavelet and Sparse, moment domain is well explored in the field of image processing. Specifically, their application in the field of texture classification (Albregtsen, Schulerud, & Yang, 1995), text recognition (Altuwaijri, Bayoumi, et al., 1994), watermarking (Alghoniemy & Tewfik, 2004), robot sensing (Markandey & DeFigueiredo, 1992) and content based image retrieval (Jones, Schaefer, & Zhu, 2004) is well understood and explored. geometric moments (GM) for 2D image defined in Chapter 2 (2.1) is given as

$$m_{pq} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} x^p y^q f(x, y) \quad (4.3)$$

GMs are the projection of image intensity function $f(x, y)$, onto monomials $x^p y^q$. In order to obtain the image $f(x, y)$ back from its moments m_{pq} , there is a need for doing an inverse transform. However, due to the non-orthogonality of the monomials this problem

is difficult. Another type of non-orthogonal moments are complex moment (Abu-Mostafa & Psaltis, 1985) that carry the same amount of information as geometric moments. Although orthogonal moments hold an advantage over non-orthogonal moments due to their orthogonality property, geometric moments are commonly used in image reconstruction and have proven to be the most efficient tool for image analysis (Bimbo, 1999). This is due to the fact that geometric moments not only illustrate the characteristics and features of the image, but the ability to allow the encoding of a shape offers a practical advantage as compared to orthogonal moments. Moreover, the redundancy produced from the non-orthogonality property is useful when dealing with noisy images. Despite all the practical advantages, one area of concern when dealing with geometric moments is the duration of the computational process, especially when it involves higher order geometric moments (Flusser, 1998).

One of the earliest methods to calculate the geometric moments of an $N \times N$ image is a straightforward computation method using N^2 additions and $2N^2$ multiplication process proposed by Reeves (Reeves, 1982). Hatamian in (Hatamian, 1986) uses cascaded single pole digital filter to compute 16 geometric moments of an $N \times N$ image and manage to reduce the number of additions and multiplications used in Reeves's method. To further improve the computation time, a computational process using the delta method is proposed in (Zakaria, Vroomen, Zsombor-Murray, & Van Kessel, 1987). In this method, the image is decomposed into individual rows of pixel and the image's moment is given as a sum of all the row moments. Yang and Albregtsen in (Yang & F., 1996) proposed a method based on Green theorem where they evaluate the double integral over an image by means of single integration along the image boundary. However, (Zakaria et al., 1987) and (Yang & F., 1996) are only suitable for binary images. Based on Hatamian's digital filter concept, Li in (B. Li, 1993) designed a fast computation method to compute geometric moments but the delay element in Hatamian filter design has become a major setback

to the design. In order to improve the computation complexity, Wong and Siu moved the delay element from feed-forward path to the feedback path (Wong & Siu, 1999).

A novel method was introduced by Spiliotis and Mertzios in (Spiliotis & Mertzios, 1998) where binary image representation is employed using non-overlapping rectangular homogeneous block. The geometric moments are obtained by summing up the moments of all blocks. This method is refined by Flusser in (Flusser, 2000). Moreover, the method in (Spiliotis & Mertzios, 1998) is extended in (Chung & Chen, 2005) to compute the low order geometric moments of a gray-scale image. Recently, a novel extended algorithm based on the method in (Spiliotis & Mertzios, 1998) is used to compute accurate values of the geometric moments of gray-scale and binary images (G. A. Papakostas, Karakasis, & Koulouriotis, 2008). Khalid in (Hosny, 2007) computed the exact values of geometric moments using the mathematical integration of the monomial terms over digital image pixels. This method removed the numerical approximation errors in conventional methods. Another method to minimize the numerical instability problem in higher order geometric moments is by using an appropriate transformation of image coordinates (Wee, Paramesran, & Mukundan, 2008). In (Wee et al., 2008), a novel set of geometric moments called symmetrical geometric moments (SGMs) are computed over an interval of $(-1,1)$ instead of $(0, N - 1)$.

Image reconstruction is needed to restore the original image after the distortions occurred in them are removed. However, non-orthogonal moments such as geometric moments do not have direct reconstruction ability (Ghorbel et al., 2005; Teague, 1980). Using the relationship between geometric moments and family of orthogonal moments such as Zernike, Tchebichef and Krawchouk moments, an indirect approach to reconstruct the original image can be performed as shown in (Mukundan, Ong, & Lee, 2001a) for Tchebichef moments. Ghorbel et. al (Ghorbel et al., 2005) uses the concept of characteristic function proposed in (Teague, 1980) to reconstruct the original image from

its geometric moments. In this method, they computed the Discrete Fourier Transforms (DFTs) of an image from its geometric moments and reconstruct the original image using the inverse DFT (IDFT). However, when the order of the geometric moments is the same as the size of the image, the computation produces error (Flusser et al., 2009). A direct method to reconstruct images from its computed geometric moments is proposed by Flusser et. al in (Flusser et al., 2009). This method successfully reconstructs the exact image up to 11×11 . For larger images, this method lost its precision and is unable to reconstruct the exact image. Recently, (Honarvar et al., 2014) introduced a novel method of image reconstruction from its geometric moments using Stirling numbers of the first and second kinds . By using the full set of the geometric moments of an image, the original image is reconstructed. This method is computationally intensive and has long computational time. These observations provided a motivation to explore the potential of using Geometric moments in the area of image restoration. To summarize, following are the three contributions in this chapter

1. A method of estimating the parameters of the PSF from the degraded image is proposed. This is achieved by training ELM using geometric moments as feature vectors.
2. Once the estimate of the PSF is done, a mathematical framework for deconvolution is proposed in moment domain to obtain the GM's of the restored image.
3. A fast inverse reconstruction architecture is proposed to transform the restored sharp image from moment domain back to the spatial domain.

The main contributions of this chapter are clearly explained in the block diagram shown in Figure 4.1. To explain this in detail, feature vectors (WGM) are taken from the blurred image. These vectors are fed in to extreme learning machine (ELM) for training

the network. Once the network is trained, it estimates the parameters (σ, w) of the PSF, which is used to deconvolve the images in the proposed moment domain. The estimation result is fed into the subtractor circuitry to transform the image from moment domain back to the spatial domain.

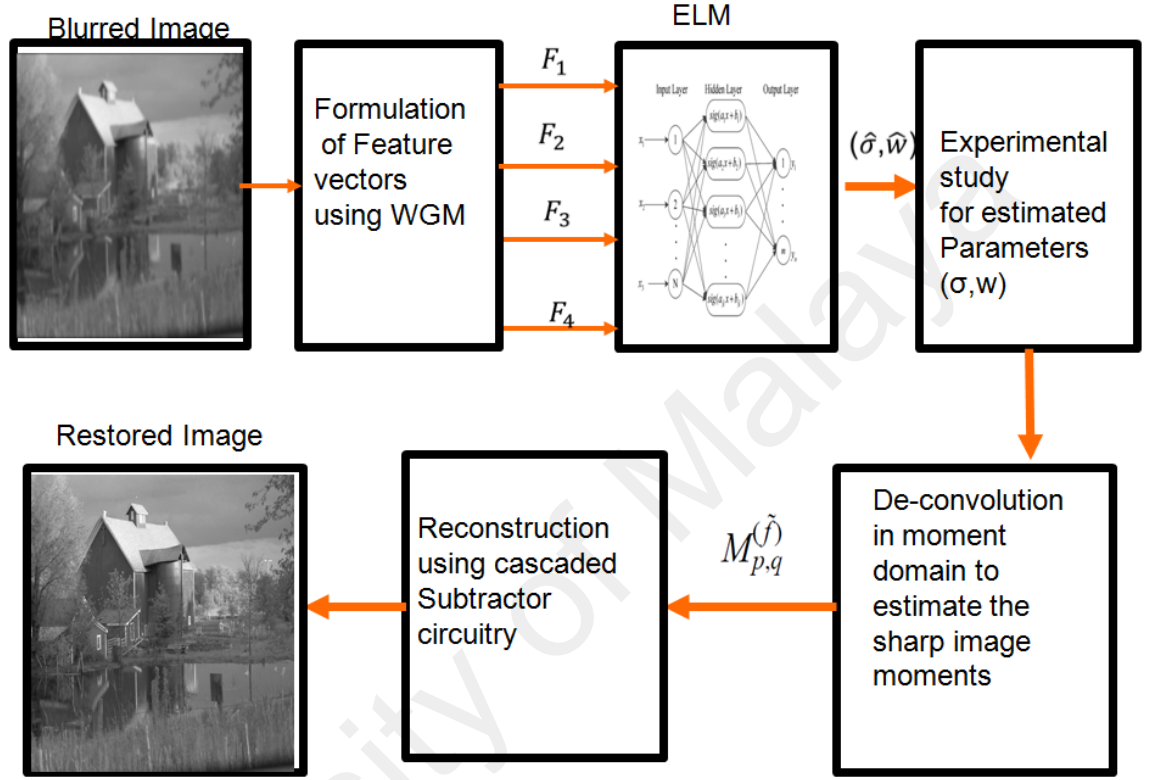


Figure 4.1: Flowchart of the proposed method

The rest of the chapter is organized as follows. Section 4.2 discusses the proposed formulation of feature vectors using Weighted geometric moments. Further their role as edge descriptors is explored. The proposed PSF parameter estimation via Extreme learning machine is discussed in Section 4.3. Section 4.4 presents the experimental study on PSF parameters estimation. This involves a discussion on cross-database analysis. Section 4.5 describes the proposed image restoration in moment domain. Comparative study with the state of the art methods (Kotera et al., 2013; A. Goldstein & Fattal, 2012; G. Liu et al., 2014; Vera et al., 2013a; Yan et al., 2012) is described in Section 4.6, where the images are taken from the same databases used in the existing methods for fair evaluation. Further, recently introduced methods that deals with Gaussian deblurring are

also compared with the existing methods. Finally, Section 4.7 concludes the chapter.

4.2 Formulation of feature vectors using Weighted geometric moments (WGM)

The classical restoration methods require a complete information about the PSF prior to restoring the degraded image. However, practically in many real applications, the PSF is often unknown or partial information is available within a certain range due to various practical constraints. As the information is rarely available, identification of the PSF from the observed degraded images has been of great interest.

In some applications such as remote sensing, the PSF of the atmospheric turbulence is modeled as

$$h(x,y) = \begin{cases} \frac{1}{\sqrt{(2\pi)\sigma}} \exp\left(\frac{-(x^2+y^2)}{2\sigma^2}\right), & \text{if } (x,y) \in w. \\ 0, & \text{otherwise.} \end{cases} \quad (4.4)$$

where, x and y are the horizontal and vertical space variables, σ is the standard deviation which parametrizes the degree of blur, w is the region of support size of the PSF. Thus, the Gaussian PSF is characterized by the two parameters σ and w respectively. To obtain a good restored image quality, proper estimation of these PSF parameters is essential.

In this chapter, a new identification method for Gaussian PSF parameters is proposed based on the observations that some of the geometric moments can represent the PSF information in terms of the edges present in an image when subjected to varying degree of Gaussian PSF. This is achieved using a well known fact that Tchebichef moments are well known edge descriptors of an image. Using the information geometric moments is expressed in terms of Tchebichef moments which are then used as feature vectors.

It has been shown that the low order Tchebichef moments can be used for edge detection (Thung, Paramesran, & Lim, 2012). Using the relationship between the Tchebichef moments and Geometric Moments (Mukundan et al., 2001b), a set of weighted Geometric moments (WGM) can be derived and used for edge detection. The weighted Geometric

moments (WGM) is defined as

$$WGM_{pq} = \sum_{k=0}^p \tilde{\alpha}_1(p, N; k) \sum_{l=0}^q \tilde{\alpha}_2(q, N; l) \sum_{i=0}^k \sum_{j=0}^l s_k^{(i)} s_l^{(j)} m_{ij} \quad (4.5)$$

where

$$\tilde{\alpha}_1(p, N; k) = (-1)^{p-k} \left(\frac{N^p}{(2p)! \binom{N+p}{2p+1}} \right) \left(\frac{p!}{k!} \right) \binom{N-1-k}{p-k} \binom{p+k}{p}$$

$$\tilde{\alpha}_2(q, N; l) = (-1)^{q-l} \left(\frac{N^q}{(2q)! \binom{N+q}{2q+1}} \right) \left(\frac{q!}{l!} \right) \binom{N-1-l}{q-l} \binom{q+l}{q}$$

The proposed weighted geometric moments (WGM) are defined as

$$WGM_{01} = \mathbf{A} m_{00} + \mathbf{B} m_{01}$$

$$WGM_{10} = \mathbf{A} m_{00} + \mathbf{B} m_{10} \quad (4.6)$$

$$WGM_{02} = \mathbf{C} m_{00} + \mathbf{D} m_{01} + \mathbf{E} m_{02}$$

$$WGM_{20} = \mathbf{C} m_{00} + \mathbf{D} m_{10} + \mathbf{E} m_{20} \quad (4.7)$$

$$WGM_{03} = \mathbf{F} m_{00} + \mathbf{G} m_{01} + \mathbf{H} m_{02} + \mathbf{I} m_{03}$$

$$WGM_{30} = \mathbf{F} m_{00} + \mathbf{G} m_{10} + \mathbf{H} m_{20} + \mathbf{I} m_{30} \quad (4.8)$$

$$WGM_{04} = \mathbf{J} m_{00} + \mathbf{K} m_{01} + \mathbf{L} m_{02} + \mathbf{M} m_{03} + \mathbf{N} m_{04}$$

$$WGM_{40} = \mathbf{J} m_{00} + \mathbf{K} m_{10} + \mathbf{L} m_{20} + \mathbf{M} m_{30} + \mathbf{N} m_{40} \quad (4.9)$$

where, the weights are defined as

$$\mathbf{A} = \left(\frac{3(3-N)}{N(N^2-1)} \right) \text{ and } \mathbf{B} = \left(\frac{6}{N(N^2-1)} \right);$$

$$\mathbf{C} = \left(\frac{5(N-1)(N-2)}{(N^2-4)(N^2-1)} \right), \mathbf{D} = \left(\frac{30(1-N)}{(N^2-4)(N^2-1)} \right) \text{ and } \mathbf{E} = \left(\frac{30}{(N^2-4)(N^2-1)} \right);$$

$$\mathbf{F} = \left(\frac{-7N(N-1)(N-2)(N-3)}{(N^2-9)(N^2-4)(N^2-1)} \right), \mathbf{G} = \left(\frac{14N(N-3)(6N-53)}{(N^2-9)(N^2-4)(N^2-1)} \right)$$

$$\mathbf{H} = \left(\frac{7N(150-30N)}{(N^2-9)(N^2-4)(N^2-1)} \right) \text{ and } \mathbf{I} = \left(\frac{140N}{(N^2-9)(N^2-4)(N^2-1)} \right);$$

$$\mathbf{J} = \left(\frac{9N^2(N-1)(N-2)(N-3)(N-4)}{(N^2-16)(N^2-9)(N^2-4)(N^2-1)} \right),$$

$$\mathbf{K} = \left(\frac{45N^2((N-3)(N-4)(17-4N))}{(N^2-16)(N^2-9)(N^2-4)(N^2-1)} \right),$$

$$\mathbf{L} = \left(\frac{405N^2(N-3)(N-4) - 3780N^2(N-1) + 6930N^2}{(N^2-16)(N^2-9)(N^2-4)(N^2-1)} \right),$$

$$\mathbf{M} = \left(\frac{9N^2(560-140N)}{(N^2-16)(N^2-9)(N^2-4)(N^2-1)} \right) \text{ and}$$

$$\mathbf{N} = \left(\frac{63N^2}{(N^2-16)(N^2-9)(N^2-4)(N^2-1)} \right).$$

4.2.1 Weighted Geometric Moments (WGM) as edge descriptors

By employing these WGM's for edge description, it will certainly encode the PSF information in a similar way as Tchebichef moment did. Now, the applicability of these WGM for the estimation of PSF parameters is discussed. Here, the characteristics of WGMs defined in (4.6)-(4.9) for various types of edge contributions i.e. horizontal, verti-

cal or diagonal present in an image is explored. This is done because edges provide vital information for humans to subjectively quantify the sharpness of the image. For this the proposed feature vector is as follows

$$\mathbf{F} = [\beta_1, \beta_2, \dots, \beta_4] \quad (4.10)$$

where, β_i is defined as

$$\beta_i = \sqrt{WGM_{0i}^2 + WGM_{i0}^2}, \text{ for } i=1,2,\dots,4 \quad (4.11)$$

The advantage of using the feature vector \mathbf{F} is that it can characterize not only horizontal and vertical edges but also the diagonal edges as well. This is explained by studying the behavior of the feature vector \mathbf{F} for varying degree of Gaussian PSF parameter σ by selecting the random patches of size 8×8 from the Cameraman image.

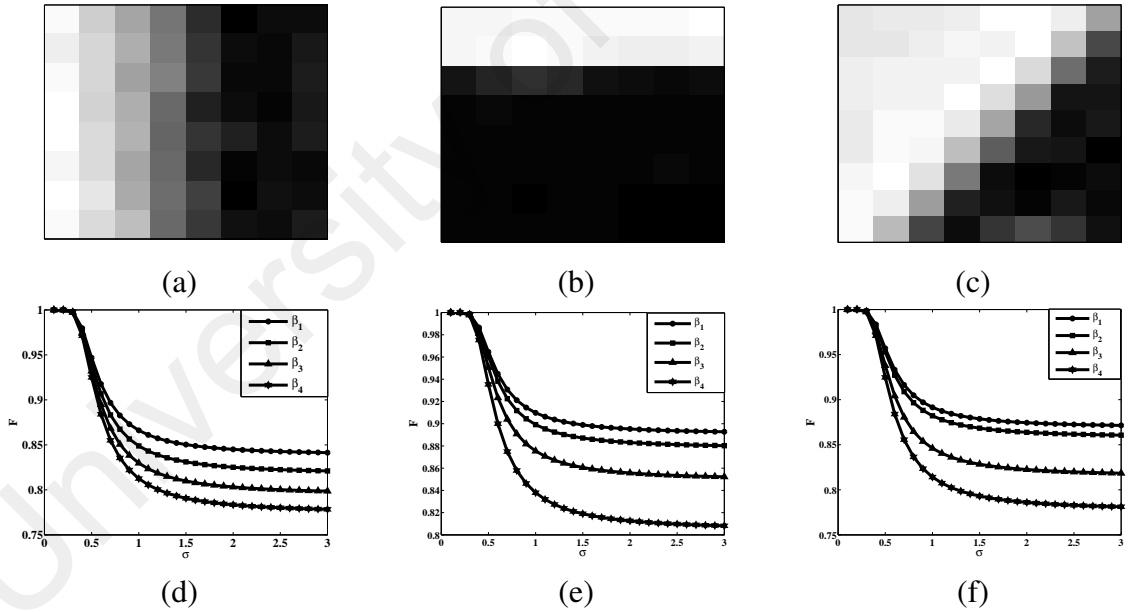


Figure 4.2: (a) Vertical (b) Horizontal and (c) Diagonal edge blocks of size 8×8 ; (d)-(f) \mathbf{F} variation with σ

One of the important characteristic in choosing the features for estimating the degree of blurriness in an image is that they should closely exhibit human visual system (HVS) perception of image blur distortion. It can be observed from Figure 4.2 that the components of the feature vector \mathbf{F} i.e β given in (4.11) decreases monotonically with the

increase in PSF parameter σ (the degree of blurriness increases). This is in accordance with the correlation between the human subjective score with the sigma σ for the LIVE image database. The same holds true for most of the natural images present in other databases also.

Now, the proposed algorithm for estimating the PSF parameters σ and w is discussed. As notified earlier that it is important to identify the edge blocks (patches of size 8×8) in an image as it encodes the information about the PSF. And the proposed algorithm relies on this concept. Hence, it becomes absolutely essential pre-process the image by classifying it into a plain and edge blocks first and then apply the proposed algorithm on the edge blocks to estimate the PSF parameters. Next, the proposed classification technique followed by the proposed algorithm is discussed.

4.2.2 Plain and Edge block classification

Here, a classification method based on Geometric moments to classify the image into plain and edge blocks is proposed. The method is based on the principle that for each 8×8 image block, the best fit ellipse is computed using Geometric moments (Teague, 1980). The ellipse is defined using two parameters as follows

$$a = \left(\frac{\mu_{20} + \mu_{02} + [(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2]^{1/2}}{\mu_{00}/2} \right)^{1/2} \quad (4.12)$$

$$b = \left(\frac{\mu_{20} + \mu_{02} - [(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2]^{1/2}}{\mu_{00}/2} \right)^{1/2} \quad (4.13)$$

where, a and b are known as semi-major and semi-minor axis respectively, of an ellipse while μ_{ij} are the central moments (Teague, 1980). Based on the value of b/a , the image block is classified as plain or edge block. The threshold value of b/a for the classification is 0.98. Any value less than that is classified as an edge, else its a plain block. This is verified by performing the classification of all the images present in the LIVE database.

For illustration, three images with varying image content are presented in Figure 4.3 along with their classification map which shows the location of edge patches present.

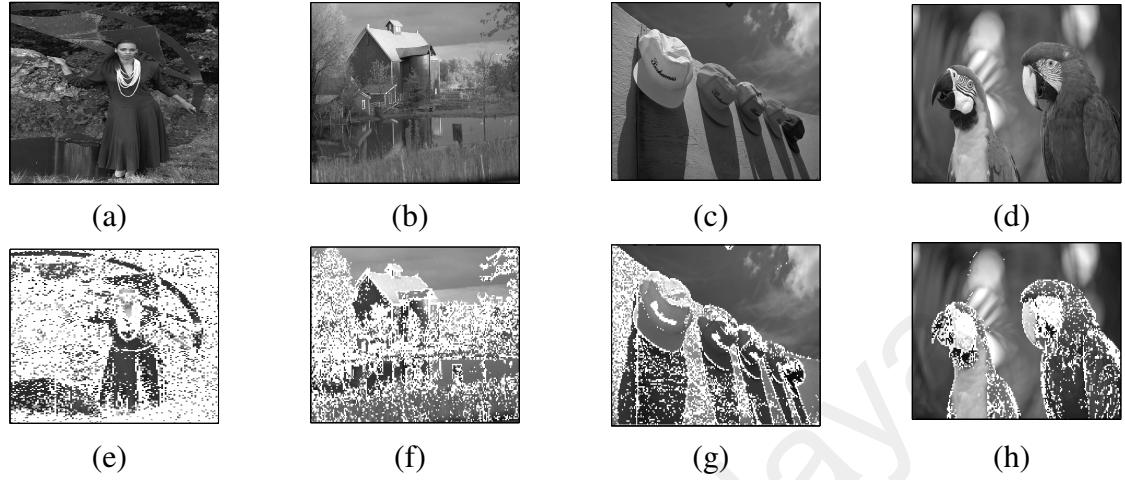


Figure 4.3: (a-c) Images with varying edge content (d)-(f) Classification map showing edge patches

Table 4.1 shows the detail breakdown of the image into edges and plain blocks. The reason for selecting these images of varying edge content is to serve two objectives: (1) To show the performance of the proposed PSF parameter algorithm for varying edge block contributions. (2) Most of the natural images fall into one of these three categories and hence can represent a large class of images.

Table 4.1: Classification of LIVE database images based on image content

Images	Plain	Edges
<i>Women</i>	30%	70%
<i>house</i>	58%	42%
<i>caps</i>	77%	23%
<i>parrots</i>	85%	15%

4.3 PSF parameters estimation via Extreme learning machine (ELM)

The proposed algorithm for estimating the PSF parameters σ and w requires the use of weighted geometric moments (WGM) which are shown to work as good feature descriptors that can effectively measure the edge profile of an image subjected to Gaussian blurring. In the proposed work, reblurring of the image is done to obtain the relative edge

information between the blurred and the reblurred image in terms of the WGM features. This is followed by the use of ELM, which uses these feature vectors to estimating the PSF parameters sigma (σ) and size (w) respectively.

Extreme learning machine (ELM) (Huang et al., 2006) is basically a feed-forward neural network used for the purpose of classification and regression analysis. It comprises of a single layer of hidden nodes, where the weights connecting the inputs to the hidden nodes are randomly initialized and are never updated. These weights connected between the hidden and output nodes are learned in a single step, which is similar to a learning of a linear model. The training of the extreme learning machine algorithm is to learn a model of the form

$$\hat{\mathbf{Y}} = \mathbf{W}_2 \sigma(\mathbf{W}_1 \mathbf{x}) \quad (4.14)$$

where, \mathbf{W}_1 is the input-to-hidden-layer weight matrix ; \mathbf{W}_2 is the hidden-to-output-layer weight matrix; σ is an activation function and \mathbf{x} is the input feature vectors.

The flowchart of the proposed algorithm to estimate the PSF parameters is given in Figure 4.4. The detailed description of the algorithm is given as follows:

1. Identify the edges and plain blocks in an image. Blurring of an image mostly affects the edges while keeping plain blocks are least affected. Hence, the algorithm will use the feature vectors that correspond to edge blocks rather than the plains to estimate the PSF parameters. The size of the edge block is 8×8 .
2. Once the edge blocks are identified, the mean of feature vector \mathbf{F}^b is calculated for all the edge blocks given as

$$\mathbf{F}^b = \frac{1}{M} \sum_{j=1}^M \mathbf{F}_j \quad (4.15)$$

where, M is the total number of edge blocks in an image and \mathbf{F} is the feature vector evaluated using (4.10).

3. Reblur the degraded image.
4. Calculate the re-blurred feature vector \mathbf{F}^{rb} by repeating step (2).
5. Use the feature vector $|\mathbf{F}^b - \mathbf{F}^{rb}|$ for training the ELM.
6. Once the ELM is trained, the estimated PSF parameters (σ, w) are used in the image restoration techniques to restore back the original image.

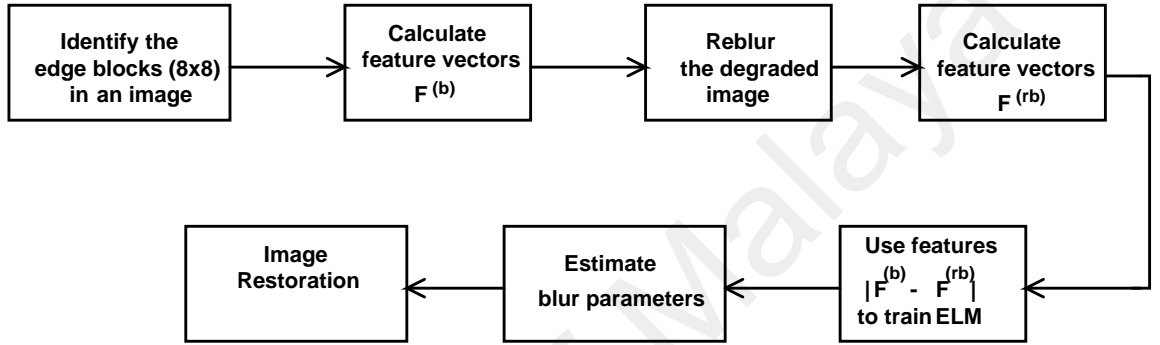


Figure 4.4: Proposed algorithm for PSF estimation

To ensure the better generalization ability of the ELM network, the optimal selection of the hidden neurons, the corresponding input weights (W) and bias values (B) are required. One way of achieving this is by randomly selecting the fixed input weights and the bias values, the optimal output weights are evaluated using least square approach. However, the generalization performance of the ELM depends on the proper selection of the hidden neurons, input weights and bias values. A selection of these parameters is crucial for proper generalization by ELM. In the proposed work, ELM network with 80 hidden neurons is considered. ELM algorithm is called 500 times for the same and cross database training/testing data and finds the mean and variance of the testing and training accuracies. Each time the ELM is called, the fixed parameters, namely weights (W) and bias (B) are initialized randomly from a uniform distribution. The input feature vectors are normalized between 0 to 1 and the weights (W) and bias (B) are initialized between ± 1 . Here, the unipolar sigmoidal activation function for the hidden neurons is opted. The

slope of the sigmoidal function is selected as 0.1 which is approximately equal to the number of input neurons (Suresh, Babu, & Kim, 2009; X. Ma et al., 2014).

For the case of testing and training from the same database, i.e. in this case LIVE database; the mean and the standard deviation of training efficiency are 96.32 and 0.0941 respectively. Similarly, the mean and standard deviation for testing efficiency are 92.46 and 0.0183 respectively. This is shown in Figure 4.5 for different runs. Further for the cross-database analysis shown in Figure 4.6, where training is performed using Berkeley dataset and testing using LIVE dataset; the mean and the standard deviation of training efficiency are 91.13 and 0.0163 respectively. Similarly, the mean and standard deviation for testing efficiency are 90.57 and 0.0146 respectively. From the Figs. 4.5-4.6, it can be seen that the random selection of the fixed parameters does not affect the generalization performance of the ELM regressor for the work.

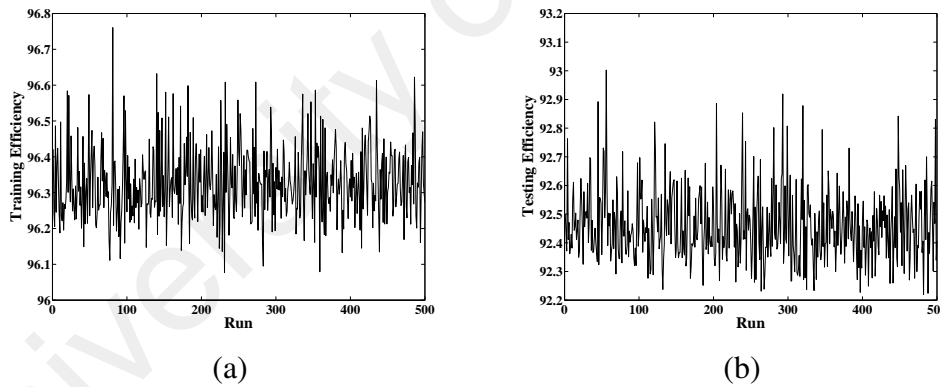


Figure 4.5: Effects of the initial parameter selection on training and testing performance for same database

Moreover, the generalization of the ELM depends on the number of hidden neurons. To illustrate this behavior, an experimental study is conducted by varying the hidden neurons from 20 to 100 with the steps of 10. The variation of the testing and training efficiency for the same and cross database analysis is shown in Figure 4.7 and 4.8 respectively. From Figure 4.7 it can be observed that the training and testing efficiency does not vary much with the variations of the hidden number of neurons and initial parameter

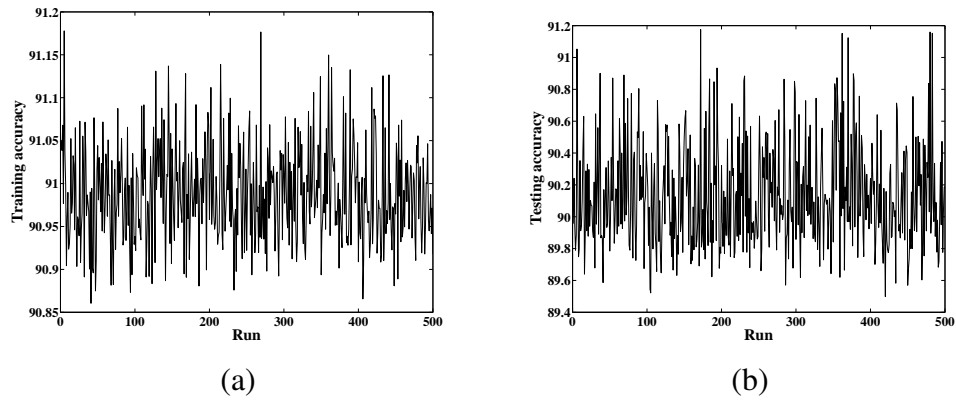


Figure 4.6: Effects of the initial parameter selection on training and testing performance for cross database

changes. In specific for the LIVE database under consideration, the training efficiency increases by a small amount from 94.14 to 96.62 as the hidden neurons increased from 20 to 100, which is not significant. The testing efficiency, however remains nearly constant to 92.53. Similarly, Figure 4.8 shows the training and testing efficiency for cross database analysis (Berkeley for training and LIVE for testing). It can be observed that both the testing and training accuracy remains nearly constant to 91.23 and 90.15 respectively, for the variation in the number of hidden neurons and initial parameter settings. These trends hold true for all the databases considered in the work for both individual and cross database analysis.

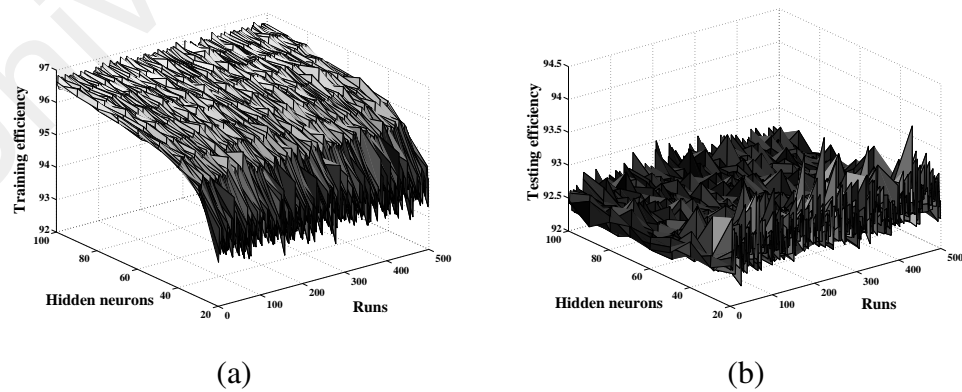


Figure 4.7: (a) Training (b) Testing accuracy variation with respect to hidden neurons and initial parameters for same database

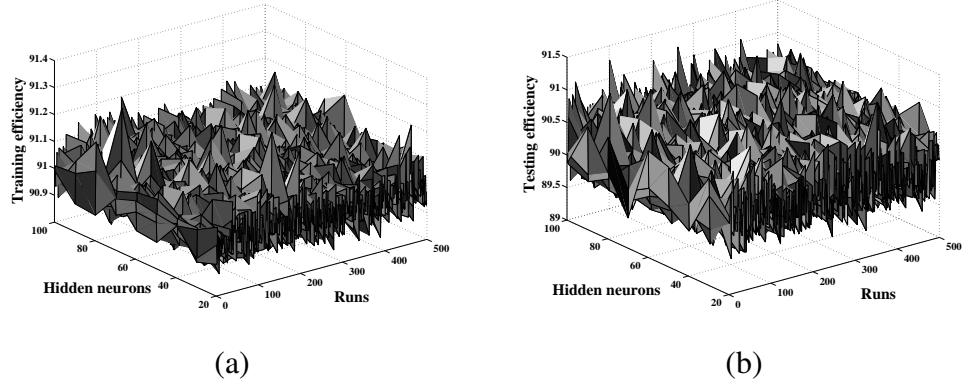


Figure 4.8: (a) Training (b) Testing accuracy variation with respect to hidden neurons and initial parameters for cross database

4.4 Experimental study for the estimated PSF parameters

In this section, first the importance of edge blocks in an image for the successful estimation of the PSF parameters (σ, w) is discussed. This is followed by performing two experiments in order to validate the performance of the proposed method on six different standard databases. In the first experiment, the proposed method of PSF estimation is trained using all the images in one database and tested on the remaining five database images i.e. cross-database analysis is performed. This is repeated for all the databases under consideration. In the second experiment the application of the proposed method in image restoration is explored where the restoration of the degraded images is performed in the geometric moment domain. The quality of the restored image is evaluated in terms of SSIM scores. Moreover, the comparative analysis with the three existing methods is also performed.

4.4.1 Data description

In this study, images are taken from six publicly accessible databases for training and testing of the proposed algorithm in order to estimate the PSF parameters: LIVE (Sheikh, Wang, Cormack, & Bovik, 2005), CSIQ (Larson & Chandler, 2010), CIDIQ (CIDIQ, n.d.), TID2008 (Ponomarenko et al., 2009), Caltech (Fink, 2003) and Berkeley

(Martin, Fowlkes, Tal, & Malik, 2001). Here, original images from all the databases are used except for Caltech and Berkeley, where a random selection of 34 and 32 images respectively is done. The images in all the databases vary in terms of its content and size. These images are successively degraded using Gaussian PSF with varying sigma (σ) and size (w) thus producing a total of 38,988 images to be used for training and testing of the proposed algorithm. The range of sigma (σ) varies from 0.3 to 4 in steps of 0.1 and size (w) from 7x7 to 17x17 in steps of 2x2. The details of the original images present in all the databases along with the number of images used for training and testing are shown in Table 5.1. The details of the original images present in all the databases along with the number of images used for training and testing are shown in Table 4.2.

Table 4.2: Details of testing and training samples for various databases

Database	Original Images	Total Images	Training	Testing
LIVE	29	6612	4600	2012
CIDIQ	23	5244	3670	1574
CSIQ	30	6840	4788	2052
Berkeley	32	7296	5107	2189
TID2008	25	5700	3990	1710
Caltech	34	7752	5425	2327

4.4.2 Significance of using Edge blocks for PSF estimation

As discussed earlier, the performance of the proposed PSF estimation algorithm relies on the edge blocks present in an image. For this a selection of four different images from the LIVE database is done. These images are of varying edge contributions as shown in Figure 4.3 along with Table 4.1. Two experiments are performed based on cross database evaluation i.e. TID2008 dataset (Ponomarenko et al., 2009) with 5700 images are used for training the ELM while testing is performed using LIVE [32] database having these four images. In the first experiment, only the edge blocks of the image are considered while in the second experiment, both the plain and edge blocks of the image are used in the computation. The proposed algorithm is implemented for both the cases and the

results are shown in Tables 4.3 and 4.4. Table 4.3 shows the estimation of σ for both the experiments along with their relative errors defined as $\left(\frac{x_0 - x}{x}\right)$ where x_0 is the measured and x is the true value. It can be observed that the edge blocks play an important role in the estimation of σ as can be seen from the third column of Table 4.3, where the algorithm closely estimates the actual sigma with a small relative error. However, the fifth column shows the results of the second experiment where the estimation of σ is not accurate when compared to the first experiment as can be observed from their relative errors. Hence, it can be concluded that edge blocks play an important role in the proposed algorithm for the estimation of σ .

Table 4.3: Role of edge block in estimating the PSF sigma (σ)

Images	σ	Edges		Edges +Plains		Images	σ	Edges		Edges+Plain	
		Estimated σ	Relative error	Estimated σ	Relative error			Estimated σ	Relative error	Estimated σ	Relative error
<i>women</i>	0.5	0.49	0.02	1.10	1.2	<i>caps</i>	0.5	0.52	0.04	1.3	1.6
	1.0	1.06	0.06	0.76	0.24		1.0	0.97	0.03	1.23	0.23
	1.5	1.53	0.02	1.67	0.11		1.5	1.47	0.02	2.17	0.44
	2.0	2.14	0.07	1.53	0.235		2.0	1.98	0.01	1.56	0.22
	2.5	2.48	0.008	1.58	0.368		2.5	2.62	0.04	2.89	0.16
	3.0	3.10	0.03	3.89	0.3		3.0	3.21	0.07	3.76	0.25
	3.5	3.61	0.03	3.12	0.10		3.5	3.67	0.04	2.54	0.27
	4.0	3.97	0.007	2.76	0.31		4.0	4.31	0.07	3.23	0.19
<i>house</i>	0.5	0.51	0.02	0.82	0.64	<i>parrots</i>	0.5	0.47	0.06	0.3	0.4
	1.0	0.98	0.02	0.57	0.43		1.0	0.96	0.04	1.6	0.6
	1.5	1.52	0.01	2.31	0.54		1.5	1.67	0.11	0.9	0.4
	2.0	1.97	0.01	1.56	0.22		2.0	2.12	0.06	2.65	0.32
	2.5	2.49	0.004	2.98	0.19		2.5	2.37	0.05	3.24	0.29
	3.0	3.23	0.07	3.67	0.22		3.0	3.32	0.1	2.27	0.24
	3.5	3.62	0.03	4.26	0.21		3.5	3.43	0.02	4.21	0.2
	4.0	4.23	0.05	3.76	0.06		4.0	3.64	0.09	3.26	0.18

A similar experiment to estimate the PSF size (w) is carried out and shown in Table 4.4. It can be observed that by considering the edge blocks of an image, the proposed algorithm closely estimates the PSF size effectively. On the other hand the error increases in the estimation if both the edge and plain blocks are considered.

Table 4.4: Role of edge block in estimating the PSF size (w)

Images	(w)	Edges	Edges +Plains	Images	(w)	Edges	Edges+Plain
		Estimated (w)	Estimated (w)			Estimated (w)	Estimated (w)
<i>women</i>	5×5	5×5	11×11	<i>caps</i>	5×5	5×5	9×9
	7×7	9×9	13×13		7×7	5×5	13×13
	9×9	7×7	5×5		9×9	11×11	5×5
	11×11	11×11	5×5		11×11	13×13	7×7
	13×13	13×13	9×9		13×13	13×13	7×7
	15×15	17×17	7×7		15×15	13×13	9×9
	17×17	17×17	13×13		17×17	13×13	11×11
<i>house</i>	5×5	5×5	11×11	<i>parrots</i>	5×5	7×7	9×9
	7×7	5×5	9×9		7×7	7×7	13×13
	9×9	11×11	5×5		9×9	7×7	5×5
	11×11	9×9	17×17		11×11	13×13	17×17
	13×13	11×11	9×9		13×13	13×13	7×7
	15×15	17×17	7×7		15×15	15×15	5×5
	17×17	15×15	9×9		17×17	15×15	9×9

4.4.3 Cross-database validation

To obtain a good quality of the restored image, correct estimation of the PSF parameters is essential. This is achieved using the ELM, which is a single hidden layer feed-forward network, where the input weights are chosen randomly and the output weights are calculated analytically. For hidden neurons, many activation functions can be used such as sigmoid, sine, Gaussian and hard limiting functions. Compared to other traditional computational intelligence techniques, ELM can provide better generalization performance at a much faster learning speed and with less human intervention. To validate the proposed method for the PSF estimation, all six databases are used.

A way to determine the generality of a machine-learning based PSF parameter estimation is the cross-database validation since images and/or distortions vary across databases. For the cross-database testing, the images from one database are used for training the ELM and tested using the images from the other remaining databases. Table 4.5 provides the Correlation Coefficients (CCs) values of the regression results of estimating sigma σ for all the possible combinations of datasets. In particular, the first row provides the CC values of the regression results when training is performed using TID2008 database and tested using other databases. In this case, the correlation coefficients (CC) values

of LIVE and CSIQ databases are 0.9245 and 0.9121 respectively, while Berkeley, CALTECH and CIDIQ provide CC values of 0.8312, 0.8023 and 0.8421 respectively. It can be observed that LIVE and CSIQ databases provide better prediction accuracies compared to other databases. Similarly, when Berkeley database is used for training, the CC scores registered for LIVE, CSIQ and CIDIQ are better compared to other databases. From this observation, it can be concluded that the proposed algorithm can estimate the PSF parameter σ well as indicated by the CC values. Similarly, Table 4.6 provides the CC values for the estimation of size w . For TID2008 database the CC values of LIVE and CSIQ databases are 0.8867 and 0.9232 respectively, while Berkeley, CALTECH and CIDIQ provide CC values of 0.8121, 0.8512 and 0.8032 respectively. In this case CSIQ database provides better prediction accuracies compared to other databases. However, when other databases like Berkeley, Caltech and CSIQ are used as training databases, the predication performance of the LIVE database is better.

Table 4.5: Correlation coefficient (CC) values for cross database performance of estimating PSF's sigma (σ).

Training	Testing					
	TID2008	Berkeley	LIVE	Caltech	CSIQ	CIDIQ
TID2008	-	0.8312	0.9245	0.8023	0.9121	0.8421
Berkeley	0.8257	-	0.8942	0.8021	0.8721	0.8736
LIVE	0.8511	0.8263	-	0.8138	0.8432	0.8045
Caltech	0.8182	0.8591	0.8021	-	0.8487	0.7821
CSIQ	0.7951	0.8421	0.9171	0.8161	-	0.8032
CIDIQ	0.8045	0.7976	0.7832	0.8287	0.8917	-

Table 4.6: Correlation coefficient (CC) values for cross database performance of estimating PSF's size (w).

Training	Testing					
	TID2008	Berkeley	LIVE	Caltech	CSIQ	CIDIQ
TID2008	-	0.8121	0.8867	0.8512	0.9232	0.8032
Berkeley	0.8491	-	0.8912	0.8275	0.8421	0.8612
LIVE	0.8421	0.8367	-	0.8034	0.8845	0.8363
Caltech	0.8412	0.8243	0.8523	-	0.7776	0.7892
CSIQ	0.8123	0.8267	0.9321	0.8261	-	0.8189
CIDIQ	0.7827	0.8237	0.7876	0.8225	0.9023	-

4.5 Proposed Image restoration in moment domain

Once the PSF is estimated, the restoration is performed using the Geometric Moments (GMs) domain. In order to estimate the GMs of the sharp image $f(x,y)$, knowledge of the GMs of the degraded image $g(x,y)$ and the PSF $h(x,y)$ is required. This is not a difficult task as the information about the $g(x,y)$ is already there while $h(x,y)$ is estimated using the proposed algorithm discussed in Figure 4.4. Transforming the $g(x,y)$ and $h(x,y)$ to GM domain is performed using (4.3) and the corresponding GMs are denoted as $M_{pq}^{(g)}$ and $M_{pq}^{(h)}$ respectively. The relationship between GMs of blurred image, sharp image and the PSF is given as

$$m_{p,q}^{(g)} = \sum_{i=0}^p \sum_{j=0}^q \binom{p}{i} \binom{q}{j} m_{i,j}^{(f)} m_{p-i,q-j}^{(h)} \quad (4.16)$$

The GMs of sharp image $m_{p,q}^{(f)}$ can be estimated by inverting the (4.16) as follows

$$m_{p,q}^{(f)} = \begin{cases} \left(\frac{1}{m_{0,0}^{(h)}} \right) \left[m_{p,q}^{(g)} - \sum_{i=0}^{p-1} \sum_{j=0}^{q-1} \binom{p}{i} \binom{q}{j} m_{i,j}^{(f)} m_{p-i,q-j}^{(h)} \right], & p,q \neq 0. \\ \frac{m_{0,0}^{(g)}}{m_{0,0}^{(h)}}, & p,q=0 \end{cases} \quad (4.17)$$

To explain the working principle of (4.17), a small image patch $f(x,y)$ of size 6×6 is blurred with a PSF $h(x,y)$ of size 3×3 PSF respectively. Let

$$f(x,y) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.18)$$

and

$$h(x,y) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad (4.19)$$

Using $g(x,y) = f(x,y) * h(x,y)$ one can obtain the blurred image as

$$g(x,y) = \begin{bmatrix} 1 & 2 & 3 & 2 & 3 & 2 \\ 3 & 4 & 6 & 5 & 4 & 3 \\ 2 & 3 & 4 & 4 & 3 & 1 \\ 3 & 4 & 3 & 2 & 1 & 1 \\ 1 & 2 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.20)$$

Using (2.1) and (2.21), the corresponding geometric moments of the blurred image

$g(x,y)$ and PSF $h(x,y)$ denoted as $m_{p,q}^{(g)}$ and $m_{p,q}^{(h)}$ respectively are given as

$$m_{p,q}^{(g)} = \begin{bmatrix} 75 & 247 & 983 & 4399 & 21239 & 108007 \\ 200 & 625 & 2383 & 10321 & 48631 & 242905 \\ 640 & 1909 & 6983 & 29257 & 134423 & 659089 \\ 2318 & 6655 & 23449 & 95155 & 426313 & 2051275 \\ 9124 & 25417 & 86675 & 341269 & 1490963 & 7035517 \\ 38030 & 103495 & 343153 & 1313611 & 5596801 & 25874275 \end{bmatrix} \quad (4.21)$$

and

$$m_{p,q}^{(h)} = \begin{bmatrix} 5 & 11 & 27 & 71 & 195 & 551 \\ 10 & 20 & 46 & 116 & 310 & 860 \\ 24 & 44 & 94 & 224 & 574 & 1544 \\ 64 & 110 & 220 & 494 & 1204 & 3110 \\ 180 & 296 & 562 & 1196 & 2770 & 6836 \\ 520 & 830 & 1516 & 3086 & 6820 & 16070 \end{bmatrix} \quad (4.22)$$

Using (4.17) one can obtain the restored geometric moments of the original image $f(x,y)$ given in (4.18) denoted by $m_{p,q}^{(\hat{f})}$, is obtained as

$$m_{p,q}^{(\hat{f})} = \begin{bmatrix} 15 & 32 & 86 & 266 & 890 & 3122 \\ 49 & 99 & 253 & 753 & 2449 & 8409 \\ 199 & 383 & 929 & 2633 & 8189 & 26993 \\ 925 & 1731 & 4057 & 11061 & 32965 & 103821 \\ 4663 & 8639 & 19961 & 53297 & 154373 & 469049 \\ 24709 & 45819 & 105793 & 280653 & 801709 & 2384229 \end{bmatrix} \quad (4.23)$$

Once the GMs of the original image is estimated, there is need to do the inverse moment transform which converts the image from moment domain back to the spatial domain. This is performed using the cascaded digital filters operating as subtractor. To understand the proposed approach better, Figure 4.9 shows the flowchart of the forward and the inverse transform process when applied to an image $f(x,y)$ and its GMs $m_{p,q}^{(f)}$ respectively.

With reference to Figure 4.9 the basic 2D cascaded digital filter architecture used by Hatamian and Wong et.al (Hatamian, 1986; Wong & Siu, 1999) is shown in Figure 4.10. Here, $H(z)$ is the digital filter. It takes the input as an image $f(x,y)$ and generates the digital filter output Y as shown in Figures 4.9 and 4.10 respectively. The only difference

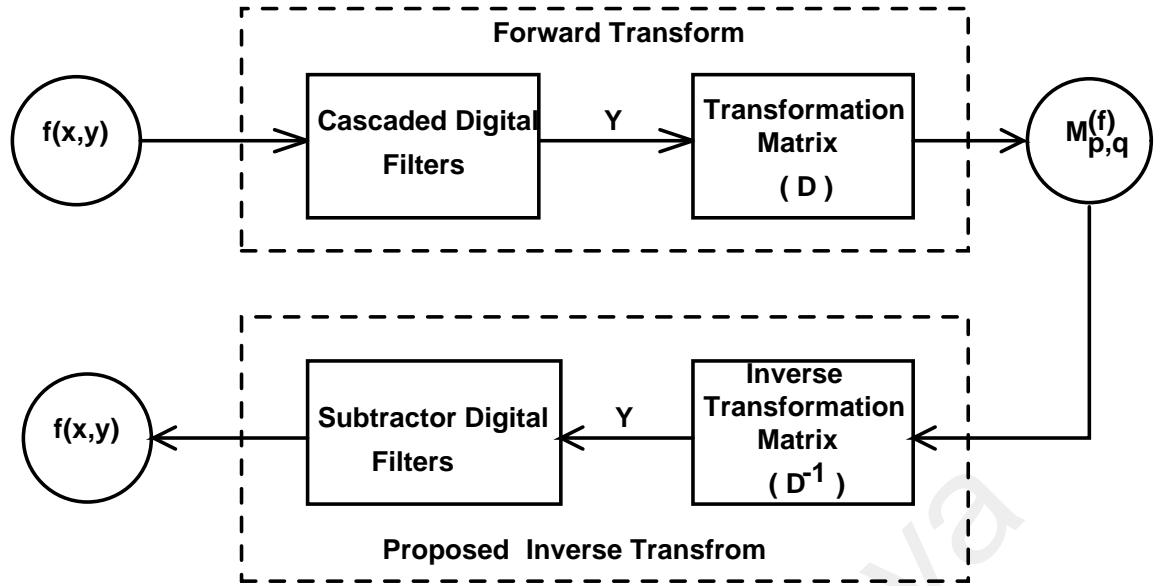


Figure 4.9: Moment transform (forward-inverse) via cascaded digital filters

between the Hatamian and Wong et al. work is the use of $H(z)$. In case of Hatamian, the $H(z) = \frac{z^{-1}}{1-z^{-1}}$ known as feed-forward filter while in Wong case it is $H(z) = \frac{1}{1-z^{-1}}$ known as feedback filter. Once the digital filter output Y is obtained, the forward transform i.e

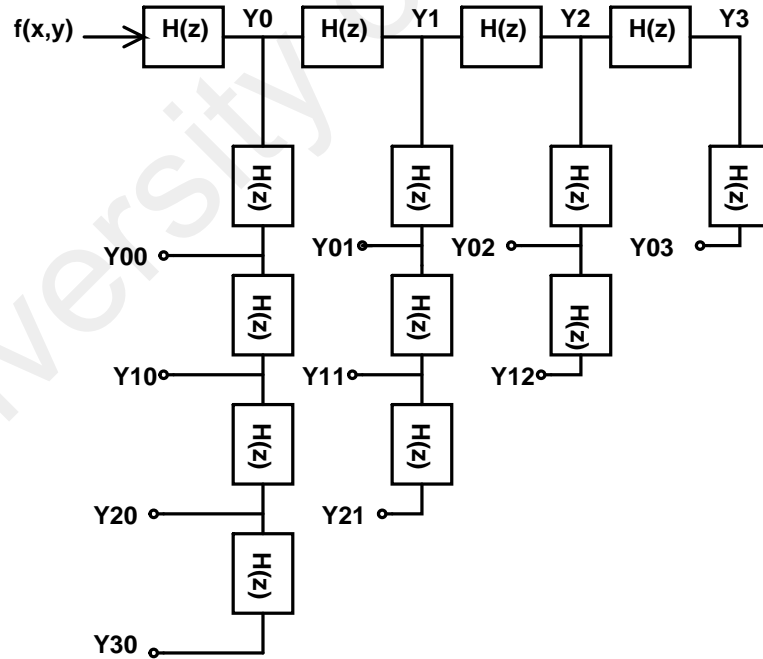


Figure 4.10: 2D cascaded digital filter architecture

geometric moments are obtained using

$$m_{p,q} = \sum_{r=0}^p \sum_{s=0}^q D_{pr} D_{qs} Y_{rs} \quad (4.24)$$

where, D is the transformation matrix for feed-forward architecture is given as (Al-Rawi & Jie, 2002)

$$D_{p,r} = \begin{cases} 0 & r > p \\ (-1)^p & r = 0, p \geq 0 \\ rD_{p-1,r-1} - (r+1)D_{p-1,r} & r > 0, p > 0 \end{cases} \quad (4.25)$$

while for the feedback architecture it is given as (Al-Rawi, 2008)

$$D_{p,r} = \begin{cases} 0 & (p > 0, r = 0) \text{ or } (r > p) \\ 1 & p = 0, r = 0 \\ r(D_{p-1,r-1} - D_{p-1,r}) & \text{otherwise} \end{cases} \quad (4.26)$$

Since the information of the restored GMs of the sharp image is given in (4.17), the focus of this chapter is to use an inverse moment transform based on the feedback architecture proposed by Wong et al. (Wong & Siu, 1999). For this, there is a need to calculate the inverse of the matrix D given in (4.26) as

$$E = D^{-1} \quad (4.27)$$

Once the inverse is calculated, the digital filter output Y can be calculated from the restored GMs of the sharp image as

$$Y_{p,q} = \sum_{r=0}^p \sum_{s=0}^q E_{pr} E_{qs} m_{rs}^{(f)} \quad (4.28)$$

Using the digital filter output Y given in (4.28), one can use the cascaded subtractor circuit to transform the digital filter outputs back into the image estimate $\hat{f}(x,y)$ in the

spatial domain as discussed in Chapter 2 (See Section 2.2.4) and is given as

$$\hat{f}(x,y) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.29)$$

which is the same as the image $f(x,y)$ given in (4.18)

4.6 Comparative analysis with the existing state of the art methods

Once the estimation of the PSF is done using the proposed algorithm, image restoration is performed to deblur the image in geometric moment domain as discussed in Sect 3. The quality of the restored image is compared with three existing methods: Kotera et al. (Kotera et al., 2013), Goldstein et al. (A. Goldstein & Fattal, 2012) and Liu et al. (G. Liu et al., 2014). These methods require the information of the PSF to restore the deblurred image by adopting different varieties of optimization techniques.

In this study, two experiments are performed using the blurred images of the LIVE database to evaluate the performance of the proposed method with the three existing methods. In the first experiment four images from the LIVE database that vary in image content as shown in Figure 4.3 are used to evaluate the visual quality in terms of SSIM scores. Further, visual quality of the two images is shown to demonstrate the effectiveness of the proposed approach. In the second experiment, a comparative analysis of the 29 original images blurred with specific pairs of (σ, w) is shown in terms of SSIM scores. Further, a detailed analysis of the proposed method with the images used in the existing methods is carried out to have a fair comparison in terms of SSIM scores.

4.6.1 Experiment 1

Table 4.7 presents the first experiment where the comparison of the proposed method with the existing ones in terms of the SSIM score is carried out. Here, the cross database evaluation is performed using the Berkeley database as training and LIVE database images for testing. The second and third column of the table shows the actual and the estimated PSF parameters. It can be observed that the proposed method estimates these parameters closely. Further, the SSIM scores of the proposed method along with the existing methods are given for the test images. The score provided in bold shows the maximum score. It can be seen that proposed method gives better SSIM scores when compared to the existing methods for blurred images caused by sigma (σ) less than 2.5. For sigma σ greater than 2.5 the proposed method in most cases performs better than the existing methods.

Table 4.7: Comparative analysis with the existing methods in terms of SSIM scores

Images	Actual (σ, w)	Estimated (σ, w)	SSIM			
			Proposed	(Kotera et al., 2013)	(G. Liu et al., 2014)	(A. Goldstein & Fattal, 2012)
<i>women</i>	(0.5,7)	(0.49, 9)	0.9673	0.9579	0.9236	0.9157
	(1.0,7)	(1.06,9)	0.9490	0.9276	0.9137	0.8905
	(1.5,9)	(1.53,7)	0.8978	0.8876	0.8643	0.8571
	(2.0,11)	(2.14,11)	0.8673	0.8602	0.8521	0.8469
	(2.5,13)	(2.48,13)	0.8251	0.8136	0.8175	0.8123
	(3.0,15)	(3.10,17)	0.8163	0.8089	0.8121	0.8103
	(3.5,15)	(3.61,17)	0.7986	0.7956	0.8032	0.8053
	(4.0,17)	(3.97,17)	0.7863	0.7921	0.7821	0.7769
	(0.5,9)	(0.51,11)	0.9721	0.9703	0.9632	0.9613
<i>house</i>	(1.0,7)	(0.98,5)	0.9695	0.9587	0.9512	0.9445
	(1.5,9)	(1.52,11)	0.9486	0.9412	0.9409	0.9312
	(2.0,11)	(1.97, 9)	0.9145	0.9043	0.8912	0.8832
	(2.5,13)	(2.49, 11)	0.9021	0.8921	0.8732	0.8702
	(3.0,13)	(3.23, 11)	0.8742	0.8834	0.8529	0.8489
	(3.5,15)	(3.62, 17)	0.8241	0.8308	0.7963	0.7843
	(4.0,17)	(4.23, 15)	0.8032	0.7952	0.7854	0.7784
	(0.5,7)	(0.52, 5)	0.9743	0.9521	0.9476	0.9212
	(1.0,7)	(0.97, 5)	0.9523	0.9356	0.9265	0.9087
<i>caps</i>	(1.5,11)	(1.47, 13)	0.9375	0.9245	0.9027	0.8945
	(2.0,9)	(1.98, 11)	0.9026	0.8875	0.8756	0.8721
	(2.5,11)	(2.62, 13)	0.8723	0.8652	0.8534	0.8487
	(3.0,11)	(3.21, 13)	0.8348	0.8479	0.8219	0.8067
	(3.5,17)	(3.67, 13)	0.8167	0.8054	0.7952	0.8012
	(4.0,15)	(4.31, 13)	0.7897	0.7921	0.7734	0.7864
	(0.5,7)	(0.47, 7)	0.9521	0.9397	0.9263	0.9123
	(1.0,9)	(0.96, 7)	0.9358	0.9178	0.9032	0.8978
	(1.5,7)	(1.67, 7)	0.9183	0.8976	0.8875	0.8786
<i>parrots</i>	(2.0,13)	(2.12, 13)	0.8832	0.8754	0.8813	0.8678
	(2.5,11)	(2.37, 13)	0.8434	0.8542	0.8412	0.8367
	(3.0,17)	(3.32, 15)	0.8253	0.8372	0.8045	0.7965
	(3.5,15)	(3.43, 15)	0.8026	0.7932	0.7861	0.7812
	(4.0,15)	(3.64, 15)	0.7945	0.7876	0.7732	0.7853

Two images from the Figure 4.3 are selected to show the visual comparison of the restoration performed using proposed and the existing methods. The two images *women*

and *house* are selected because of their varying edge content. The *women* image has the largest number of edge blocks while the image *house* has the nearly equal number of edge and plain blocks. Figure 4.11 shows the visual comparison of the image restoration performed on the *women* image. Here, the image is Gaussian blurred using $\sigma = 1.5$ and $w = 11$. It also shows the estimated PSF obtained using the proposed and the existing methods. It can be observed that the quality of the restored image is comparable for both the proposed and the method in (Kotera et al., 2013). One possible reason for the difference in the quality of the restored images with the other two methods in (A. Goldstein & Fattal, 2012; G. Liu et al., 2014) is the close estimation of the PSF from the degraded image. Any discrepancy in the estimation of the PSF will result in the degradation of the image quality. The PSF estimated using proposed method and the method in (Kotera et al., 2013) is quite close to the PSF with which the original image is distorted. Similar observations and conclusions can be made in the case of *house* image which is Gaussian blurred using $\sigma = 2.0$ and $w = 15$ as shown in Figure 4.12.



Figure 4.11: Comparison of the deblurring results on *woman* image blurred with Gaussian PSF of $(\sigma, w)=(1.5, 11)$

4.6.2 Experiment 2

In this experiment, Figure 4.13 presents the detailed comparative analysis of the proposed method with the existing ones in terms of the SSIM score for all the 29 LIVE database images. The analysis is based on cross database evaluation performed using

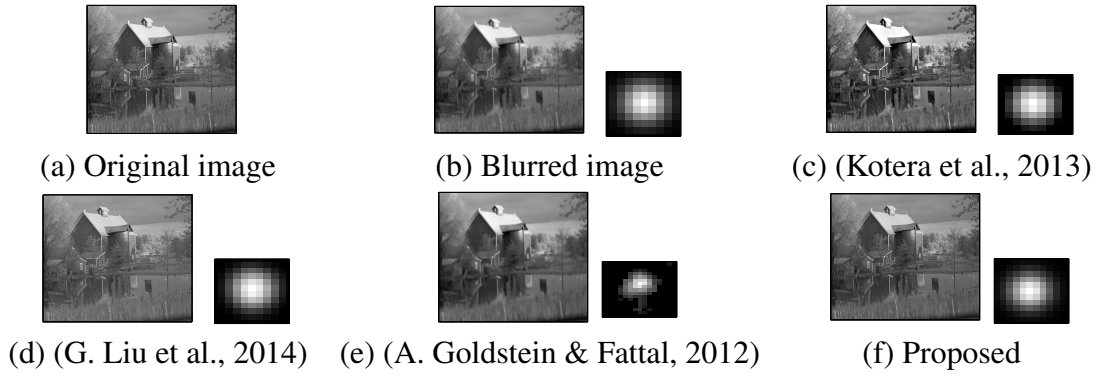


Figure 4.12: Comparison of the deblurring results on *house* image blurred with Gaussian PSF of $(\sigma, w)=(2.0,15)$

Berkeley database images for training and LIVE database images for testing. For instance, Figure 4.13(a) shows the SSIM score for all the restored 29 test images of LIVE database that are degraded using the PSF parameters $(\sigma, w)=(1.0,11)$. It can be seen that the proposed method provides better SSIM scores when compared to the restoration performed in (Kotera et al., 2013; A. Goldstein & Fattal, 2012; G. Liu et al., 2014) for most of the test images. Similar observations can be seen for Figure 4.13(b) where the SSIM scores of the restored test images degraded with PSF parameters $(2.0,13)$. Further, Figs. 4.13(c)-(d) shows the restored test images degraded by $(3.0,15)$ and $(4.0,15)$ respectively. In this case it can be observed that in most cases (Kotera et al., 2013) performs better when compared to the proposed method. However, the proposed method performs well than (A. Goldstein & Fattal, 2012) and (G. Liu et al., 2014) for all the images present in the LIVE database.

4.6.3 Comparative Analysis using images from the published database images

In order to gauge the effectiveness of the proposed method with the existing ones, a fair comparison in terms of the restoration is performed using the images present in the existing methods. Table 4.8 presents the comparative analysis of the restoration performed on the blurred images. Here, the blurred images are taken from the existing method by (G. Liu et al., 2014) for fair comparison. The restoration results obtained by the proposed

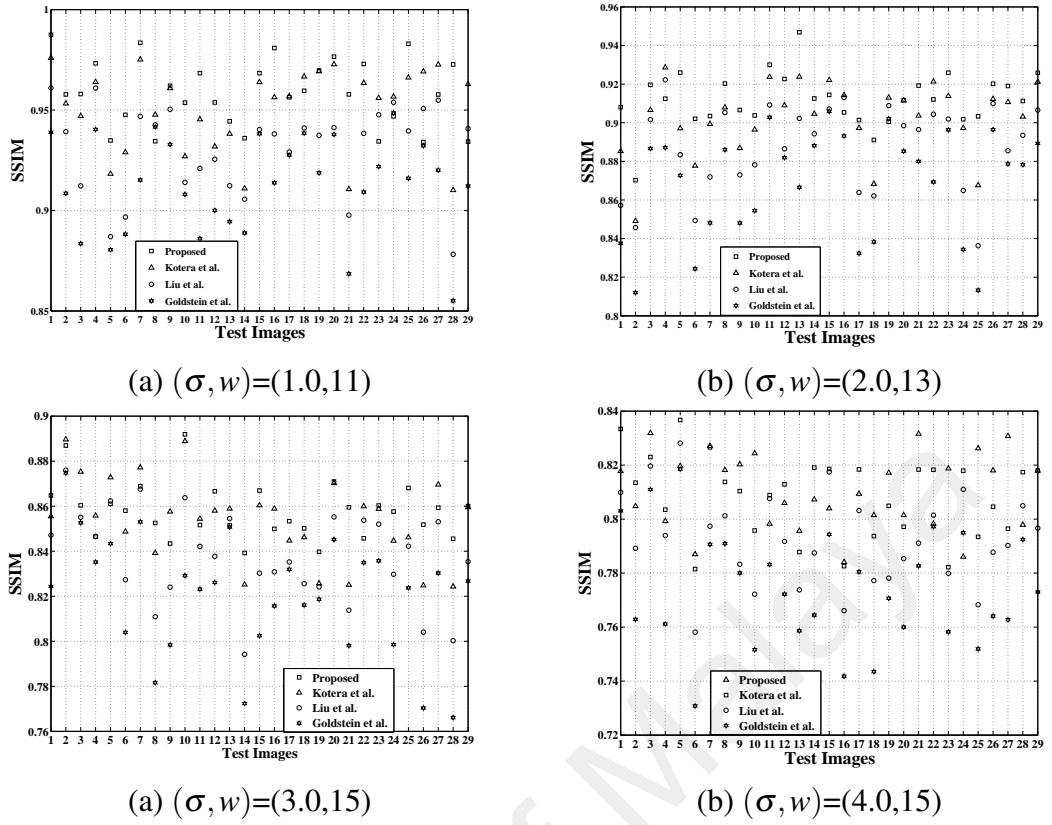













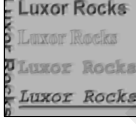
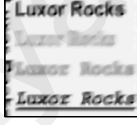







Figure 4.13: Comparative analysis of SSIM score for the LIVE database images for different values of PSF parameters (σ, w) with existing methods namely: Kotera et al. (Kotera et al., 2013), Goldstein et al. (A. Goldstein & Fattal, 2012) and (G. Liu et al., 2014)

method are based on the training of ELM using the Berkeley database and testing using the LIVE database and the images taken from (G. Liu et al., 2014). It can be observed that the proposed method does not provide satisfactory restoration results when compared to the work done by (G. Liu et al., 2014). The reason for this is that the images used in (G. Liu et al., 2014) are blurred using arbitrary PSF: a combination of Gaussian and motion PSF and the proposed method is based on restoration performed using Gaussian PSF. Due to the mismatch in the PSF estimation, the quality of the restored image using the proposed method is degraded compared to (G. Liu et al., 2014). This is observed strongly in the case of two images shown in second and third row of Table 4.8 where the quality of the restoration is not as good as other methods when evaluated in terms of SSIM score. However, for the images shown in first and last row of the table, the results are

Table 4.8: Quantitative comparison with the existing methods by using the images present in them



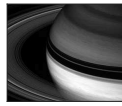

Blurred Image	Restored Image			
	(Kotera et al., 2013)	(G. Liu et al., 2014)	(A. Goldstein & Fattal, 2012)	Proposed
				
SSIM	0.7932	0.8312	0.7876	0.8279
				
SSIM	0.7834	0.7912	0.7543	0.7215
				
SSIM	0.7638	0.7852	0.7722	0.7621
				
SSIM	0.7965	0.8534	0.7651	0.8689

competitive with the existing methods.

Moreover, two recently proposed methods by (Vera et al., 2013a) and (Yan et al., 2012) that focus specifically for Gaussian blurred images are selected for a fair comparison with the proposed approach. Further, the two sharp images used in (G. Liu et al., 2014) are Gaussian blurred and taken for comparison with the proposed method. Doing this will explore the advantages and shortcomings of the proposed approach when compared with these methods. The restoration results obtained by the method is based on the training of ELM using Caltech database and testing using CSIQ database and the images taken from (G. Liu et al., 2014; Vera et al., 2013a; Yan et al., 2012). A detailed analysis of the restoration performed on the blurred images taken from these existing papers is carried out and shown in Table 4.9. The second and third column of the table shows the actual and the estimated PSF parameters. It can be observed that the proposed method estimates these parameters closely. Further, the SSIM scores of the proposed method along with the existing methods are given for the test images. The score provided in bold shows

the maximum score. It can be seen that proposed method gives better SSIM scores when compared to the existing methods for blurred images caused by sigma (σ) less than 2.5. For sigma σ greater than 2.5 the proposed method in most cases performs better than the existing methods. Further, Table 4.10 presents the quantitative comparison of the restora-

Table 4.9: Comparative analysis with the existing methods in terms of SSIM scores


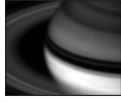
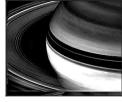
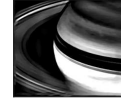
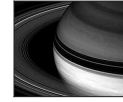





Images	Actual (σ, w)	Estimated (σ, w)	SSIM			
			Proposed	(Vera et al., 2013a)	(Yan et al., 2012)	(G. Liu et al., 2014)
	(0.5,7)	(0.52, 5)	0.9742	0.9645	0.9567	0.9357
	(1.0,9)	(0.98, 11)	0.9432	0.9236	0.9187	0.9021
	(1.5,11)	(1.53, 11)	0.9289	0.9034	0.8798	0.8723
	(2.0,11)	(2.12, 11)	0.8956	0.8645	0.8541	0.8421
	(2.5,13)	(2.48, 15)	0.8545	0.8325	0.8267	0.8176
	(3.0,13)	(3.13, 15)	0.8457	0.8248	0.8125	0.8067
	(3.5,15)	(3.52, 15)	0.8053	0.8124	0.8056	0.7978
	(4.0,17)	(4.05, 17)	0.7986	0.8045	0.7897	0.7675
	(0.5,7)	(0.52, 5)	0.9736	0.9631	0.9434	0.9315
	(1.0,9)	(1.12, 7)	0.9276	0.9125	0.9035	0.8879
	(1.5,11)	(1.49, 11)	0.9157	0.9072	0.8967	0.8746
	(2.0,11)	(1.88, 11)	0.9045	0.8921	0.8764	0.8658
	(2.5,13)	(2.46, 15)	0.8578	0.8469	0.8257	0.8445
	(3.0,13)	(3.17, 17)	0.8023	0.8034	0.8153	0.7812
	(3.5,15)	(3.65, 17)	0.7836	0.7971	0.8049	0.7851
	(4.0,17)	(4.16, 17)	0.7732	0.7845	0.7743	0.7621
	(0.5,7)	(0.51, 7)	0.9624	0.9543	0.9212	0.9145
	(1.0,9)	(1.12, 11)	0.9335	0.9065	0.9043	0.8956
	(1.5,11)	(1.49, 11)	0.9037	0.8723	0.8996	0.8732
	(2.0,11)	(2.12, 11)	0.8769	0.8582	0.8459	0.8256
	(2.5,13)	(2.47, 13)	0.8321	0.8497	0.8365	0.8171
	(3.0,13)	(3.16, 15)	0.8247	0.8136	0.8037	0.7937
	(3.5,15)	(3.48, 15)	0.8176	0.7959	0.7947	0.7745
	(4.0,17)	(4.17, 17)	0.7732	0.7818	0.7786	0.7632
	(0.5,7)	(0.52, 7)	0.9513	0.9456	0.9387	0.9261
	(1.0,9)	(1.17, 11)	0.9345	0.9214	0.9102	0.9073
	(1.5,11)	(1.62, 11)	0.9145	0.9025	0.8812	0.8942
	(2.0,11)	(2.16, 11)	0.8947	0.8867	0.8671	0.8468
	(2.5,13)	(2.32, 13)	0.8665	0.8734	0.8486	0.8271
	(3.0,13)	(3.21, 13)	0.8481	0.8346	0.8136	0.7936
	(3.5,15)	(3.77, 13)	0.8109	0.8347	0.8074	0.7645
	(4.0,17)	(3.66, 15)	0.7742	0.8067	0.7547	0.7321

tion results by the proposed method with the existing ones. In the first row of the table, it can be observed that the sharp features near the rings of the *saturn* are restored effectively using both the proposed and the method in (Vera et al., 2013a). This is reflected when a comparison is evaluated in terms of SSIM scores. Similarly, the texture of the *moon* shown in second row of table is restored effectively using the proposed method.

4.7 Conclusion

This chapter proposed the use of Weighted geometric moments (WGM) and extreme learning machine (ELM) for estimating the PSF. Once the PSF is known, deconvolution approach in moment domain is used to obtain the geometric moments of the restored image. This is followed by performing the transformation from moment domain to spatial

Table 4.10: Quantitative comparison with the existing methods by using the images present in them

Original Image	Blurred Image	Restored Image		
		(Vera et al., 2013a)	(Yan et al., 2012)	Proposed
				
SSIM	0.6245	0.8136	0.8037	0.8247
				
SSIM	0.5873	0.8067	0.7547	0.7742

domain using 2D cascaded digital filters operating as subtractors. The quality of the restored image is assessed using SSIM. A detailed comparative analysis is carried out with the three existing methods using the images from their database and it has been found that in most cases the proposed method gave better SSIM scores. Further, two recent methods that specifically deal with Gaussian blurred images are taken for a fair comparison with our method and it has been found that the proposed method provides competitive results.

CHAPTER 5 : ESTIMATION OF THE PSF PARAMETERS USING TCHEBICHEF MOMENTS FOR IMAGE RESTORATION

In this chapter, the estimation of the PSF parameter for Gaussian blurred images is carried out. The objective is to estimate the sigma and size (σ, w) as accurately as possible so that the performance of the restoration scheme is maximized. Tchebichef moments and extreme learning machine (ELM) are utilized for estimating these parameters. To validate the proposed method, experimental work has been carried out on six different databases, namely LIVE, CSIQ, CIDIQ, VCL, Caltech and Berkeley. These databases are used for the purpose of training/testing of the ELM and performs cross database analysis for PSF parameter estimation. Once the parameters are estimated, the information of the PSF is used along with widely used split Bregman algorithm to restore the degraded images. The restoration results obtained from the proposed method provides competitive results when compared with the existing methods in terms of SSIM score.

5.1 Introduction

PSF estimation is applicable to several practical image processing applications, such as image restoration, deblurring and autofocus (Rooms, Ronsse, Pizurica, & Philips, 2002), turbulence removal (van Eekeren et al., 2012) and super-resolution (Van Eekeren, Schutte, Dijk, de Lange, & van Vliet, 2006). The goal of image restoration is to recover an approximate version of the original image from the degraded image. The image degradation happens due to various reasons such as camera motion, atmospheric turbulence and out of focus.

Generally, a typical image deblurring problem can be categorized into non-blind and blind. Non-blind image deblurring requires the prior model for image deblurring while in blind deblurring the PSF operator is unknown. However, in most of the practical applica-

tions the PSF is not known exactly resulting in the limited applications of non-blind image deblurring. Hence, blind deconvolution is the practical method for restoring the blurred images. The Bayesian framework method for achieving the blind restoration task is considered in (Likas & Galatsanos, 2004). Here, the parameters of the Gaussian PSF are estimated using Wavelet decomposition (Rooms, Philips, & Portilla, 2004). Restoration of the images caused by motion blur is also discussed in (Rekleitis, 1996). Furthermore, Radon transforms is also used to estimate the PSF kernel by analyzing the prominent edges in the image (Cho, Paris, Horn, & Freeman, 2011). Other methods explored are cepstral method and steerable filters (Krahmer et al., 2006).

In image deblurring, edges are the prominent structures that are affected the most. The standard approach to edge detection fails to estimate the PSF parameters in the case of blur. In (Elder & Zucker, 1998) the method of local scale control is applied to the problem of estimating the edges in an image which in turns helps in estimating the PSF parameters. Furthermore, they proved that the edges covering a large range of blur scales and contrast can be recovered accurately with no input parameter other than the second moment of the sensor noise. Bouma et al. (Bouma et al., 2012) estimates the PSF parameters using the first order derivative. They showed that accurate, precise and efficient PSF estimation can be computed at the edge locations. The method is robust against the small variations that occur due to noise and dislocation. Dijk et al. (Dijk et al., 2003) estimated the sharpness of the natural image using Gaussian models. They located the edges in an image followed by application of Gaussian derivatives. It provides a response function to which an estimate of the edge width is obtained and the corresponding sharpness measure is proposed. Hu et al. (H. Hu & De Hann, 2006) proposed a novel PSF estimation method in which the blurred input image is first re-blurred by Gaussian PSF of different blur radii. Then the difference ratios are calculated between the multiple re-blurred images and the input image to determine the unknown PSF radius.

Chiang et al. (Chiang & Boulton, 1997) proposed a super-resolution algorithm for enhancing image resolution based on edge models and PSF estimation. A cubic polynomial is used for estimating these parameters. Their technique is robust to lighting variation and edge localization. Bae et al. (Bae & Durand, 2007) adopted the method proposed in (Elder & Zucker, 1998) along with the addition of bilaterally filtering step to remove the outliers in the estimate present due to blurry features such as soft shadows. This modified approach of using brute-force strategy for fitting second order derivative models is more robust than the original one as they found that the localization of the zero crossing for the third order derivative is not robust. Apart from spatial domain, wavelet domain is also explored for the estimation of PSF parameters (Rooms, Pizurica, & Philips, 2002; Tong, Li, Zhang, & Zhang, 2004). Rooms et al. (Rooms, Pizurica, & Philips, 2002) used the fact that the sharpness of the sharpest edges in the blurred image, contains the information about the blurring PSF. Hence, they proposed a smoothness measure known as Lipschitz exponent, which is computed for these sharpest edges. A relationship between the variance of a Gaussian PSF and the Lipschitz exponent is derived and it was shown that it depends only on the blur in the image and not on the image contents. Hence, it was used to estimate the sigma σ of the PSF. Bayes classification based blur feature detection, which uses local auto-correlation congruency is explored in (R. Liu, Li, & Jia, 2008). Another method proposed by (B. Su, Lu, & Tan, 2011) is based on alpha channel feature which was used to detect the local blur feature in an image. Ducottet et al. (Ducottet, Fournel, & Barat, 2004) proposed a new method for edge detections and its characterization using wavelet technique. The technique depends on the modeling of contours as smoothed singularities of transition, peak and line. Kayargadde (Kayargadde, 1995) used polynomial to transform along with multi-filter banks to derive the relationship between the Hermite coefficients of blurred and un-blurred images. This result helped in solving PSF related problems as they exploit the relationship to estimate the parameters of

a blurred edge and the parameters of the image blurring kernel. Recently Soleimani et al. (Soleimani, Rooms, & Philips, 2013) proposed an efficient PSF estimation technique with the use of multi-scale quadrature filters. They used the energy of the filters with first and second order derivatives of the Gaussian for estimation.

Once the parameters of the PSF are estimated, its values can be used in several restoration algorithms to obtain the restored image. In particular, Zhang et al. (J. Zhang, Zhao, & Gao, 2014) proposed the group based sparse restoration technique for restoring the image. It uses advance dictionary learning method that promotes the grouping of similar patches for achieving the task of image restoration. Hu et al. (Z. Hu, Huang, & Yang, 2010) proposed deblurring algorithm that exploits sparsity constraints of image patches and use over-complete dictionary that facilitate in recovering the latent image without solving an ill-posed deconvolution problem. The dictionary is learned and updated automatically without using additional images. Once the PSF is estimated, deblurring is performed to restore the degraded image.

The concept of reblurring the blurred image is adopted to obtain the edge information from the image. Li et al. (C. Li, Yuan, Bovik, & Wu, 2011) and Bong et al. (Bong & Khoo, 2014) showed how it can be used for image quality assessment. Li et al. (C. Li et al., 2011) method reblurs the test image intentionally and computes the local sample statistics in the vicinity of detected edges of the original and reblurred images, respectively. The statistical information is differenced and normalized to construct a blur index for quality score. Bong et al. (Bong & Khoo, 2014) used the concept of reblurring to calculate the differential statistics in terms of the local histograms. Further, Haibo et al. also used the concept of reblurring for the estimation of PSF parameter in which the problem was modeled as Gaussian low pass filtering and the PSF kernel is identified from it (H. Hu & De Hann, 2006).

Leida et al. proposed a no-reference image quality for blurred images based on the

use of Tchebichef moments which acts as effective shape descriptors (L. Li et al., 2015). The earlier work in (Thung et al., 2012), which showed that Tchebichef features, encodes the edge information efficiently, paved the way to propose a method to estimate the PSF parameters of a degraded image using Tchebichef moments as feature vectors. In the proposed work, the image is reblurred to obtain the relative edge information between the blurred and the reblurred image in terms of the Tchebichef features. These feature vectors are obtained by taking the difference of the Tchebichef moments of the blurred and the re-blurred images which are together used with the extreme learning machine (ELM) (Huang et al., 2006) to estimate the PSF parameters. Once the PSF is estimated, the split Bregman algorithm is used to restore the deblurred image and the quality of the restored image is evaluated using structural similarity (SSIM) index (Z. Wang, Bovik, Sheikh, & Simoncelli, 2004b) which is a method for measuring the similarity between two images.

The contributions and the overall layout of the chapter is clearly explained in the block diagram which is shown in Figure 5.1. It can be observed here that the Tchebichef moment based feature vectors are extracted from the blurred image. The feature vectors are used for training the extreme learning machine. Once the network is trained, it estimates the parameters of the PSF namely (σ, w) for the unknown blurred image. Using this information, split Bregman algorithm is adopted to restore the original image back. The rest of the chapter is organized as follows. In Section 5.2, preliminaries for Gaussian PSF model and Tchebichef moments are discussed. This is followed by the brief introduction on the necessity to classify the image into edge and plain blocks. Further, how Tchebichef moments are used as edge descriptors in this study. A detail explanation of the proposed algorithm is explained in Section 5.3. In Section 5.4, the experiment using cross-database validation from six publicly available databases is carried out to assess the performance of the proposed method to estimate the PSF parameters. A brief discussion about the

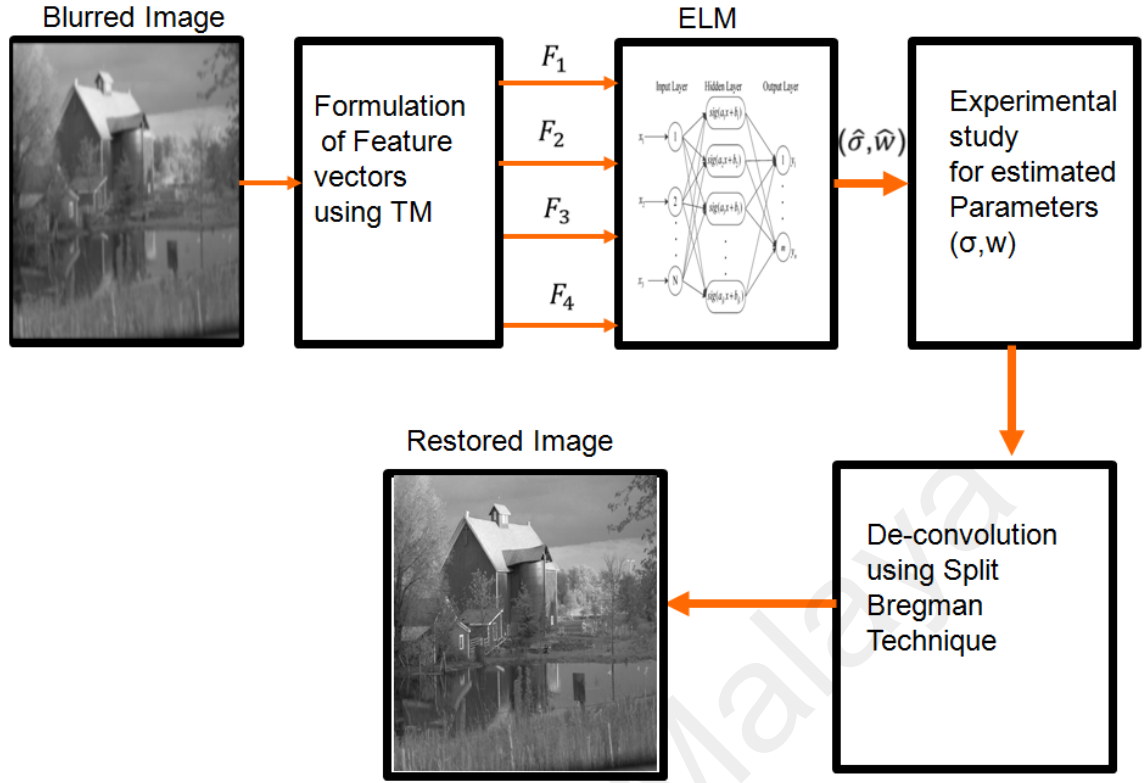


Figure 5.1: Flowchart of the proposed method

image restoration carried out using split Bregman algorithm is discussed in Section 5.5. A comparative analysis of the proposed method with the existing methods (Kotera et al., 2013; A. Goldstein & Fattal, 2012; G. Liu et al., 2014; Vera et al., 2013a; Yan et al., 2012) is carried out to verify the quality of the restored image using SSIM scores. Moreover, images are taken from their database to have a fair comparison with the proposed method. Further, two more methods (Vera et al., 2013a; Yan et al., 2012) that focuses on deblurring of the Gaussian blurred images are taken for comparison with the proposed method. This task is performed in Section 5.6. Section 5.7 gives the concluding remarks.

5.2 Formulation of feature vectors using Tchebichef moments (TM)

Gaussian based point spread function is the most common blur found in number of optical and image processing applications. In general, the Gaussian PSF is given as

$$h(x, y) = \begin{cases} \frac{1}{\sqrt{(2\pi)\sigma}} \exp\left(\frac{-(x^2+y^2)}{2\sigma^2}\right), & \text{if } (x, y) \in w. \\ 0, & \text{otherwise.} \end{cases} \quad (5.1)$$

where, σ is the standard deviation; w is the support size of the PSF. In general, w is represented by a matrix of size of $P \times P$. The parameters that govern the Gaussian PSF are the standard deviation (σ) and size (w). Proper estimation of these parameters are essential to have a good restored image quality. This chapter deals with the estimations of PSF parameters using Tchebichef moments and ELM.

The Tchebichef moment of order $(p+q)$ for an image with intensity function $f(x,y)$ defined in Chapter 2 (Section 2.3) is given as

$$T_{p,q} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \tilde{t}_p(x;N) \tilde{t}_q(y;N) f(x,y) \quad (5.2)$$

where, the image is of size $N \times N$ and $\tilde{t}_p(x;N)$ and $\tilde{t}_q(y;N)$ are the normalized Tchebichef polynomials defined as (Mukundan et al., 2001b)

$$\tilde{t}_p(x;N) = \frac{t_p(x;N)}{\sqrt{\rho(p,N)}}, \quad \tilde{t}_q(y;N) = \frac{t_q(y;N)}{\sqrt{\rho(q,N)}} \quad (5.3)$$

and $t_n(x;N)$ is the n th-order N -point Tchebichef polynomial defined as

$$t_n(x;N) = n! \sum_{k=0}^n (-1)^{n-k} \binom{N-1-k}{n-k} \binom{n+k}{n} \binom{x}{k} \quad (5.4)$$

For simplicity, the convention $\tilde{t}_n(x)$ is adopted to represent $\tilde{t}_p(x;N)$. The orthonormal version of Tchebichef polynomial $\tilde{t}_n(x)$ is calculated using recurrence relation as

$$\tilde{t}_n(x) = \alpha_1(2x+1-N)\tilde{t}_{n-1}(x) + \alpha_2\tilde{t}_{n-2}(x) \quad (5.5)$$

$$n = 2, 3, \dots, N-1; x = 0, 1, \dots, N-1$$

where

$$\alpha_1 = \frac{1}{n} \sqrt{\frac{4n^2-1}{N^2-n^2}} \quad (5.6)$$

$$\alpha_2 = \frac{1-n}{n} \sqrt{\frac{2n+1}{2n-3}} \sqrt{\frac{N^2-(n-1)^2}{N^2-n^2}} \quad (5.7)$$

The initial conditions for the recurrence relation given in (5.5) is given as

$$\tilde{t}_0(x) = \frac{1}{\sqrt{(N)}} \quad (5.8)$$

$$\tilde{t}_1(x) = (2x + 1 - N) \sqrt{\frac{3}{N(N^2 - 1)}} \quad (5.9)$$

Tchebichef moments in (5.2) can be represented in matrix form which is easily implemented in MATLAB. The Tchebichef moments for an image $\mathbf{X} = f(x, y)|_{x,y=0}^{N-1}$ up to the order $(p + q)$ can be implemented as

$$\mathbf{T} = \mathbf{P}\mathbf{X}\mathbf{Q}^T \quad (5.10)$$

where \mathbf{P} and \mathbf{Q} are the Tchebichef polynomials defined in matrix form as

$$\mathbf{P} = \begin{bmatrix} K_0(0) & \dots & K_0(N-1) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ K_n(0) & \dots & K_n(N-1) \end{bmatrix} \quad (5.11)$$

$$\mathbf{Q} = \begin{bmatrix} K_0(0) & \dots & K_0(N-1) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ K_m(0) & \dots & K_m(N-1) \end{bmatrix} \quad (5.12)$$

Tchebichef moments of an image \mathbf{X} are basically the projections of the image on to the basis functions ϕ_{pq} , which is the product of \tilde{t}_p and \tilde{t}_q defined as

$$\tilde{\mathbf{t}}_p = [\tilde{t}_p(0) \tilde{t}_p(1) \dots \tilde{t}_p(N-1)] \quad (5.13)$$

$$\tilde{\mathbf{t}}_q = [\tilde{t}_q(0) \tilde{t}_q(1) \dots \tilde{t}_q(N-1)] \quad (5.14)$$

$$\phi_{pq} = [\tilde{\mathbf{t}}_p]^T \tilde{\mathbf{t}}_q \quad (5.15)$$

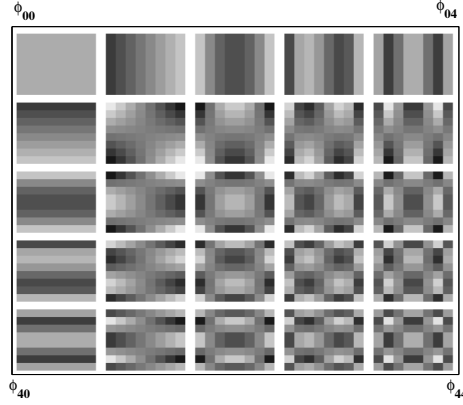


Figure 5.2: Basis function ϕ_{pq} for Tchebichef moment

A complete set of basis image ϕ_{pq} for a 8×8 image block is shown in Figure 5.2. The Tchebichef moments T_{pq} measures the correlation between the basis image ϕ_{pq} and the image \mathbf{X} .

5.2.1 Tchebichef moments as edge block descriptor

For image processing applications features of an image play an important role. As the proposed work is in estimating the PSF parameter of the degraded image, one can explore the possibility of using Tchebichef features in achieving this task. It is well known fact that blurring of an image affects its edges most when compared to the plain or texture regions. Therefore, the selection of feature vectors should be in such a way that it can capture the edges in the presence of blur and be able to help in estimating the PSF parameter from the degraded image. This chapter explores the use of Tchebichef features for the estimation of PSF parameter. From Figure 5.2 it can be observed that the basis image vector $\phi_{pq}(v) = [\phi_{01}, \phi_{02}, \dots, \phi_{08}]$ comprises of mostly the vertical edges and $\phi_{pq}(h) = [\phi_{10}, \phi_{20}, \dots, \phi_{80}]$ consist of mostly the horizontal edges. Therefore, the Tchebichef moments T_{pq} will give a high positive value of correlation with $\phi_{pq}(v)$ or $\phi_{pq}(h)$ if there are strong vertical or horizontal edges present in an image. Therefore, the feature vectors $\mathbf{F}_h = [T_{10}, T_{20}, T_{30}, T_{40}]$ and $\mathbf{F}_v = [T_{01}, T_{02}, T_{03}, T_{04}]$ are selected to

characterize the edges of an image. To understand the features \mathbf{F}_h and \mathbf{F}_v better their behavior on various kinds of edges is studied under varying PSF parameter σ . For this some practical edges are selected that are normally found in images and perform the following experiments as follows:

A typical vertical edge image block found in most of the images is shown in Figure 5.3(a). Here, one can study the effect of Gaussian PSF parameter σ applied to this image block and observe the corresponding behavior of Tchebichef moments T_{pq} which are used as components of the feature vector \mathbf{F}_h and \mathbf{F}_v . As discussed earlier, the correlation will be high with the basis image vector $\phi_{pq}(v)$ and will be low for $\phi_{pq}(h)$. Hence, the components of the feature vectors \mathbf{F}_v will be of large magnitude when compared to \mathbf{F}_h as shown in Figs. 5.3(b)-(c). The plot shows the variation of T_{pq} which are used as components of the feature vectors \mathbf{F}_h and \mathbf{F}_v with σ respectively.

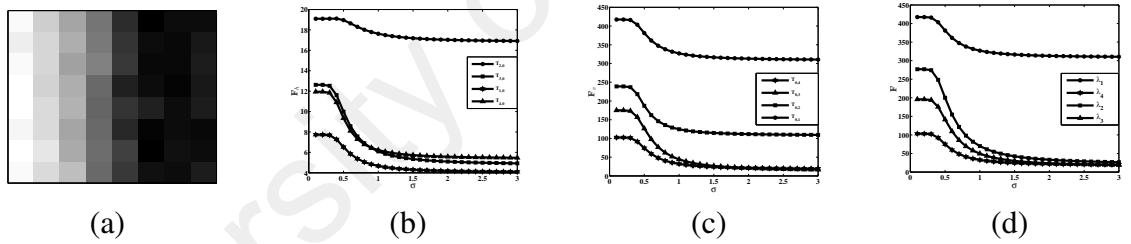


Figure 5.3: (a) Vertical edge block of size 8×8 (b)-(d) T_{pq} variation with σ

Further, Figure 5.4(a) shows an example of a horizontal edge image block. Here, the basis image vector $\phi_{pq}(h)$ has large correlation when compared to $\phi_{pq}(v)$. Hence, the magnitude of \mathbf{F}_h vector is large when compared to \mathbf{F}_v . The magnitude plot of the T_{pq} which are used as components of feature vector \mathbf{F}_h and \mathbf{F}_v is shown in Figs. 5.4(b)-(c) respectively. Similarly, Figure 5.5 shows the magnitude plot of the feature vectors for diagonal edges present in an image.

From observing Figs. 5.3-5.5 it can be concluded that depending on the horizontal or vertical edge the magnitude of the feature vector \mathbf{F}_h and \mathbf{F}_v varies. This makes the

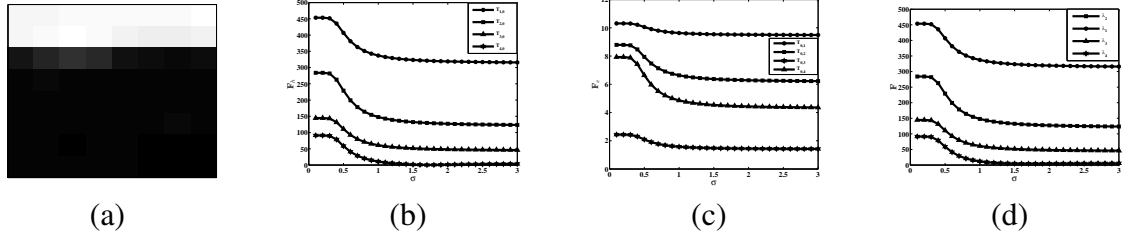


Figure 5.4: (a) Horizontal edge block of size 8×8 (b)-(d) T_{pq} variation with σ

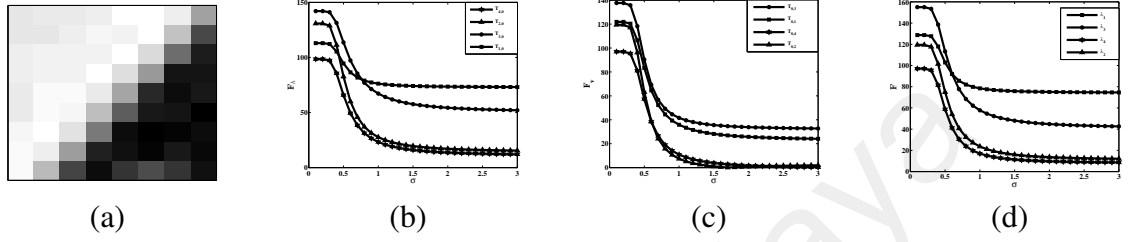


Figure 5.5: (a) Diagonal edge block of size 8×8 (b)-(d) T_{pq} variation with σ

components of feature vector exhibits large dynamic range. In order avoid this problem, the components of the feature vector \mathbf{F} are proposed as

$$\mathbf{F} = [\lambda_1, \lambda_2, \dots, \lambda_4] \quad (5.16)$$

where, λ_i is defined as

$$\lambda_i = \sqrt{T_{0i}^2 + T_{i0}^2}, \text{ for } i=1,2,\dots,4 \quad (5.17)$$

The dynamic range of the components (λ_i) of the feature vector \mathbf{F} is invariant to horizontal or vertical edges as can be seen from Figs. 5.3(d), 5.4(d) and 5.5(d). This feature vector will be used for training the ELM for estimating sigma (σ) and w .

5.2.2 Plain and Edge block classification

Since the Gaussian PSF affects the edges which the low order Tchebichef moments captures well, it is essential that the edge blocks of an image are identified. Therefore, this section presents a brief review of image block classification method (Thung et al., 2012). Here, the images are divided into non-overlapping 8×8 blocks which are transformed into moment domain using discrete Tchebichef transform. In general, there are two types

of blocks present in an image, namely plain and edge. Among them the plain block has the lowest moment energy compared to edge blocks. They proposed moment block energy for horizontal edge (HE) and vertical edge (VE) and diagonal edges (DE) as

$$HE = \sum_{i=1}^3 T_{i0}^2 \quad (5.18)$$

$$VE = \sum_{j=1}^3 T_{0j}^2 \quad (5.19)$$

$$DE = \sum_{i=1}^3 T_{0j}^2 + \sum_{j=1}^3 T_{i0}^2 \quad (5.20)$$

where T_{ij} represents the moment value of the i -th row and j -th column of the moment block. These values are normalized using SSM defined as

$$\begin{aligned} SSM &= \left(\sum_{i,j=0}^7 T_{ij}^2 \right) - T_{00}^2 \\ &= \sum_{i,j=0}^7 [I_b(i,j) - \bar{I}_b] \end{aligned} \quad (5.21)$$

Here, T_{00}^2 is removed as it corresponds to dc component of the image and contains no valuable information about the image content. It has been observed that plain block has lower moment energy when compared to edge block. Hence, the HE/E_{ij} , VE/E_{ij} are used as input features for the classifier proposed in (Thung et al., 2012) for the selection of edge block. This is required as the proposed algorithm relies on the selection of these edge blocks.

5.3 PSF parameters estimation via Extreme learning machine (ELM)

In this section the proposed algorithm to estimate the (σ, w) of the Gaussian PSF is discussed. Low order Tchebichef moments act as a good feature descriptor that can efficiently encode the information of the edges present in an image. In the proposed work, the image is reblurred to obtain the relative edge information between the blurred and the reblurred image in terms of the Tchebichef features. This is followed by the use

of ELM, which uses these feature vectors to estimating the PSF parameters sigma (σ) and size (w) respectively. The flowchart of the proposed algorithm to estimate the PSF parameters (σ, w) is given in Figure 5.6. The detail description of the algorithm is given as follows

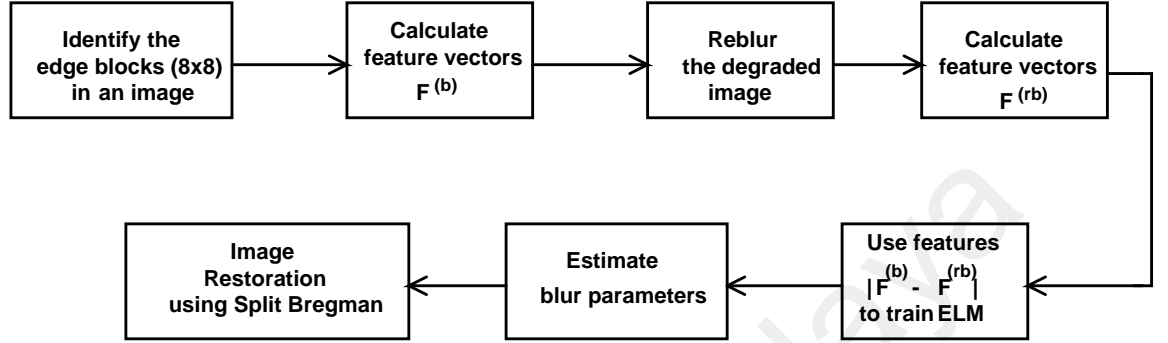


Figure 5.6: Flowchart of the proposed algorithm

1. Identify the edges and plain blocks in an image. Blurring of an image mostly affects the edges while keeping plain blocks are least affected. Hence, the algorithm will use the feature vectors that correspond to edge blocks rather than plains to estimate the PSF parameters. The size of the edge block is 8×8 .
2. Once the edge blocks are identified, the mean of feature vector $F^{(b)}$ is calculated for all the edge blocks given as

$$F^{(b)} = \frac{1}{M} \sum_{j=1}^M F_j \quad (5.22)$$

where, M is the total number of edge blocks in an image and F is the feature vector evaluated using (5.16).

3. Reblur the degraded image.
4. Calculate the re-blurred feature vector $F^{(rb)}$ by repeating step (2).
5. Use the feature vector $|F^{(b)} - F^{(rb)}|$ for training the ELM.

6. Once the ELM is trained, the estimated PSF parameters (σ, w) are used in the split Bregman based image restoration technique to restore back the original image.

To ensure the better generalization ability of the ELM network, the optimal selection of the hidden neurons, the corresponding input weights (W) and bias values (B) are required. One way of achieving this is by randomly selecting the fixed input weights and the bias values, the optimal output weights are evaluated using least square approach. However, the generalization performance of the ELM depends on the proper selection of the hidden neurons, input weights and bias values. A selection of these parameters is crucial for proper generalization by ELM. In the proposed work, one can consider ELM network with 80 hidden neurons. ELM algorithm is called 500 times for the same and cross database training/testing data and finds the mean and variance of the testing and training accuracies. Each time the ELM is called, the fixed parameters, namely weights (W) and bias (B) are initialized randomly from a uniform distribution. The input feature vectors are normalized between 0 to 1 and the weights (W) and bias (B) are initialized between ± 1 . Here, one can opt for unipolar sigmoidal activation function for the hidden neurons. The slope of the sigmoidal function is selected as 0.1 which is approximately equal to the number of input neurons.

In the case of testing and training from the same database, i.e. in this case LIVE database; the mean and the standard deviation of training efficiency are 93.13 and 0.0953 respectively. Similarly, the mean and standard deviation for testing efficiency are 91.26 and 0.0196 respectively. This is shown in Figure 5.7 for different runs. Further for the cross-database analysis shown in Figure 5.8, where training is performed using Berkeley dataset and testing using LIVE dataset; the mean and the standard deviation of training efficiency are 98.38 and 0.0121 respectively. Similarly, the mean and standard deviation for testing efficiency are 97.93 and 0.0178 respectively. From the Figs. 5.7-5.8, it can be

seen that the random selection of the fixed parameters does not affect the generalization performance of the ELM regressor for the proposed work.

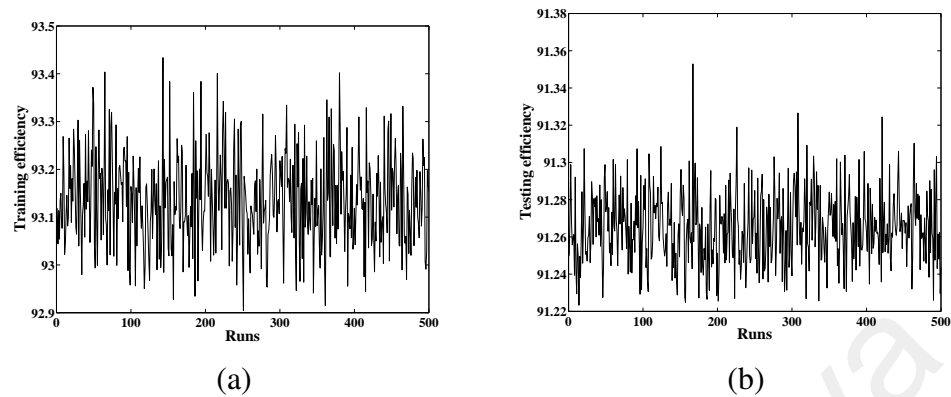


Figure 5.7: Effects of the initial parameter selection on training and testing performance for same database.

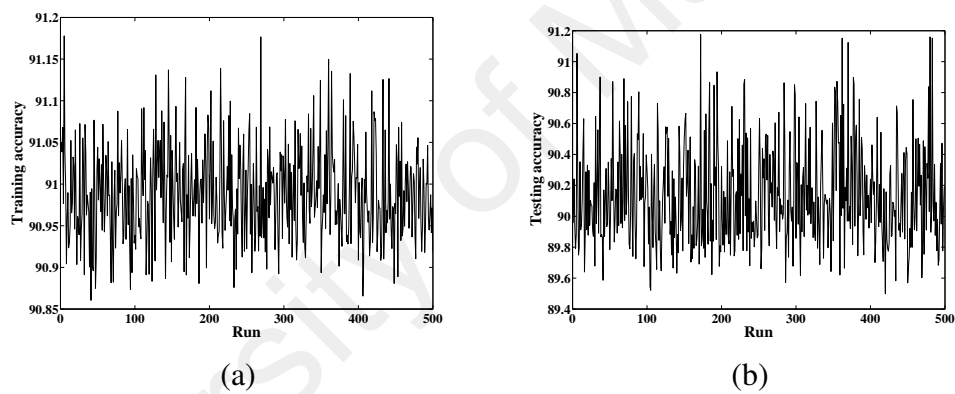


Figure 5.8: Effects of the initial parameter selection on training and testing performance for cross database.

Moreover, the generalization of the ELM depends on the number of hidden neurons. To illustrate this behavior, an experimental study is conducted by varying the hidden neurons from 20 to 100 with the steps of 10. The variation of the testing and training efficiency for the same and cross database analysis is shown in Figure 5.9 and 5.10 respectively. From Figure 5.9 it can be observed that the training and testing efficiency do not vary much with the variations of the hidden number of neurons and initial parameter changes. In specific for the LIVE database under consideration, the training efficiency increases by a small amount from 93.6 to 94.2 as the hidden neurons increased from 20 to

100, which is not significant. The testing efficiency, however remains nearly constant to 91.2. Similarly, Figure 5.10 shows the training and testing efficiency for cross database analysis (Berkeley for training and LIVE for testing). It can be observed that both the testing and training accuracy remains nearly constant to 98.38 and 97.9 respectively, for the variation in the number of hidden neurons and initial parameter settings. These trends hold true for all the databases considered in the proposed work for both individual and cross database analysis.

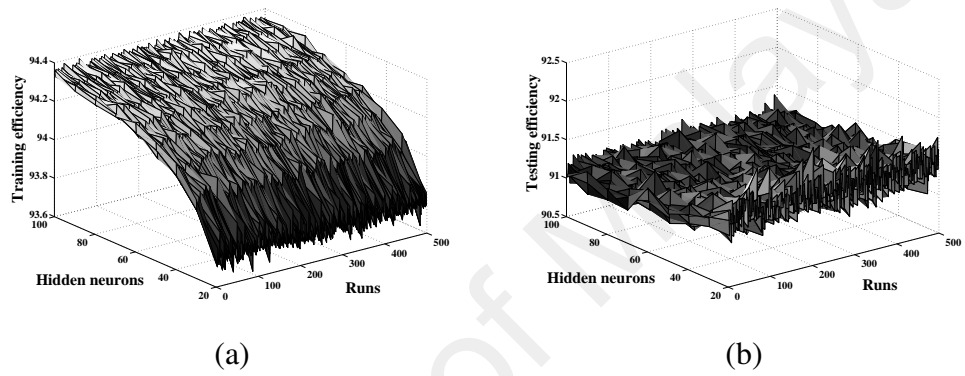


Figure 5.9: (a) Training (b) Testing accuracy variation with respect to hidden neurons and initial parameters for same database

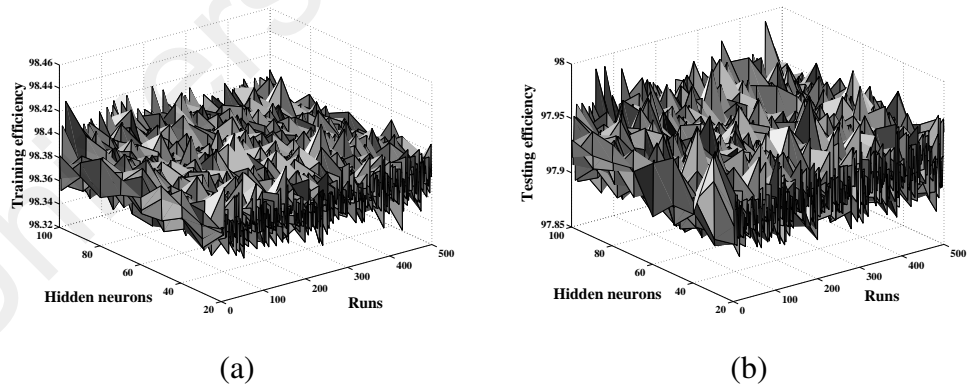


Figure 5.10: (a) Training (b) Testing accuracy variation with respect to hidden neurons and initial parameters for cross database

5.4 Experimental study for the estimated PSF parameters

In this section, three experiments are carried out to validate the performance of the proposed method on six different standard databases. In the first experiment, the proposed

method is trained and tested using the same database. In the second experiment, the proposed method is trained using all the images in one database and tested on the remaining five database images. This is repeated for all the databases under consideration. In the third experiment the application of the proposed method in image restoration using a split Bregman algorithm is carried out where the quality of the restored image is evaluated in terms of SSIM scores. Further, the comparative analysis with the three existing methods is also performed.

5.4.1 Data description

In this study, images are taken from six publicly accessible databases for training and testing of the proposed algorithm in order to estimate the PSF parameters: LIVE (Sheikh, Wang, et al., 2005), CSIQ (Larson & Chandler, 2010), CIDIQ (*CIDIQ*, n.d.), VCL (Zarić et al., 2012), Caltech (Fink, 2003) and Berkeley (Martin et al., 2001). Here, original images from all the databases are used except for Caltech and Berkeley, where one can randomly selected 34 and 32 images respectively. The images in all the databases vary in terms of its content and size. These images are successively degraded using Gaussian PSF with varying sigma (σ) and size (w) thus producing a total of 38,988 images to be used for training and testing of the proposed algorithm. The range of sigma (σ) varies from 0.3 to 4 in steps of 0.1 and size (w) from 7x7 to 17x17 in steps of 2x2. The details of the original images present in all the databases along with the number of images used for training and testing are shown in Table 5.1.

Table 5.1: Details of testing and training samples for various databases

Database	Original Images	Total Images	Training	Testing
LIVE	29	6612	4600	2012
CSIQ	30	6840	4788	2052
CIDIQ	23	5244	3670	1574
VCL	23	5244	3670	1574
Caltech	34	7752	5425	2327
Berkeley	32	7296	5107	2189

5.4.2 Significance of using edge blocks in PSF estimation

This section shows the importance of using only the edge blocks for the estimation of PSF parameters using the proposed algorithm. As blurring of the image affects the edges present in an image, hence, edge block provides useful information about the PSF parameters. In order to justify the dependence of the proposed algorithm on the edges present in an image, four images are selected from the LIVE database with varying content as shown in Figure 5.11. These images are selected based on the percentage of edges present in them (Thung et al., 2012).

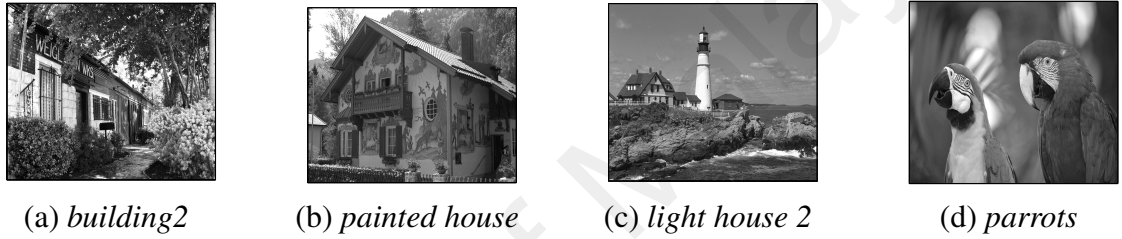


Figure 5.11: Test Images taken from LIVE database

Table 5.2 shows the detail breakdown of the selected images into edge and plain blocks. Images are arranged in Table 5.2 with decreasing contribution of edges. This is done to observe the performance of the proposed algorithm on images with varying edge contribution.

Table 5.2: Classification of LIVE database images based on image content

Images	Plain	Edges
<i>Building2</i>	3%	97%
<i>painted house</i>	31%	69%
<i>light house 2</i>	51%	49%
<i>parrots</i>	85%	15%

Two experiments are conducted which are based on cross database evaluation i.e. Berkeley dataset (Martin et al., 2001) with 7296 images are used for training the ELM while testing is performed using LIVE (Sheikh, Wang, et al., 2005) database of 4 images mentioned in Table 5.2. In the first experiment, only the edge blocks of the image are

considered for training while in the second experiment, both the plain and edge blocks of the image are used for training the ELM. The estimation of the PSF parameters (σ, w) obtained using ELM for both the experiments are shown in Tables 5.3 and 5.4.

The estimation of σ for both the experiments along with their relative errors defined as $(x_0 - x)/x$ where x_0 is the measured and x is the true value is shown in Table 5.3. It can be observed that the edge blocks play an important role in the estimation of σ as can be seen from the third column of Table 5.3, where the algorithm closely estimates the actual sigma with a small relative error. However, the fifth column shows the results of the second experiment where the estimation of σ is not accurate when compared to the first experiment as can be observed from their relative errors. Hence, it can be concluded that edge blocks play an important role in the proposed algorithm for the estimation of σ .

Table 5.3: Role of edge block in estimating the PSF sigma (σ)

Images	σ	Edges		Edges +Plains		Images	σ	Edges		Edges+Plain	
		Estimated σ	Relative error	Estimated σ	Relative error			Estimated σ	Relative error	Estimated σ	Relative error
building2	0.5	0.51	0.02	1.51	2.02	paintedhouse	0.5	0.52	0.04	1.12	1.24
	1.0	0.99	0.01	2.14	1.14		1.0	1.05	0.05	2.54	0.54
	1.5	1.54	0.02	2.23	0.48		1.5	1.48	0.01	1.23	0.18
	2.0	2.07	0.03	2.87	0.43		2.0	2.08	0.04	3.22	0.61
	2.5	2.49	0.004	1.76	0.29		2.5	2.48	0.008	1.32	0.47
	3.0	3.11	0.03	3.95	0.31		3.0	3.12	0.04	4.13	0.37
	3.5	3.51	0.002	2.84	0.18		3.5	3.49	0.002	2.81	0.19
	4.0	4.11	0.02	4.52	0.13		4.0	4.15	0.03	3.21	0.19
lighthouse2	0.5	0.52	0.04	1.24	1.48	parrots	0.5	0.54	0.08	1.83	2.66
	1.0	1.04	0.04	0.78	0.22		1.0	1.07	0.07	0.32	0.68
	1.5	1.54	0.02	1.02	0.32		1.5	1.63	0.08	2.56	0.70
	2.0	1.89	0.05	1.32	0.34		2.0	2.19	0.09	1.67	0.17
	2.5	2.45	0.02	3.65	0.46		2.5	2.34	0.06	3.26	0.30
	3.0	3.16	0.05	2.15	0.28		3.0	3.26	0.08	2.24	0.25
	3.5	3.65	0.04	4.12	0.17		3.5	3.78	0.09	4.32	0.23
	4.0	4.16	0.04	3.23	0.19		4.0	3.67	0.08	3.27	0.182

A similar experiment to estimate the PSF size (w) is carried out and shown in Table 5.4. It can be observed that by considering the edge blocks of an image, the proposed algorithm closely estimates the PSF size. On the other hand the error increases in the estimation of PSF size (w) if both the edge and plain blocks are considered.

Table 5.4: Role of edge block in estimating the PSF size (w)

Images	(w)	Edges	Edges +Plains	Images	(w)	Edges	Edges+Plain
		Estimated (w)	Estimated (w)			Estimated (w)	Estimated (w)
<i>building2</i>	5×5	7×7	11×11	<i>paintthouse</i>	5×5	5×5	13×13
	7×7	5×5	3×3		7×7	7×7	11×11
	9×9	11×11	13×13		9×9	11×11	3×3
	11×11	11×11	17×17		11×11	13×13	17×17
	13×13	15×15	7×7		13×13	11×11	9×9
	15×15	15×15	9×9		15×15	15×15	7×7
	17×17	17×17	11×11		17×17	15×15	9×9
<i>lighthouse2</i>	5×5	5×5	13×13	<i>parrots</i>	5×5	5×5	7×7
	7×7	5×5	3×3		7×7	7×7	11×11
	9×9	7×7	3×3		9×9	11×11	15×15
	11×11	11×11	17×17		11×11	11×11	5×5
	13×13	15×15	9×9		13×13	13×13	7×7
	15×15	17×17	9×9		15×15	13×13	7×7
	17×17	17×17	7×7		17×17	15×15	9×9

5.4.3 Training and testing using same databases

Estimating the PSF parameters is essential to obtain a good quality image, for which ELM is used (Section 4.3). Here, the evaluation is done using 10 runs. In this validation scheme, for each run, a random partition is done by selecting 70% of the data for testing while the remaining 30% as training. This is motivated by a similar method adopted in recent state of the art methods for cross validation analysis of the estimated parameters (Mittal, Soundararajan, & Bovik, 2013; Mittal et al., 2012; Xie, Lu, Bovik, Jiang, & Meng, 2016). Once each run is done, the performance of the regression analysis is carried in terms of correlation coefficient (CC). The task of partitioning the data and its CC evaluation is repeated for all the 10 runs. Once this is done, the average accuracy which is the mean of the CC values obtained for each run is reported. To validate the proposed method for the PSF estimation, all six databases are used. The Tchebichef features $|F^b - F^{rb}|$ are calculated for each of the blurred images using the proposed algorithm. Once the ELM is trained, the performance of the proposed method is evaluated from the same database. Tables. 5.5 and 5.6 show the scatter plot for the estimated value of σ and w respectively for different databases. From Table 5.5, it can be observed that the correlation coefficients (CC) values for all the databases are in the range

of 0.9712(VLC) to 0.9934(CSIQ). Similarly for Table 5.6 the CC values obtained are in the range of 0.9348(CIDIQ) to 0.9883(Caltech).

Table 5.5: Estimation of PSF parameter σ for various databases

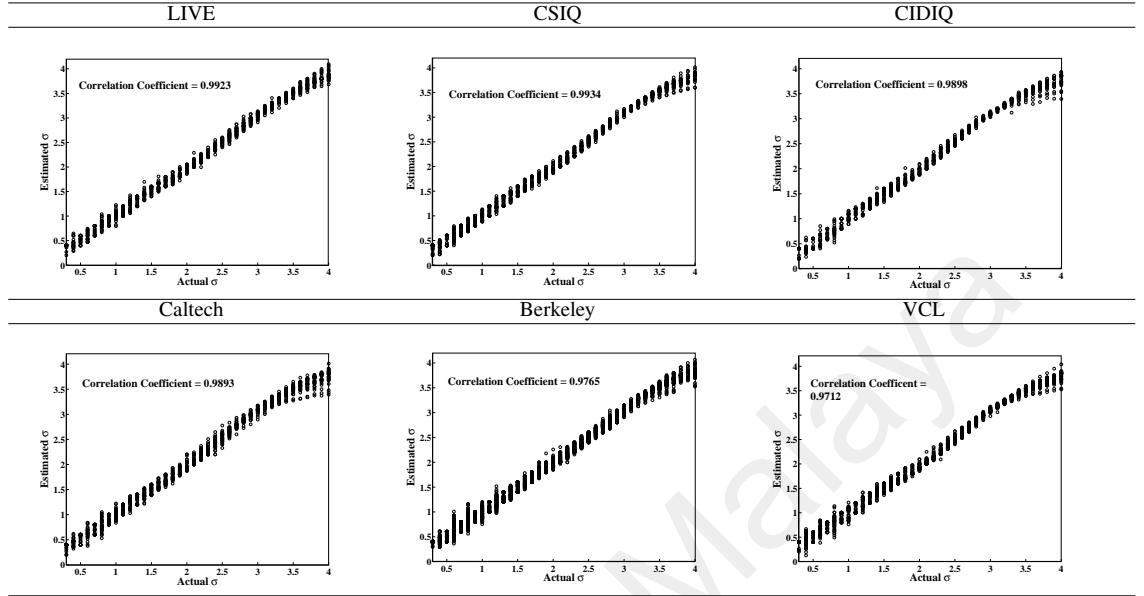
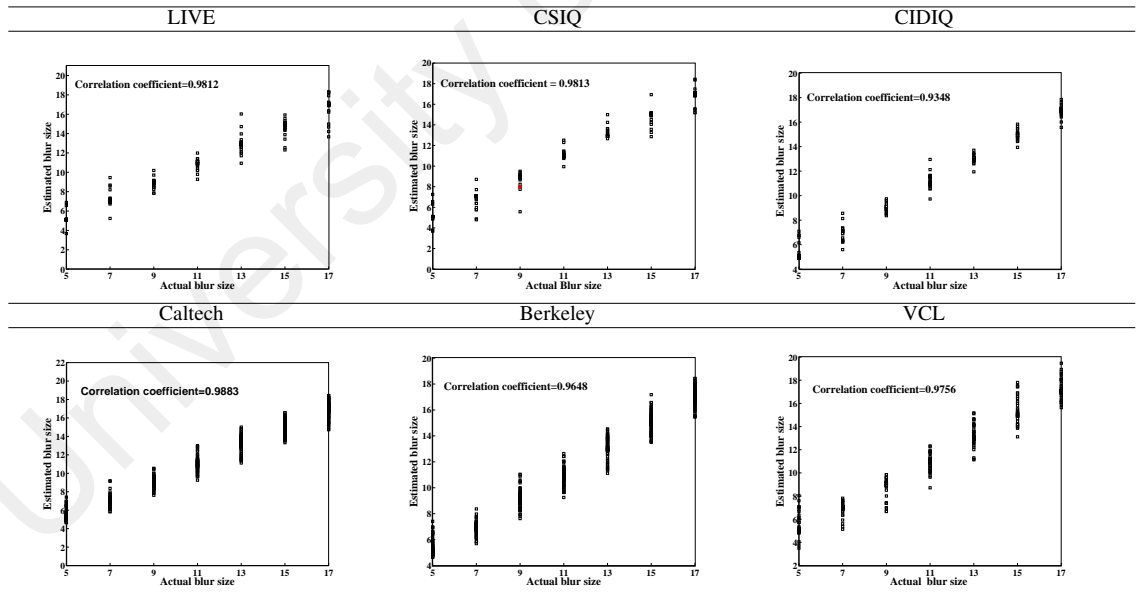


Table 5.6: Estimation of PSF parameter (w) for various databases



5.4.4 Cross-database validation

A way to determine the generality of a machine-learning based PSF parameter estimator is the cross-database validation since images and/or distortions vary across databases. For the cross-database testing, the images from one database are used for training the

ELM and tested using the images from the other remaining databases. Table 5.11 provides the correlation coefficient (CC) values of the regression results of estimating sigma (σ) for all the possible combinations of datasets. For instance, the first row provides the CC values of the regression results when training is performed using VCL database and tested using other databases.

Table 5.7: Correlation coefficient (CC) values for cross database performance of estimating PSF's sigma (σ).

Training	Testing					
	VCL	Berkeley	LIVE	Caltech	CSIQ	CIDIQ
VCL	-	0.8521	0.9235	0.8157	0.9178	0.8612
Berkeley	0.8527	-	0.8658	0.8045	0.8812	0.8478
LIVE	0.8712	0.8134	-	0.8268	0.8537	0.8147
Caltech	0.8257	0.8267	0.8145	-	0.8512	0.7834
CSIQ	0.8124	0.8356	0.8856	0.7862	-	0.8234
CIDIQ	0.8178	0.8024	0.7945	0.8345	0.9078	-

In this case, the correlation coefficients (CC) values of LIVE and CSIQ databases are 0.9235 and 0.9178 respectively, while Berkeley, CALTECH and CIDIQ provide CC values of 0.8521, 0.8157 and 0.8612 respectively. It can be observed that LIVE and CSIQ databases provide better prediction accuracies compared to other databases. Similarly, when Berkeley database is used for training, the CC scores registered for LIVE and CSIQ is better compared to other databases. From this it can be observed that the proposed algorithm can estimate the PSF parameter (σ) well as indicated by the CC values. Similarly, Table 5.7 provides the CC values for the estimation of size (w). For VCL database the CC values of LIVE and CSIQ databases are 0.9335 and 0.9034 respectively, while Berkeley, CALTECH and CIDIQ provide CC values of 0.8921, 0.8412 and 0.8367 respectively. In this case also LIVE database provides better prediction accuracies compared to other databases. The same holds true when Berkeley, CSIQ and CIDIQ are used as training databases.

Table 5.8: Correlation coefficient (CC) values for cross database performance of estimating PSF's size (w).

Training	Testing					
	VCL	Berkeley	LIVE	Caltech	CSIQ	CIDIQ
VCL	-	0.8921	0.9335	0.8412	0.9034	0.8367
Berkeley	0.8521	-	0.8834	0.8436	0.8523	0.8723
LIVE	0.8265	0.9025	-	0.8425	0.8932	0.8451
Caltech	0.8425	0.8702	0.8623	-	0.8726	0.8266
CSIQ	0.8325	0.8627	0.9145	0.8568	-	0.7923
CIDIQ	0.8046	0.8315	0.9056	0.8734	0.8923	-

5.5 Image restoration using split Bregman algorithm

Once the estimation of PSF is done using the proposed algorithm, image restoration is performed to deblur the image using the existing optimization technique known as split Bregman as shown in Figure 5.12. The quality of the restored image is compared with three existing methods: Kotera et al. (Kotera et al., 2013), Goldstein et al. (A. Goldstein & Fattal, 2012) and Liu et al. (G. Liu et al., 2014). These methods require the information of the PSF to restore the deblurred image by adopting different varieties of optimization techniques. In particular, Kotera et al. (Kotera et al., 2013) proposed the deconvolution algorithm that employs maximum a posterior estimation combined with sparse priors and advance numerical methods to restore the degraded image. Goldstein et al. (A. Goldstein & Fattal, 2012) proposed a method for estimating the PSF using statistical irregularities in their power spectrum. The power law model together with spectral whitening is used to estimate the power spectrum of the PSF. Later, PSF is recovered using a phase retrieval algorithm. Further, the task of image restoration is performed using Bregman technique. Liu et al. (G. Liu et al., 2014) proposed the method of PSF estimation by analyzing and comparing the spectrum of an image as a convolution operator that changes before and after blurring operation. Once the estimation is done, deblurring is performed using the technique discussed in Krishnan et al. (Krishnan & Fergus, 2009).

Here, the details of the Split Bregman algorithm used for solving the general image

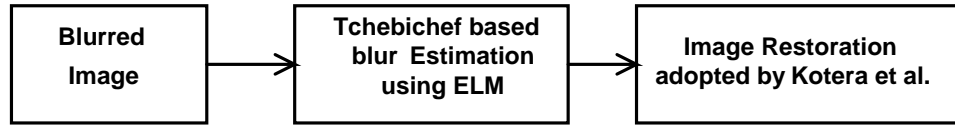


Figure 5.12: Proposed restoration method

restoration problems is discussed which appears in the following form

$$\hat{f} = \underset{f}{\operatorname{argmax}} \frac{1}{2} \|g - Hf\|_2^2 + \beta \|f\|_1 \quad (5.23)$$

where g is the observed blurred image, H and \hat{f} are the estimated PSF and original image respectively. Further, $\|(\cdot)\|_2^2$ and $\|(\cdot)\|_1$ are the L_2 and L_1 norms respectively. The L_1 norm of a vector $\mathbf{x} = [x_1, x_2, x_3, \dots, x_N]$ of length N is defined as

$$\|\mathbf{x}\|_1 = \sum_{j=1}^N |x_j| \quad (5.24)$$

while L_2 norm is defined as

$$\|\mathbf{x}\|_2^2 = \sum_{j=1}^N |x_j|^2 \quad (5.25)$$

The L_1 norm involved in (5.23) is non-smooth and non-separable which makes it difficult for the optimization methods to obtain the optimal solution. A natural idea to solve (5.23) is to use a smoothed L_1 norm to approximate the actual L_1 norm and then apply optimization methods for smooth functions. This approach was commonly used in total variation methods earlier. However, the better the approximation is to L_1 norm, the algorithm will converge slower. Therefore, in order to make this algorithm converge faster, smoother approximation to L_1 norm is not the proper solution. Another difficulty arises due to non-separability issue of the priors involved in (5.23). To overcome these issues of non-separable and discontinuous behavior the method of split Bregman algorithm is proposed. The split Bregman algorithm for (5.23) is demonstrated to be an efficient tool for solving the problems arising from total variation norm minimization problems for image restoration such as image deblurring. The optimization problem given in (5.23) can be solved by the split Bregman algorithm which was proposed by (T. Goldstein & Osher,

2009). Here, the steps involved in solving the optimization problem using split Bregman are carried out. In particular, the estimate of the original image \hat{f} given in (5.23) is obtained by solving the following sub-problems

$$\hat{f}_i \leftarrow \operatorname{argmax}_f \frac{1}{2} \|g - Hf\|_2^2 + \frac{\lambda}{2} \|f - u_{i-1} + b_{i-1}\|_2^2 \quad (5.26)$$

$$u_i \leftarrow \operatorname{argmax}_u \beta \|u\|_1 + \frac{\lambda}{2} \|u - \hat{f}_i - b_{i-1}\|_2^2 \quad (5.27)$$

$$b_i \leftarrow b_{i-1} + (\hat{f}_i - u_i) \quad (5.28)$$

where b is the Bregman relaxation variable. Here the equations (5.26)-(5.28) are iterated recursively for a given number of iterations i . These equations as a whole constitutes the split Bregman algorithm. The sub-problem defined in (5.26) is easy to solve as it involves least square solution. If information about H is available one can employ the conjugate gradients method to solve the equations. Further, (5.27) is an L_1 optimization problem that can be solved easily using iterative threshold methods (Fadili & Starck, 2006). The final step shown in (5.28) is to update the Bregman relaxation variable. Further details of the algorithm can be found in (T. Goldstein & Osher, 2009).

5.6 Comparative analysis with the existing state of the art methods

In this study, two experiments are performed using the blurred images of the LIVE database to evaluate the performance of the proposed method with the three existing methods. In the first experiment four images from the LIVE database that vary in image content as shown in Table 5.2 are used to evaluate the visual quality in terms of SSIM scores. Further, visual quality of the two images is shown to demonstrate the effectiveness of the proposed approach. In the second experiment, a comparative analysis of the 29 original images blurred with specific pairs of (σ, w) is shown in terms of SSIM scores. Further, analysis of the proposed method with the images used in the existing methods is carried out to have a fair comparison in terms of SSIM scores.

5.6.1 Experiment 1

Table 5.9 presents the first experiment where the comparison of the proposed method with the existing ones in terms of the SSIM score is carried out. Here, the cross database evaluation is performed using the Berkeley database as training and LIVE database images for testing. The second and third column of the table shows the actual and the estimated PSF parameters. It can be observed that the proposed method estimates these parameters closely. Further, the SSIM scores of the proposed method along with the existing methods are given for the test images. The score provided in bold shows the maximum score. It can be seen that proposed method gives better SSIM scores when compared to the existing methods for blurred images caused by sigma (σ) less than 2.0. For sigma (σ) greater than 2.0 the proposed method in most cases performs better than the existing methods.

Table 5.9: Comparative analysis with the existing methods in terms of SSIM scores

Images	Actual (σ, w)	Estimated (σ, w)	SSIM			
			Proposed	(Kotera et al., 2013)	(G. Liu et al., 2014)	(A. Goldstein & Fattal, 2012)
<i>building2</i>	(0.5,7)	(0.51, 5)	0.9742	0.9645	0.9567	0.9357
	(1.0,9)	(0.99,11)	0.9512	0.9354	0.9234	0.9134
	(1.5,11)	(1.54,11)	0.9132	0.8956	0.8867	0.8678
	(2.0,11)	(2.07,11)	0.8876	0.8582	0.8478	0.8356
	(2.5,13)	(2.49,15)	0.8612	0.8476	0.8354	0.8112
	(3.0,13)	(3.11,15)	0.8329	0.8251	0.8121	0.8092
	(3.5,15)	(3.51,15)	0.8113	0.7989	0.7923	0.7843
	(4.0,17)	(4.11,17)	0.8067	0.8145	0.7767	0.7512
<i>lighthouse2</i>	(0.5,7)	(0.52,5)	0.9736	0.9631	0.9434	0.9315
	(1.0,9)	(1.04,7)	0.9331	0.9236	0.9125	0.8926
	(1.5,11)	(1.54,11)	0.9223	0.9175	0.9067	0.8824
	(2.0,11)	(1.89, 11)	0.9165	0.9031	0.8882	0.8752
	(2.5,13)	(2.45, 15)	0.8678	0.8569	0.8361	0.8318
	(3.0,13)	(3.16, 17)	0.8354	0.8236	0.8153	0.7925
	(3.5,15)	(3.65, 17)	0.8143	0.8371	0.8049	0.7851
	(4.0,17)	(4.16, 17)	0.7832	0.8178	0.7843	0.7721
<i>paintedhouse</i>	(0.5,7)	(0.52, 7)	0.9832	0.9643	0.9587	0.9432
	(1.0,9)	(1.05, 11)	0.9456	0.9303	0.9278	0.9272
	(1.5,11)	(1.48, 11)	0.9165	0.9015	0.8996	0.8732
	(2.0,11)	(2.08, 11)	0.8856	0.8784	0.8578	0.8341
	(2.5,13)	(2.48, 13)	0.8621	0.8697	0.8486	0.8271
	(3.0,13)	(3.12, 15)	0.8556	0.8612	0.8265	0.8072
	(3.5,15)	(3.49, 15)	0.8376	0.8359	0.7974	0.7745
	(4.0,17)	(4.15, 17)	0.8032	0.8118	0.7786	0.7632
<i>parrots</i>	(0.5,7)	(0.54, 7)	0.9692	0.9594	0.9487	0.9332
	(1.0,9)	(1.07, 11)	0.9465	0.9364	0.9212	0.9173
	(1.5,11)	(1.63, 11)	0.9367	0.9252	0.8996	0.8732
	(2.0,11)	(2.19, 11)	0.9132	0.9013	0.8778	0.8541
	(2.5,13)	(2.34, 13)	0.8732	0.9108	0.8486	0.8271
	(3.0,13)	(3.26, 13)	0.8456	0.8381	0.8265	0.8072
	(3.5,15)	(3.78, 13)	0.8212	0.8280	0.7974	0.7745
	(4.0,17)	(3.67, 15)	0.7897	0.8167	0.7786	0.7632

Two images from the Table 5.9 are selected to show the visual comparison of the restoration performed using proposed and the existing methods. The two images *parrots*

and *buliding2* are selected because of their varying edge content. The *building2* image has the largest number of edge blocks while the image *parrot* has the lowest number of edge blocks. Figure 5.13 shows the visual comparison of the image restoration performed on the parrots image. Here, the image is Gaussian blurred using $\sigma = 2.5$ and $w = 13$. It also shows the estimated PSF obtained using the proposed and the existing methods. It can be observed that the quality of the restored image is comparable for both the proposed and the method in (Kotera et al., 2013). One possible reason for the difference in the quality of the restored images with the other two methods in (A. Goldstein & Fattal, 2012; G. Liu et al., 2014) is the close estimation of the PSF from the degraded image. Any discrepancy in the estimation of the PSF will result in the degradation of the image quality. The PSF estimated using proposed method and the method in (Kotera et al., 2013) is quite close to the PSF with which the original image is distorted. This is validated by zooming into the details of the parrots image shown in Figure 5.14. Here, the sharp features near the eyes of the parrots are restored effectively using both the proposed and method in (Kotera et al., 2013). Similar observations and conclusions can be made in the case of building2 image which is Gaussian blurred using $\sigma = 1.5$ and $w = 11$ as shown in Figure 5.15. Further, Figure 5.16 shows the zoomed region of the highlighted section of the image. It can be observed that the texture of the wall is well restored using the proposed method when compared to the existing methods.

5.6.2 Experiment 2

In this experiment, Figure 5.17 presents the detailed comparative analysis of the proposed method with the existing ones in terms of the SSIM score for all the 29 LIVE database images. The analysis is based on cross database evaluation performed using the Berkeley database images for training and LIVE database images for testing. For instance, Figure 5.17(a) shows the SSIM score for all the restored 29 test images of LIVE

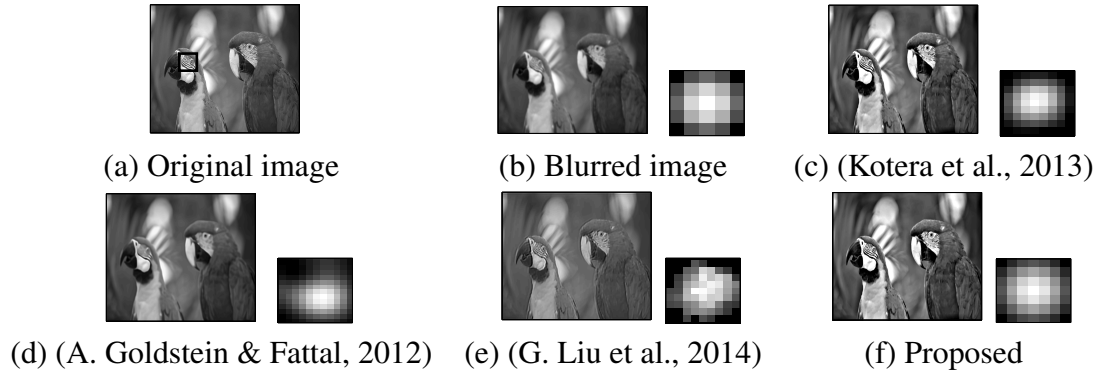


Figure 5.13: Comparison of the deblurring results on *parrots* image blurred with Gaussian PSF of $\sigma=2.5$ and $w = 13$

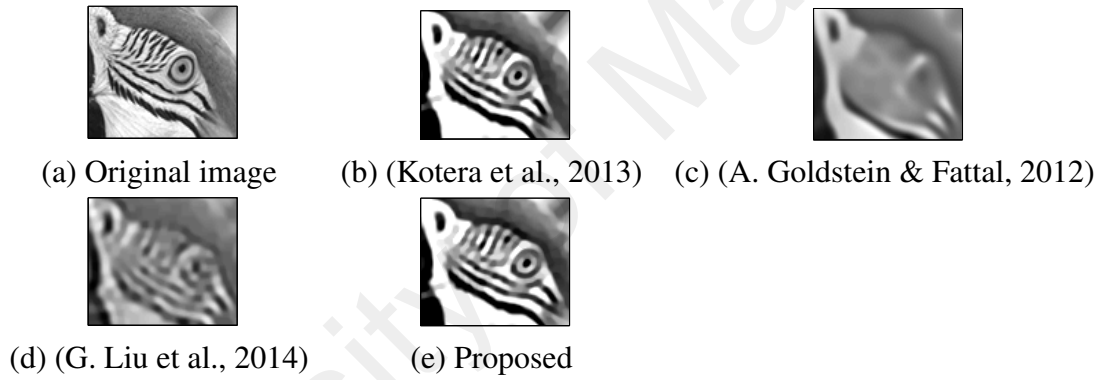


Figure 5.14: Zooming the details of *parrots* image for observing the local edge restoration

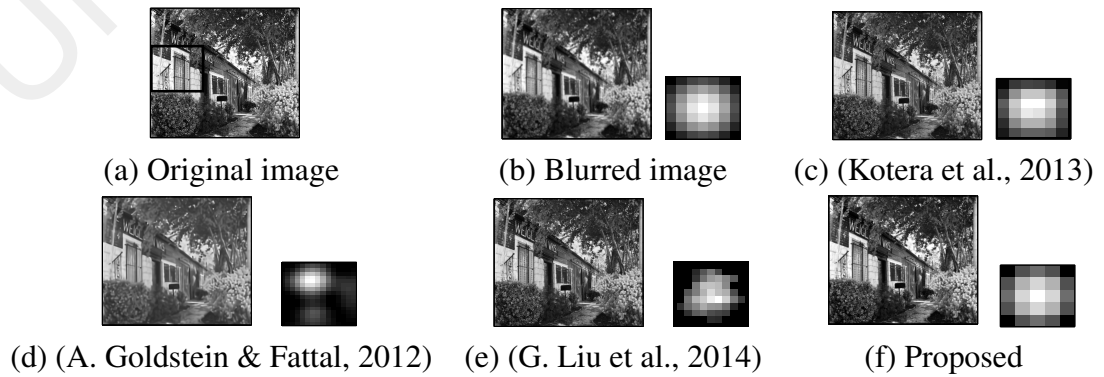


Figure 5.15: Comparison of the deblurring results on *building* image blurred with Gaussian PSF of $\sigma=1.5$ and $w = 11$

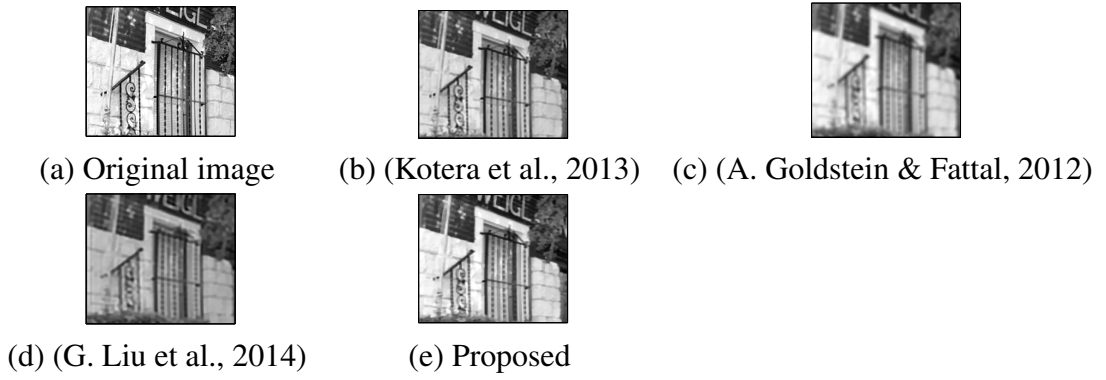


Figure 5.16: Zooming the details of *building2* image for observing the local edge restoration

database that are degraded using the PSF parameters $(\sigma, w)=(1.0, 9)$. It can be seen that the proposed method provides better SSIM scores when compared to the restoration performed in (Kotera et al., 2013), (A. Goldstein & Fattal, 2012), (G. Liu et al., 2014) for most of the test images. Similar observations can be seen for Figure 5.17(b) where the SSIM scores of the restored test images degraded with PSF parameters (2.0,13). Further, Figs. 5.17(c)-(d) shows the restored test images degraded by (3.0, 15) and (4.0, 17) respectively. In this case it can be observed that in most cases (Kotera et al., 2013) performs better when compared to the proposed method. However, the proposed method performs well than (A. Goldstein & Fattal, 2012) and (G. Liu et al., 2014) for all the images present in the LIVE database.

5.6.3 Comparative Analysis using images from the published database images

In order to measure the effectiveness of the proposed method with the existing ones, a fair comparison in terms of the restoration is performed using the images present in the existing methods. The restoration results obtained by the proposed method is based on the training of ELM using Berkeley database and testing using the images taken from (Kotera et al., 2013) and (A. Goldstein & Fattal, 2012). Table 5.10 presents the comparative analysis of the restoration performed on the blurred images. Here, the blurred images are taken from the existing method by (Kotera et al., 2013) and (A. Goldstein & Fattal,

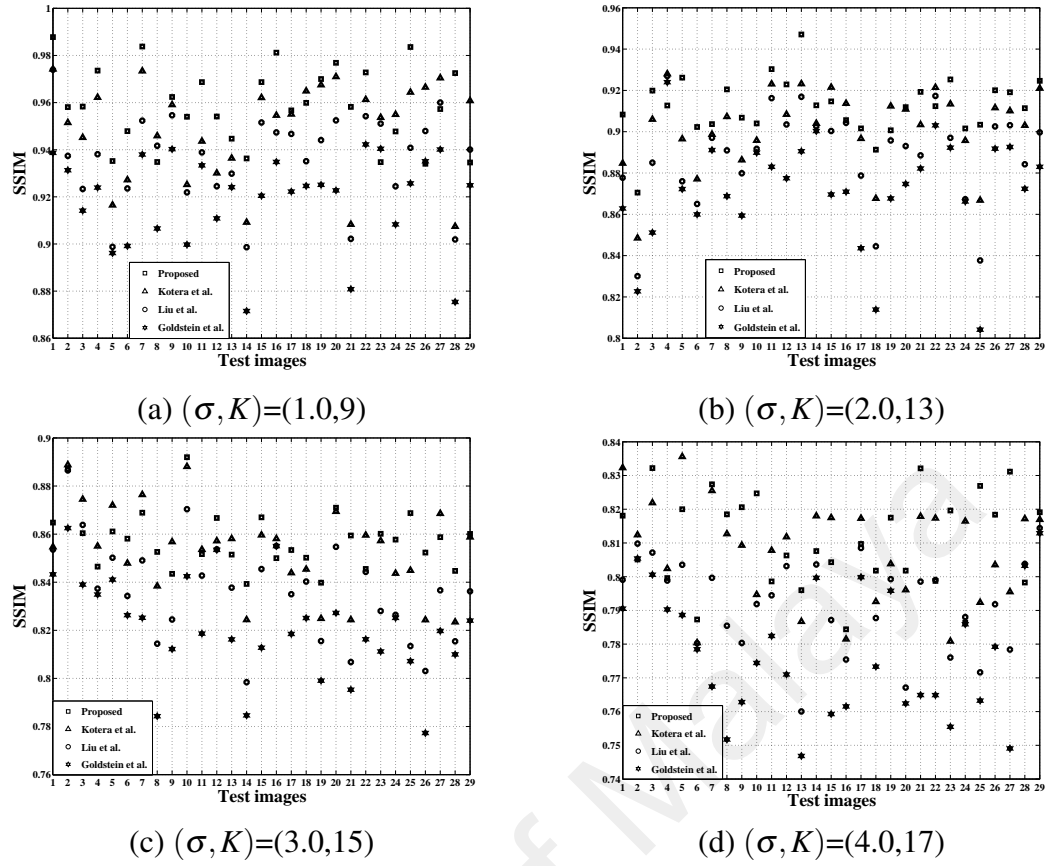


























Figure 5.17: Comparative analysis of SSIM score for the LIVE database images for different values of PSF parameters (σ, w) with existing methods namely: Kotera et al. (Kotera et al., 2013), Goldstein et al. (A. Goldstein & Fattal, 2012) and Liu et al. (G. Liu et al., 2014)

2012) for fair comparison. It can be observed that the proposed method does not provide satisfactory restoration results when compared to the work done by (Kotera et al., 2013). The reason for this is that the images used in (Kotera et al., 2013) are blurred using arbitrary PSF: a combination of Gaussian and motion PSF and the proposed method is based on restoration performed using Gaussian PSF. Due to the mismatch in the PSF estimation, the quality of the restored image using the proposed method is degraded compared to (Kotera et al., 2013). This is observed strongly in the case of two images shown in second and third row of Table 4.8 where the quality of the restoration is not as good as other methods when evaluated in terms of SSIM score. However, for the images shown in first and last row of the table, the results are competitive with the existing methods.











Moreover, two recently proposed methods by (Vera et al., 2013a) and (Yan et al.,

Table 5.10: Quantitative comparison with the existing methods by using the images present in them

Original Image	Blurred Image	Restored Image			
		(Kotera et al., 2013)	(A. Goldstein & Fattal, 2012)	(G. Liu et al., 2014)	Proposed
					
SSIM	0.6275	0.8214	0.8092	0.8378	0.8426
					
SSIM	0.6973	0.8472	0.8295	0.8356	0.8366
					
SSIM	0.6758	0.8472	0.8167	0.8223	0.8412
					
SSIM	0.6135	0.8045	0.7934	0.8247	0.8321

2012) that focus specifically for Gaussian blurred images are selected for a fair comparison with the proposed approach. Further, taking two of the images used in the paper (A. Goldstein & Fattal, 2012) for restoration helps us to understand the advantages and shortcomings of the proposed approach. Here, the restoration results obtained by the proposed method are based on the training of ELM using Caltech database and testing using the images taken from (A. Goldstein & Fattal, 2012). Table 5.11 presents the restoration results of the proposed method with the existing ones. For the restoration of face image shown in the first row, the proposed method and (Vera et al., 2013a) provides SSIM scores of 0.8346 and 0.8312 respectively. While for the other image, the proposed method provides a slightly less score when compared to (Vera et al., 2013a). These observations show that the proposed method is competitive with the existing methods when evaluated in terms of SSIM scores.

Table 5.11: Quantitative comparison with the existing methods by using the images present in them

Original Image	Blurred Image	Restored Image		
		(Vera et al., 2013a)	(Yan et al., 2012)	Proposed
				
SSIM	0.6534	0.8312	0.8225	0.8346
				
SSIM	0.5821	0.8167	0.8024	0.8112

5.7 Conclusion

This chapter focused on estimating the PSF present in the images by using edge information. The low order Tchebichef moments are good edge descriptors and hence they are used in this study. The proposed algorithm first identifies the edge blocks in the image by using low order Tchebichef moments. The difference in the Tchebichef moments between the edge blocks of the original and the reblurred image is used in the training and testing of ELM to estimate the PSF parameters sigma and size. From the results of the cross-database validation, the proposed method estimates the PSF parameters well, as shown by the correlation coefficient (CC) values. Further, once the PSF is estimated, the degraded image is restored and its quality is assessed using SSIM scores. A comparative analysis with the three existing methods is carried out using the LIVE databases images and the images from their own database. Also, the comparison is performed with the recently proposed methods that deals with Gaussian deblurring. In both the analysis, it has been found that most of the time the proposed method gave better SSIM scores.

CHAPTER 6 : CONCLUSION

Geometric moments (GM) were introduced by Hu (1962) and are widely used in image processing applications. The use of GMs in past was not considered for any image restoration task. This is due to the non-orthogonal nature of the geometric moments that leads to information redundancy. However, this problem has been solved recently which paves the way for us to explore the potential of using geometric moments in the area of image restoration. Hence, in this thesis the restoration of blurred images degraded due to Gaussian and motion blur is addressed using geometric moments. This is achieved by mathematically formulating the blur in moment domain, followed by using variational method along with Euler-Lagrange identity and alternate minimization (AM) framework to implement the restoration of the blurred images. The proposed method formulates the partial differential equation (PDE) in moment domain which uses an iterative procedure to deblur the image. Once the image is deblurred using Geometric moments, a reconstruction method using Stirling numbers is used to restore the image from moment domain to spatial domain. In this study, detailed experiments are carried out to demonstrate the effectiveness of the proposed method on the quality of the restored images by considering the effects of the regularization parameter and blur size. The quality of the restored image is evaluated using image quality assessment methods such as Structural SIMilarity (SSIM) index and Blind/Reference-less Image Spatial Quality Evaluator (BRISQUE). It has been observed that the proposed method is efficient for the task of deblurring when compared to the other existing methods.

Next, a different approach to restore the blurred images is adopted in this thesis. In this method, first the Gaussian blur is estimated followed by the use of existing non-blind algorithms for carrying out the image restoration task. In particular, the task of blur estimation is performed using Weighted geometric moments (WGM) and Tchebichef mo-

ments (TM) methods respectively. These moments are used as feature vectors which can efficiently capture the edge behavior under the influence of the blur. The proposed feature vectors along with the extreme learning machine (ELM) can be used to estimate the Gaussian blur parameters (σ, w) that creates the blurring effects. The effectiveness of the proposed methods to estimate the blur parameters is examined using cross validation. The image used in this analysis are from the standard databases, namely LIVE, CSIQ, CIDIQ, VCL, Caltech, TID2008 and Berkeley. The images in all the databases vary in terms of its content and size. These images are successively degraded using Gaussian blur with varying sigma (σ) and size (w) thus producing a total of 38,988 images to be used for training and testing of the proposed algorithm. The range of sigma (σ) varies from 0.3 to 4 in steps of 0.1 and size (w) from 7×7 to 17×17 in steps of 2×2 . Further, the cross validation evaluation is done using 10 runs. In this validation scheme, for each run, a random partition is done by selecting 70% of the data for testing while the remaining 30% as training. Once each run is done, the performance of the regression analysis is carried in terms of correlation coefficient (CC). The task of partitioning the data and its CC evaluation is repeated for all the 10 runs. Once this is done, the average accuracy which is the mean of the CC values obtained from each run is reported. Once the blur is estimated from the proposed methods, its information is utilized in the image restoration algorithms implemented in the moment and spatial domains. Specifically, for the case of TM based method, it is done using split Bregman technique, while in case of WGM method, images are deblurred using proposed moment domain technique. Once the image is deblurred using WGM method, a proposed reconstruction method known as subtractor based cascaded digital filter is employed to transform the image from moment domain to spatial domain. The method can deal with geometric moments computed from any coordinate format such as $(0, N - 1)$ and $(-1, 1)$. However, this transformation is not needed in the case of TM based method because the image deblurred using a split Bregman algorithm

is already in spatial domain. Once the images are restored using the proposed method, a fair comparison is carried out with the five existing methods. The existing methods are selected based on their ability to restore Gaussian and complex motion blurs. The results show that the proposed methods (TM and WGM) in most of the cases performs better than the five existing methods in terms of the visual quality evaluated using structural similarity (SSIM) index.

6.1 Future Research Directions

There are still many possibilities for extensions of the theory and improvement for the applications discussed in this thesis. Discrete orthogonal moments such as Krawtchouk, Zernike and Legendre etc. can be used for restoring the images via non-blind and blind algorithms proposed in Chapter 3. A comparative study in terms of the quality of the restored image can be carried out. Further, the role of these orthogonal moments can also be explored to see whether they can be used as an effective edge indicator or not. If they can be used as an effective edge indicator, then whether they can provide a better blur estimation or not. A comparative analysis can be carried out to evaluate the estimation accuracy of the blur and quality of the restored image using these discrete orthogonal moments.

REFERENCES

- Abu-Mostafa, Y. S., & Psaltis, D. (1985). Image normalization by complex moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*(1), 46–55.
- Afghahi, M., & Yuan, J. (1991). Double-edge-triggered d-flip-flops for high-speed cmos circuits. *Solid-State Circuits, IEEE Journal of*, 26(8), 1168–1170.
- Aizenberg, I., Bregin, T., Butakoff, C., Karnaukhov, V., Merzlyakov, N., & Milukova, O. (2002). Type of blur and blur parameters identification using neural network and its application to image restoration. In *Artificial neural networks Uicann 2002* (pp. 1231–1236). Springer.
- Albregtsen, F., Schulerud, H., & Yang, L. (1995). Texture classification of mouse liver cell nuclei using invariant moments of consistent regions. In *Computer analysis of images and patterns* (pp. 496–502).
- Alghoniemy, M., & Tewfik, A. H. (2004). Geometric invariance in image watermarking. *Image Processing, IEEE Transactions on*, 13(2), 145–153.
- Almeida, M., & Figueiredo, M. (2013). Parameter estimation for blind and non-blind deblurring using residual whiteness measures. *Image Processing, IEEE Transactions on*, 22(7), 2751–2763.
- Almeida, M. S., & Almeida, L. B. (2010). Blind and semi-blind deblurring of natural images. *IEEE Transactions on Image Processing*, 19(1), 36–52.
- Almeida, M. S., Figueiredo, M., et al. (2013). Blind image deblurring with unknown boundaries using the alternating direction method of multipliers. In *Image processing (icip), 2013 20th ieee international conference on* (pp. 586–590).
- Al-Rawi, M. (2008). Fast zernike moments. *Journal of Real-Time Image Processing*, 3(1), 89–96.
- Al-Rawi, M., & Jie, Y. (2002). Practical fast computation of zernike moments. *Journal of Computer Science and Technology*, 17(2)(2), 181–188.
- Altuwaijri, M. M., Bayoumi, M., et al. (1994). Arabic text recognition using neural networks. In *Circuits and systems, 1994. iscas'94., 1994 ieee international symposium on* (Vol. 6, pp. 415–418).
- Amizic, B., Molina, R., & Katsaggelos, A. K. (2012). Sparse bayesian blind image deconvolution with parameter estimation. *EURASIP Journal on Image and Video Processing*, 2012(1), 1–15.
- Bae, S., & Durand, F. (2007). Defocus magnification. In *Computer graphics forum* (Vol. 26, pp. 571–579).
- Banham, M. R., & Katsaggelos, A. K. (1997). Digital image restoration. *Signal Process-*

ing Magazine, *IEEE*, 14(2), 24–41.

- Bar, L., Sochen, N., & Kiryati, N. (2004). Variational pairing of image segmentation and blind restoration. *Computer Vision-ECCV 2004*, 166–177.
- Ben-Ezra, M., & Nayar, S. (2003). Motion deblurring using hybrid imaging. In *Computer vision and pattern recognition, 2003. proceedings. 2003 ieee computer society conference on* (Vol. 1, p. I-657-I-664 vol.1).
- Bimbo, A. D. (1999). *Visual information retrieval*. Morgan Kaufmann.
- Bioucas-Dias, J. M., Figueiredo, M. A., & Oliveira, J. P. (2006). Total variation-based image deconvolution: a majorization-minimization approach. In *Acoustics, speech and signal processing, 2006. icassp 2006 proceedings. 2006 ieee international conference on* (Vol. 2, pp. 861–864).
- Bong, D. B. L., & Khoo, B. E. (2014). Blind image blur assessment by using valid reblur range and histogram shape difference. *Signal Processing: Image Communication*, 29(6), 699–710.
- Bouma, H., Dijk, J., & van Eekeren, A. W. (2012). Precise local blur estimation based on the first-order derivative. In *Spie defense, security, and sensing* (pp. 839904–839904).
- Cai, J.-F., Dong, B., & Shen, Z. (2016). Image restoration: a wavelet frame based model for piecewise smooth functions and beyond. *Applied and Computational Harmonic Analysis*, 41(1), 94–138.
- Campisi, P., & Egiazarian, K. (2007). *Blind image deconvolution: theory and applications*. CRC press.
- Candes, E. J., & Donoho, D. L. (2002). Recovering edges in ill-posed inverse problems: Optimality of curvelet frames. *Annals of statistics*, 784–842.
- Carasso, A. S. (2003). The apex method in image sharpening and the use of low exponent lévy stable laws. *SIAM Journal on Applied Mathematics*, 63(2), 593–618.
- Ceccarelli, M. (2007). A finite markov random field approach to fast edge-preserving image recovery. *Image and Vision Computing*, 25(6), 792–804.
- Chambolle, A., De Vore, R. A., Lee, N.-Y., & Lucier, B. J. (1998). Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *Image Processing, IEEE Transactions on*, 7(3), 319–335.
- Chan, T., & Shen, J. (2005). *Image processing and analysis: Variational, pde, wavelet, and stochastic methods*. Society for Industrial and Applied Mathematics.
- Chan, T., & Wong, C.-K. (1998). Total variation blind deconvolution. *Image Processing, IEEE Transactions on*, 7(3), 370–375.
- Chang, K.-H., Paramesran, R., Asli, B. H. S., & Lim, C.-L. (2012). Efficient hardware

- accelerators for the computation of tchebichef moments. *Circuits and Systems for Video Technology, IEEE Transactions on*, 22(3), 414–425.
- Chang, M. S., Yun, S. W., & Park, P. (2007). Psf search algorithm for dual-exposure type blurred image. *Int. Journal of Applied Science, Engineering and Technology*, 4.
- Chantas, G., Galatsanos, N., Likas, A., & Saunders, M. (2008). Variational bayesian image restoration based on a product of-distributions image prior. *Image Processing, IEEE Transactions on*, 17(10), 1795–1805.
- Chantas, G. K., Galatsanos, N. P., & Likas, A. C. (2006). Bayesian restoration using a new nonstationary edge-preserving image prior. *Image Processing, IEEE Transactions on*, 15(10), 2987–2997.
- Chen, J., Dong, W., Feng, H., Xu, Z., & Li, Q. (2013). High quality non-blind image deconvolution using the fields of experts prior. *Optik-International Journal for Light and Electron Optics*, 124(18), 3601–3606.
- Chen, L., & Yap, K.-H. (2003). Identification of blur support size in blind image deconvolution. In *Information, communications and signal processing, 2003* (Vol. 1, p. 503-507).
- Chiang, M.-C., & Boulton, T. E. (1997). Local blur estimation and super-resolution. In *cvpr* (p. 821).
- Cho, T. S., Paris, S., Horn, B. K., & Freeman, W. T. (2011). Blur kernel estimation using the radon transform. In *Computer vision and pattern recognition (cvpr), 2011 ieee conference on* (pp. 241–248).
- Chung, K.-L., & Chen, P.-C. (2005). An efficient algorithm for computing moments on a block representation of gray-scale image. *Pattern Recognition*, 38(12), 2578-2586.
- CIDIQ. (n.d.). <http://www.ansatt.hig.no/mariusp/CIDIQ.zip>.
- Coifman, R., & Donoho, D. (1995). *Translation invariant denoising, wavelets and statistics, anelis antoniadis*. Springer, New York, NY, USA.
- Danielyan, A., Katkovnik, V., & Egiazarian, K. (2012). Bm3d frames and variational image deblurring. *Image Processing, IEEE Transactions on*, 21(4), 1715–1728.
- Dash, R., & Majhi, B. (2014). Motion blur parameters estimation for image restoration. *Optik-International Journal for Light and Electron Optics*, 125(5), 1634–1640.
- Daubechies, I., et al. (1992). *Ten lectures on wavelets* (Vol. 61). SIAM.
- Deshpande, A. M., & Patnaik, S. (2014). Single image motion deblurring: An accurate psf estimation and ringing reduction. *Optik-International Journal for Light and Electron Optics*, 125(14), 3612–3618.
- Dijk, J., Van Ginkel, M., Van Asselt, R. J., Van Vliet, L. J., & Verbeek, P. W. (2003). A new sharpness measure based on gaussian lines and edges. In *Computer analysis of*

images and patterns (pp. 149–156).

- D.Li, R. M. M., & Simske, S. (2007). Atmospheric turbulence degraded image restoration using principal components analysis. *IEEE Geoscience and Remote Sensing Letters*, 4(3), 340–344.
- Dobrosotskaya, J. A., & Bertozzi, A. L. (2008). A wavelet-laplace variational technique for image deconvolution and inpainting. *IEEE Transactions on Image Processing*, 17(5), 657–663.
- Dong, W., Zhang, L., Shi, G., & Li, X. (2013). Nonlocally centralized sparse representation for image restoration. *Image Processing, IEEE Transactions on*, 22(4), 1620–1630.
- Donoho, D. L. (1995a). De-noising by soft-thresholding. *Information Theory, IEEE Transactions on*, 41(3), 613–627.
- Donoho, D. L. (1995b). Nonlinear solution of linear inverse problems by wavelet-vaguelette decomposition. *Applied and computational harmonic analysis*, 2(2), 101–126.
- Ducottet, C., Fournel, T., & Barat, C. (2004). Scale-adaptive detection and local characterization of edges based on wavelet transform. *Signal Processing*, 84(11), 2115–2137.
- Duijster, A., Scheunders, P., & De Backer, S. (2009). Wavelet-based em algorithm for multispectral-image restoration. *Geoscience and Remote Sensing, IEEE Transactions on*, 47(11), 3892–3898.
- Elad, M., Figueiredo, M. A., & Ma, Y. (2010). On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE*, 98(6), 972–982.
- Elder, J. H., & Zucker, S. W. (1998). Local scale control for edge detection and blur estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(7), 699–716.
- Fadili, M., & Starck, J.-L. (2006). Sparse representation-based image deconvolution by iterative thresholding. *ADA IV*.
- Fahmy, M. F., Raheem, G. M. A., Mohamed, U. S., & Fahmy, O. F. (2012). A new fast iterative blind deconvolution algorithm. *Journal of Signal and Information Processing*, 3(1), 98–108.
- Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., & Freeman, W. T. (2006, July). Removing camera shake from a single photograph. *ACM Trans. Graph.*, 25(3), 787–794.
- Fieguth, P. (2010). *Statistical image processing and multidimensional modeling*. Springer.
- Figueiredo, M. A., & Nowak, R. D. (2003). An em algorithm for wavelet-based image

- restoration. *Image Processing, IEEE Transactions on*, 12(8), 906–916.
- Fink, M. (2003). The full images for natural knowledge caltech office db.
- Flusser, J. (1998). Fast calculation of geometric moments of binary images.
- Flusser, J. (2000). Refined moment calculation using image block representation. *IEEE Transaction on Image Processing*, 9(11).
- Flusser, J., Zitova, B., & Suk, T. (2009). *Moments and moment invariants in pattern recognition*. Wiley.
- Ghorbel, F., Derrode, S., Dhahbi, S., Mezhoud, R., et al. (2005). Reconstructing with geometric moments. In *Proc. int. conf. on machine intelligence: Acidca-icmi*.
- Goldstein, A., & Fattal, R. (2012). Blur-kernel estimation from spectral irregularities. In *Computer vision—eccv 2012* (pp. 622–635). Springer.
- Goldstein, T., & Osher, S. (2009). The split bregman method for l1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2), 323–343.
- Golub, G. H., Heath, M., & Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2), 215–223.
- Gunturk, B. K., & Li, X. (2012). *Image restoration: Fundamentals and advances*. CRC Press.
- Guo, Y., Lee, H., & Teo, C. (1997). Blind restoration of images degraded by space-variant blurs using iterative algorithms for both blur identification and image restoration. *Image and vision computing*, 15(5), 399–410.
- Hansen, P. C. (1992). Analysis of discrete ill-posed problems by means of the l-curve. *SIAM review*, 34(4), 561–580.
- Hansen, P. C. (2010). *Discrete inverse problems: insight and algorithms* (Vol. 7). SIAM.
- Hatamian, M. (1986). A real-time two-dimensional moment generating algorithm and its single chip implementation. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 34(3), 546–553.
- He, X., Fan, J., & Zheng, Z. (2015). Image restoration with l 2-type edge-continuous overlapping group sparsity. *Pattern Recognition Letters*, 68, 211–216.
- Honarvar, B., Paramesran, R., & Lim, C.-L. (2014). Image reconstruction from a complete set of geometric and complex moments. *Signal Processing*, 98(0), 224 - 232.
- Hosny, K. M. (2007). Exact and fast computation of geometric moments for gray level images. *Applied Mathematics and Computation*, 189(2), 1214–1222.
- Hosny, K. M., Papakostas, G. A., & Koulouriotis, D. (2013). Accurate reconstruction of noisy medical images using orthogonal moments. In *Digital signal processing (dsp)*,

2013 18th international conference on (pp. 1–6).

- [Computer software manual]. (n.d.). Retrieved from <http://classes.soe.ucsc.edu/ee264/Winter09/SECURE/5-LocalOperations.pdf>
- Hu, H., & De Hann, G. (2006). Low cost robust blur estimator. In *Image processing, 2006 ieee international conference on* (pp. 617–620).
- Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2), 179–187.
- Hu, Z., Huang, J.-B., & Yang, M.-H. (2010). Single image deblurring with adaptive dictionary learning. In *Image processing (icp), 2010 17th ieee international conference on* (pp. 1169–1172).
- Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, 70(1), 489–501.
- Jia, J. (2007). Single image motion deblurring using transparency. In *Computer vision and pattern recognition, 2007. cvpr '07. ieee conference on* (p. 1-8).
- Jones, B., Schaefer, G., & Zhu, S. (2004). Content-based image retrieval for medical infrared images. In *Engineering in medicine and biology society, 2004. iembs'04. 26th annual international conference of the ieee* (Vol. 1, pp. 1186–1187).
- Katsaggelos, A. K. (1989). Iterative image restoration algorithms. *Optical engineering*, 28(7), 287735–287735.
- Kayargadde, V. (1995). *Feature extraction for image quality prediction* (Unpublished doctoral dissertation). Technische Universiteit Eindhoven.
- Kennedy, L. M., & Basu, M. (2000). Application of projection pursuit learning to boundary detection and deblurring in images. *Pattern Recognition*, 33(12), 2019–2031.
- Kotera, J., Šroubek, F., & Milanfar, P. (2013). Blind deconvolution using alternating maximum a posteriori estimation with heavy-tailed priors. In *Computer analysis of images and patterns* (pp. 59–66).
- Kotoulas, L., & Andreadis, I. (2005, June). Real-time computation of zernike moments. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(6), 801–809. doi: 10.1109/TCSVT.2005.848302
- Kotoulas, L., & Andreadis, I. (2006). Fast computation of chebyshev moments. *Circuits and Systems for Video Technology, IEEE Transactions on*, 16(7), 884–888.
- Kotoulas, L., & Andreadis, I. (2008). Fast moment generating architectures. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(4), 533–537.
- Krahmer, F., Lin, Y., McAdoo, B., Ott, K., Wang, J., Widemann, D., & Wohlberg, B. (2006). Blind image deconvolution: Motion blur estimation. *IMA Preprints Series*, 2133–5.

- Krishnan, D., & Fergus, R. (2009). Fast image deconvolution using hyper-laplacian priors. In *Advances in neural information processing systems* (pp. 1033–1041).
- Krishnan, D., Tay, T., & Fergus, R. (2011). Blind deconvolution using a normalized sparsity measure. In *Ieee conference in computer vision and pattern recognition (cvpr)* (p. 233-240).
- Krylov, A. S., & Nasonov, A. V. (2009). Adaptive image deblurring with ringing control. In *2009 fifth international conference on image and graphics* (pp. 72–75).
- Kumar, A., & Paramesran, R. (2014). Geometric moment extraction equipment for image processing applications. *Review of Scientific Instruments*, 85(4), 044710.
- Kundur, D., & Hatzinakos, D. (1996). Blind image deconvolution. *Signal Processing Magazine, IEEE*, 13(3), 43–64.
- Langer, A., Osher, S., & Schönlieb, C.-B. (2013). Bregmanized domain decomposition for image restoration. *Journal of Scientific Computing*, 54(2-3), 549–576.
- Larson, E. C., & Chandler, D. M. (2010). Most apparent distortion: full-reference image quality assessment and the role of strategy. *Journal of Electronic Imaging*, 19(1), 011006–011006.
- Levin, A. (2006). Blind motion deblurring using image statistics. In *Advances in neural information processing systems* (pp. 841–848).
- Levin, A., Weiss, Y., Durand, F., & Freeman, W. T. (2009). Understanding and evaluating blind deconvolution algorithms. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on* (pp. 1964–1971).
- Li, B. (1993). A new computation of geometric moments. *Pattern Recognition*, 26(1), 109-113.
- Li, C., Yuan, W., Bovik, A., & Wu, X. (2011). No-reference blur index using blur comparisons. *Electronics letters*, 47(17), 962–963.
- Li, L., Lin, W., Wang, X., Yang, G.-M., Bahrami, K., & Kot, A. C. (2015). No-reference image blur assessment based on discrete orthogonal moments.
- Li, X. (2011). Fine-granularity and spatially-adaptive regularization for projection-based image deblurring. *Image Processing, IEEE Transactions on*, 20(4), 971–983.
- Likas, C., & Galatsanos, N. P. (2004). A variational approach for bayesian blind image deconvolution. *Signal Processing, IEEE Transactions on*, 52(8), 2222–2233.
- Liu, G., Chang, S., & Ma, Y. (2014). Blind image deblurring using spectral properties of convolution operators. *Image Processing, IEEE Transactions on*, 23(12), 5047–5056.
- Liu, J., & Zhang, T. (2005). Recognition of the blurred image by complex moment invariants. *Pattern Recognition Letters*, 26(8), 1128–1138.

- Liu, R., Li, Z., & Jia, J. (2008). Image partial blur detection and classification. In *Computer vision and pattern recognition, 2008. cvpr 2008. IEEE conference on* (pp. 1–8).
- Ma, L., Li, S., Zhang, F., & Ngan, K. N. (2011). Reduced-reference image quality assessment using reorganized dct-based image representation. *Multimedia, IEEE Transactions on*, 13(4), 824–829.
- Ma, L., & Zeng, T. (2016). Image deblurring via total variation based structured sparse model selection. *Journal of Scientific Computing*, 67(1), 1–19.
- Ma, X., Wang, H., Xue, B., Zhou, M., Ji, B., & Li, Y. (2014). Depth-based human fall detection via shape features and improved extreme learning machine. *IEEE journal of biomedical and health informatics*, 18(6), 1915–1922.
- Maidment, A. D., Bakic, P. R., & Gavenonis, S. (2012). *Breast imaging: 11th international workshop, iwdm 2012, philadelphia, pa, usa, july 8-11, 2012. proceedings*. Springer.
- Mairal, J., Bach, F., Ponce, J., Sapiro, G., & Zisserman, A. (2009). Non-local sparse models for image restoration. In *Computer vision, 2009 IEEE 12th international conference on* (pp. 2272–2279).
- Mairal, J., Elad, M., & Sapiro, G. (2008). Sparse representation for color image restoration. *Image Processing, IEEE Transactions on*, 17(1), 53–69.
- Mallat, S. (1999). *A wavelet tour of signal processing*. Academic press.
- Markandey, V., & DeFigueiredo, R. J. (1992). Robot sensing techniques based on high-dimensional moment invariants and tensors. *IEEE transactions on robotics and automation*, 8(2), 186–195.
- Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Computer vision, 2001. ICCV 2001. proceedings. eighth IEEE international conference on* (Vol. 2, pp. 416–423).
- Martinez, J., & Thomas, F. (2002). Efficient computation of local geometric moments. *Image Processing, IEEE Transactions on*, 11(9), 1102–1111.
- Masschaele, B., Dierick, M., Van Hoorebeke, L., Jacobs, P., Vlassenbroeck, J., & Cnudde, V. (2005). Neutron ct enhancement by iterative de-blurring of neutron transmission images. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 542(1), 361–366.
- Meyer-Baese, A., & Schmid, V. J. (2014). *Pattern recognition and signal analysis in medical imaging*. Elsevier.
- Mittal, A., Moorthy, A., & Bovik, A. (2012). No-reference image quality assessment in the spatial domain. *Image Processing, IEEE Transactions on*, 21(12), 4695–4708.

- Mittal, A., Soundararajan, R., & Bovik, A. C. (2013). Making a completely blind image quality analyzer. *IEEE Signal Processing Letters*, 20(3), 209–212.
- Moghaddam, M. E. (2008). Out of focus blur estimation using genetic algorithm. In *Systems, signals and image processing, 2008. iwSSIP 2008. 15th international conference on* (pp. 417–420).
- Money, J. H., & Kang, S. H. (2008). Total variation minimizing blind deconvolution with shock filter reference. *Image and Vision Computing*, 26(2), 302–314.
- Mukundan, R., Ong, S. H., & Lee, P. A. (2001a). Discrete vs. continuous orthogonal moments for image analysis. In *International conference on imaging science, systems, and technology* (p. 23-29).
- Mukundan, R., Ong, S. H., & Lee, P. A. (2001b, Sep). Image analysis by Tchebichef moments. *IEEE Transactions on Image Processing*, 10(9), 1357-1364.
- Nakamura, R., Mitsukura, Y., & Hamada, N. (2013). Iterative pca approach for blind restoration of single blurred image. In *International symposium on intelligent signal processing and communications systems* (p. 543-546).
- Neelamani, R., Choi, H., & Baraniuk, R. (2004). Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems. *Signal Processing, IEEE Transactions on*, 52(2), 418–433.
- Oliveira, J. P., Figueiredo, M. A., & Bioucas-Dias, J. M. (2007). Blind estimation of motion blur parameters for image deconvolution. In *Pattern recognition and image analysis* (pp. 604–611). Springer.
- Osher, S., Burger, M., Goldfarb, D., Xu, J., & Yin, W. (2005). An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2), 460–489.
- Pang, Z.-F., & Yang, Y.-F. (2011). A projected gradient algorithm based on the augmented lagrangian strategy for image restoration and texture extraction. *Image and Vision Computing*, 29(2), 117–126.
- Papakostas, G., Boutalis, Y., Karras, D., & Mertzios, B. (2009). Pattern classification by using improved wavelet compressed zernike moments. *Applied Mathematics and Computation*, 212(1), 162 - 176. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0096300309001313> doi: <http://dx.doi.org/10.1016/j.amc.2009.02.029>
- Papakostas, G., Tsougenis, E., & Koulouriotis, D. (2010). Near optimum local image watermarking using krawtchouk moments. In *Imaging systems and techniques (IST), 2010 IEEE international conference on* (pp. 464–467).
- Papakostas, G. A., Boutalis, Y. S., Karras, D. A., & Mertzios, B. G. (2007). A new class of zernike moments for computer vision applications. *Information Sciences*, 177(13), 2802–2819.

- Papakostas, G. A., Karakasis, E. G., & Koulouriotis, D. E. (2008). Efficient and accurate computation of geometric moments on gray-scale images. *Pattern Recognition*, 41, 1895-1904.
- Ponomarenko, N., Lukin, V., Zelensky, A., Egiazarian, K., Carli, M., & Battisti, F. (2009). Tid2008-a database for evaluation of full-reference visual quality assessment metrics. *Advances of Modern Radioelectronics*, 10(4), 30-45.
- Portilla, J., Strela, V., Wainwright, M. J., & Simoncelli, E. P. (2003). Image denoising using scale mixtures of gaussians in the wavelet domain. *Image Processing, IEEE Transactions on*, 12(11), 1338-1351.
- Raskar, R., Agrawal, A., & Tumblin, J. (2006, July). Coded exposure photography: motion deblurring using fluttered shutter. *ACM Trans. Graph.*, 25(3), 795-804.
- Reeves, A. P. (1982). Parallel algorithms for real-time image processing. *Multicomputers and Image Processing: Algorithms and Programs*, 7-18.
- Rekleitis, I. N. (1996). Optical flow recognition from the power spectrum of a single blurred image. In *Image processing, 1996. proceedings., international conference on* (Vol. 3, pp. 791-794).
- Richardson, W. H. (1972). Bayesian-based iterative method of image restoration. *JOSA*, 62(1), 55-59.
- Rooms, F., Philips, W. R., & Portilla, J. (2004). Parametric psf estimation via sparseness maximization in the wavelet domain. In *Optics east* (pp. 26-33).
- Rooms, F., Pizurica, A., & Philips, W. (2002). Estimating image blur in the wavelet domain. In *Ieee international conference on acoustics speech and signal processing* (Vol. 4, pp. 4190-4190).
- Rooms, F., Ronsse, M., Pizurica, A., & Philips, W. (2002). Psf estimation with applications in autofocus and image restoration. In *Ieee signal processing symposium, leuven, belgium*.
- Ruiz, P., Zhou, X., Mateos, J., Molina, R., & Katsaggelos, A. K. (2015). Variational bayesian blind image deconvolution: A review. *Digital Signal Processing*(0), 1-7.
- Schelten, K., Nowozin, S., Jancsary, J., Rother, C., & Roth, S. (2015). Interleaved regression tree field cascades for blind image deconvolution. In *Ieee winter conference on applications of computer vision*.
- Schmidt, U., Rother, C., Nowozin, S., Jancsary, J., & Roth, S. (2013). Discriminative non-blind deblurring. In *Computer vision and pattern recognition (cvpr), 2013 ieee conference on* (pp. 604-611).
- Schuler, C. J., Burger, H. C., Harmeling, S., & Scholkopf, B. (2013). A machine learning approach for non-blind image deconvolution. In *Computer vision and pattern recognition (cvpr), 2013 ieee conference on* (pp. 1067-1074).

- Shan, Q., Jia, J., & Agarwala, A. (2008). High-quality motion deblurring from a single image. In *Acm transactions on graphics (tog)* (Vol. 27, p. 73).
- Sheikh, H. R., & Bovik, A. C. (2006). Image information and visual quality. *Image Processing, IEEE Transactions on*, 15(2), 430–444.
- Sheikh, H. R., Bovik, A. C., & Cormack, L. (2005). No-reference quality assessment using natural scene statistics: Jpeg2000. *Image Processing, IEEE Transactions on*, 14(11), 1918–1927.
- Sheikh, H. R., Wang, Z., Cormack, L., & Bovik, A. C. (2005). *Live image quality assessment database release 2*.
- Shi, Y., Hong, H., Song, J., & Hua, X. (2015). Blind image deblurring with edge enhancing total variation regularization. In *Selected proceedings of the photoelectronic technology committee conferences held august-october 2014* (pp. 95222B–95222B).
- Shi, Y., & Yang, X. (2010). New total variation regularized l1 model for image restoration. *Digital Signal Processing*, 20(6), 1656–1676.
- Sim, D.-G., Kim, H.-K., & Park, R.-H. (2004). Invariant texture retrieval using modified zernike moments. *Image and Vision Computing*, 22(4), 331–342.
- Soleimani, S., Rooms, F., & Philips, W. (2013). Efficient blur estimation using multi-scale quadrature filters. *Signal Processing*, 93(7), 1988–2002.
- Spiliotis, I. M., & Mertzios, B. G. (1998). Real-time computation of two-dimensional moments on binary image using image block representation. *IEEE Transaction on Image Processing*, 7, 1609–1615.
- Sroubek, F., & Flusser, J. (2003). Multichannel blind iterative image restoration. *Image Processing, IEEE Transactions on*, 12(9), 1094–1106.
- Šroubek, F., & Milanfar, P. (2012). Robust multichannel blind deconvolution via fast alternating minimization. *Image Processing, IEEE Transactions on*, 21(4), 1687–1700.
- Stern, A., Kruchakov, I., Yoavi, E., & Kopeika, N. S. (2002). Recognition of motion-blurred images by use of the method of moments. *Applied Optics*, 41(11), 2164–2171.
- Stoker, D. S., Wedd, J., Lavelle, E., & van der Laan, J. (2013, Mar). Restoration and recognition of distant, blurry irises. *Appl. Opt.*, 52(9), 1864–1875.
- Su, B., Lu, S., & Tan, C. L. (2011). Blurred image region detection and classification. In *Proceedings of the 19th acm international conference on multimedia* (pp. 1397–1400).
- Su, M., & Basu, M. (2002). A hybrid learning system for image deblurring. *Pattern recognition*, 35(12), 2881–2894.

- Suresh, S., Babu, R. V., & Kim, H. J. (2009). No-reference image quality assessment using modified extreme learning machine classifier. *Applied Soft Computing*, 9(2), 541–552.
- Tam, N. W., Lee, J.-S., Hu, C.-M., Liu, R.-S., & Chen, J.-C. (2011). A haar-wavelet-based Lucy–Richardson algorithm for positron emission tomography image restoration. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 648, S122–S127.
- Tang, S., Gong, W., Li, W., & Wang, W. (2014). Non-blind image deblurring method by local and nonlocal total variation models. *Signal Processing*, 94(0), 339 - 349.
- Tao, S., Dong, W., Feng, H., Xu, Z., & Li, Q. (2013). Non-blind image deconvolution using natural image gradient prior. *Optik - International Journal for Light and Electron Optics*, 124(24), 6599 - 6605.
- Teague, M. R. (1980). Image analysis via the general theory of moments*. *JOSA*, 70(8), 920–930.
- Teh, C.-H., & Chin, R. T. (1988). On image analysis by the methods of moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(4), 496–513.
- Telatar, Z. (2005). Adaptive filter design for image deblurring by using multi-criteria blurred image information. *Digital Signal Processing*, 15(1), 4–18.
- Thung, K.-H., Paramesran, R., & Lim, C.-L. (2012). Content-based image quality metric using similarity measure of moment vectors. *Pattern Recognition*, 45(6), 2193–2204.
- Tong, H., Li, M., Zhang, H., & Zhang, C. (2004). Blur detection for digital images using wavelet transform. In *Multimedia and expo, 2004. icme'04. 2004 IEEE international conference on* (Vol. 1, pp. 17–20).
- Vainikko, G. M. (1982). The discrepancy principle for a class of regularization methods. *USSR computational mathematics and mathematical physics*, 22(3), 1–19.
- Vairy, M., & Venkatesh, Y. V. (1995). Deblurring gaussian blur using a wavelet array transform. *Pattern Recognition*, 28(7), 965–976.
- Van Eekeren, A., Schutte, K., Dijk, J., de Lange, D.-J. J., & van Vliet, L. J. (2006). Super-resolution on moving objects and background. In *Image processing, 2006 IEEE international conference on* (pp. 2709–2712).
- van Eekeren, A. W., Schutte, K., Dijk, J., Schwering, P. B., van Iersel, M., & Doelman, N. J. (2012). Turbulence compensation: an overview. In *Spie defense, security, and sensing* (pp. 83550Q–83550Q).
- Vega, M., Mateos, J., Molina, R., & Katsaggelos, A. K. (2012). Astronomical image restoration using variational methods and model combination. *Statistical Methodology*, 9(1), 19–31.

- Vera, E., Vega, M., Molina, R., & Katsaggelos, A. K. (2013a). Iterative image restoration using nonstationary priors. *Applied optics*, 52(10), D102–D110.
- Vera, E., Vega, M., Molina, R., & Katsaggelos, A. K. (2013b, Apr). Iterative image restoration using nonstationary priors. *Appl. Opt.*, 52(10), D102–D110.
- Wang, G., & Wang, S. (2006). Recursive computation of tchebichef moment and its inverse transform. *Pattern Recognition*, 39(1), 47–56.
- Wang, Z., Bovik, A., Sheikh, H., & Simoncelli, E. (2004b). Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4), 600–612.
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004a). Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4), 600–612.
- Wee, C.-Y., Paramesran, R., & Mukundan, R. (2008). Fast computation of geometric moments using a symmetric kernel. *Pattern Recognition*, 41, 2369–2380.
- Wiener, N. (1949). *Extrapolation, interpolation, and smoothing of stationary time series* (Vol. 2). MIT press Cambridge, MA.
- Wong, W.-H., & Siu, W.-C. (1999). Improved digital filter structure for fast moments computation. *IEE Proceedings-Vision, Image and Signal Processing*, 146(2), 73–79.
- Xiang, S., Meng, G., Wang, Y., Pan, C., & Zhang, C. (2012). Image deblurring with matrix regression and gradient evolution. *Pattern Recognition*, 45(6), 2164–2179.
- Xie, F., Lu, Y., Bovik, A. C., Jiang, Z., & Meng, R. (2016). Application-driven no-reference quality assessment for dermoscopy images with multiple distortions. *IEEE Transactions on Biomedical Engineering*, 63(6), 1248–1256.
- Xue, W., Zhang, L., Mou, X., & Bovik, A. (2014). Gradient magnitude similarity deviation: A highly efficient perceptual image quality index.
- Yan, L., Fang, H., & Zhong, S. (2012). Blind image deconvolution with spatially adaptive total variation regularization. *Optics letters*, 37(14), 2778–2780.
- Yang, L., & F., A. (1996). Fast and exact computation of cartesian geometric moments using discrete green's theorem. *Pattern Recognition*, 29(7), 1061–1073.
- Yap, P., Raveendran, P., & Ong, S. (2001). Chebyshev moments as a new set of moments for image reconstruction. In *Neural networks, 2001. proceedings. ijcn'01. international joint conference on* (Vol. 4, pp. 2856–2860).
- Yap, P.-T., & Raveendran, P. (2002). Image restoration of noisy images using tchebichef moments. In *Circuits and systems, 2002. apccas'02. 2002 asia-pacific conference on* (Vol. 2, pp. 525–528).
- Yin, H., & Hussain, I. (2006). Blind source separation and genetic algorithm for image

- restoration. In *Advances in space technologies, 2006 international conference on* (pp. 167–172).
- Yuan, L., Sun, J., Quan, L., & Shum, H.-Y. (2007, July). Image deblurring with blurred/noisy image pairs. *ACM Trans. Graph.*, 26(3).
- Zakaria, M., Vroomen, L., Zsombor-Murray, P., & Van Kessel, J. (1987). Fast algorithm for the computation of moment invariants. *Pattern Recognition*, 20(6), 639–643.
- Zarić, A., Tatalović, N., Brajković, N., Hlevnjak, H., Lončarić, M., Dumić, E., & Grgić, S. (2012). Vcl@ fer image quality assessment database. *AUTOMATIKA: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, 53(4), 344–354.
- Zhang, J., Zhang, Q., & He, G. (2008). Blind deconvolution: multiplicative iterative algorithm. *Optics letters*, 33(1), 25–27.
- Zhang, J., Zhao, D., & Gao, W. (2014). Group-based sparse representation for image restoration. *Image Processing, IEEE Transactions on*, 23(8), 3336–3351.
- Zhang, L., Zhang, D., & Mou, X. (2011). Fsim: a feature similarity index for image quality assessment. *Image Processing, IEEE Transactions on*, 20(8), 2378–2386.
- Zhang, Y., Duijster, A., & Scheunders, P. (2012). A bayesian restoration approach for hyperspectral images. *Geoscience and Remote Sensing, IEEE Transactions on*, 50(9), 3453–3462.

LIST OF PUBLICATIONS AND PAPERS PRESENTED

1. A. Kumar, R. Paramesran, B.H Shakibaei, "Moment Domain Representation of Non-Blind Image Deblurring", *Applied Optics* 53(10), 167-171, 2014.
2. A Kumar and R. Paramesran, "Geometric moment extraction equipment for image processing applications", *Review of Scientific Instruments*, 85(4), 2014.
3. R. Paramesran and A Kumar, "Application of Image Analysis by Geometric Moments", *IEEE Transactions on Image Electronics and Visual Computing* 2(1), 1-10, 2014.
4. A. Kumar, R. Paramesran, C.L. Lim and S.C. Dass, "Estimation of the blur parameters using Tchebichef moments for image restoration". (Under review)
5. A. Kumar, M.F Hassan and R. Paramesran, "Restoration of the blurred images in moment domain via reconstruction architecture". (Under review)

APPENDIX A : DATABASE



Figure A.1: 29 Reference Images of LIVE Database

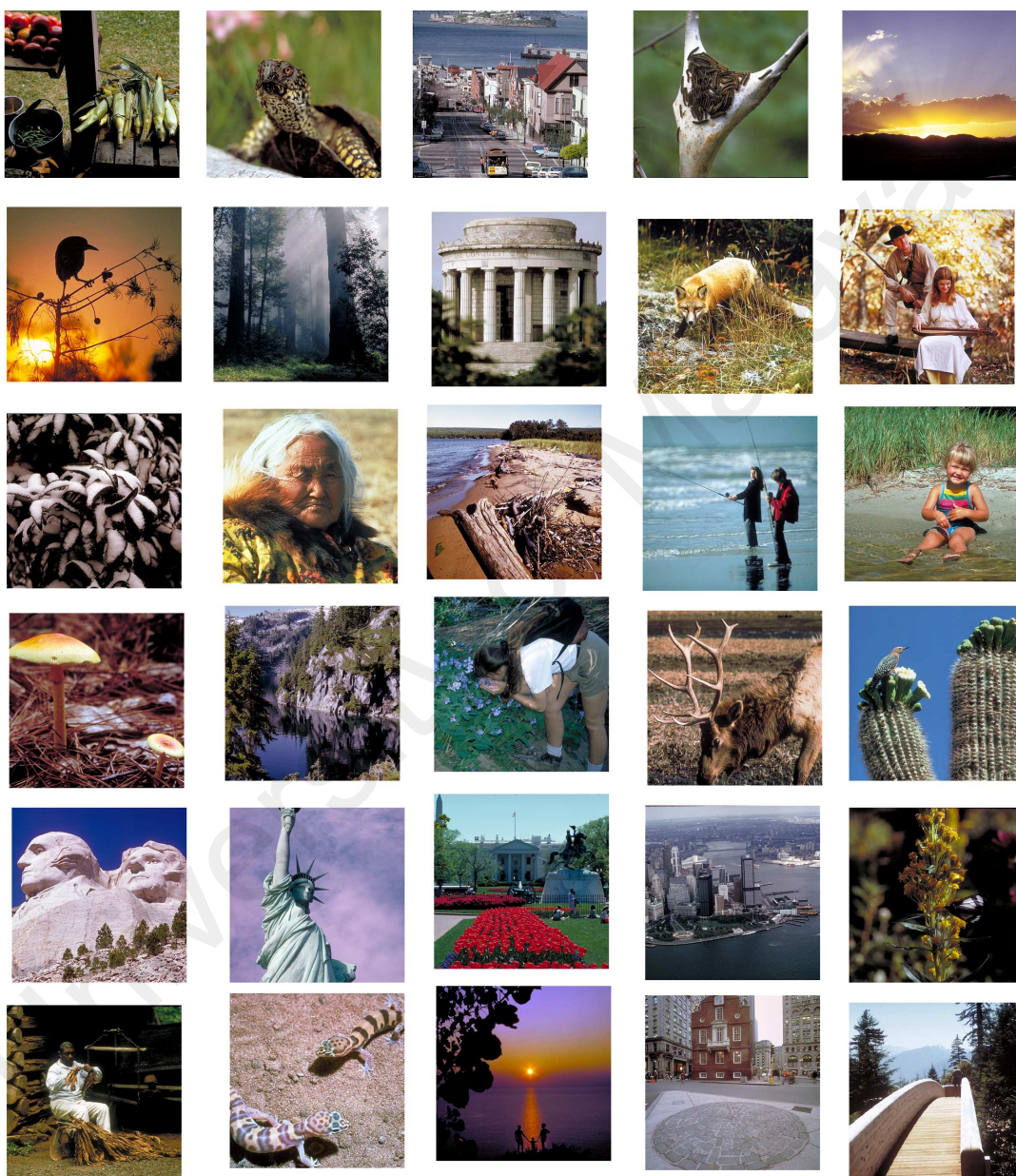


Figure A.2: 30 Reference Images of CSIQ Database

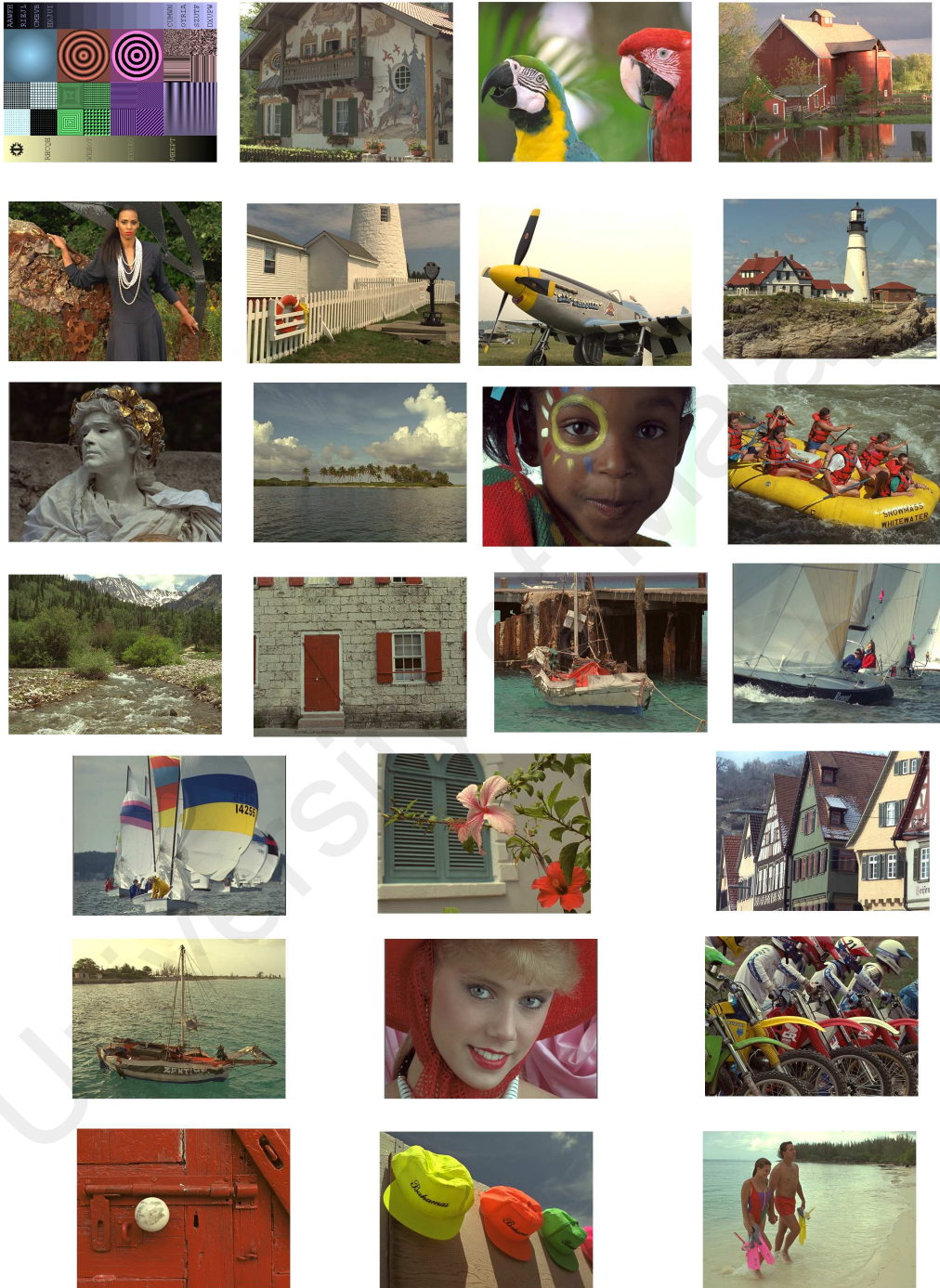


Figure A.3: 25 Reference Images of TID2008 Database



Figure A.4: 12 Reference Images of Caltech Database

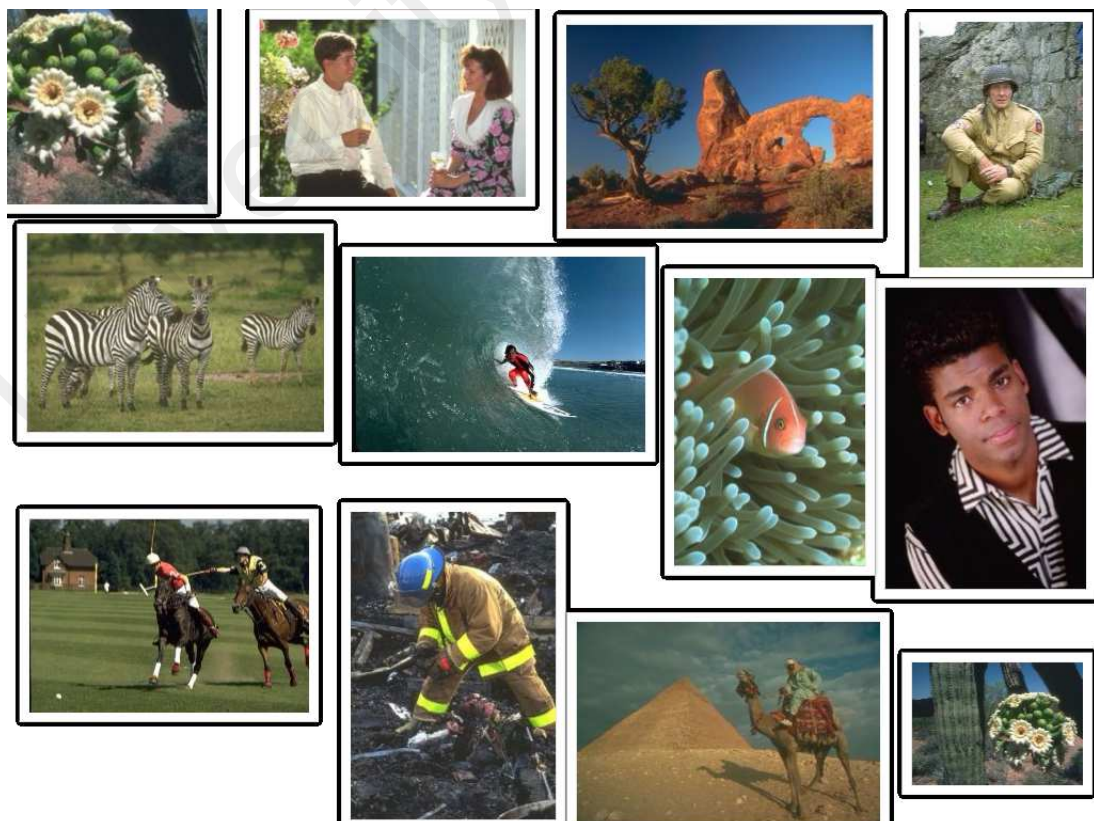


Figure A.5: 12 Reference Images of Berkeley Database



Figure A.6: Sample of blurred images obtained from Caltech and Berkeley Databases



Figure A.7: Sample of blurred images obtained from LIVE, CSIQ and TID2008 Databases

APPENDIX B : SOURCE CODES

```

%%%%%%%% Restoration using Geometric Moments %%%%%%%%%%

warning off all;

clear all;

clc;

order=8;

f=im;

[m,n,lev]=size(f);

if(lev>1)

    f1 = rgb2gray(f);

else

    f1=f;

end

image_final=double(f1);

image_final=image_final./(max(image_final(:)));

colormap(gray);imagesc(image_final)

h=fspecial('gaussian',[5 5],4);

tic

[mf mh mg]=moments(image_final,h,order);

h_est=fspecial('gaussian',[5 5],4);

est_mh=double(zeros(order+1,order+1));

for p=0:order

    for q=0:order

        for y=1:size(h_est,1)

            for x=1:size(h_est,2)

                est_mh(p+1,q+1) = est_mh(p+1,q+1)+(x-1)^p*(y-1)^q*h_est(x,y);

            end

        end

    end

end

for j3=0:order

    for k3=0:order

        fprintf('order j=%d , k=%d \n',j3,k3);

        if ((j3==0)&&(k3==0))

            A(j3+1)=iterate_moment_0(est_mh,mg); %Errors incurred in estimating

```

```

else
    [B,plot1]=fast_general_moment(est_mh,mg,j3,k3,A);    %Errors incurred in estimating
    A(j3+1,k3+1)=B;
%    fprintf('A  %d \n',A);
end
end
end

reconst = image_reconst(A,order+1,order+1);
subplot(121),colormap(gray);imagesc(image_final);
subplot(122),colormap(gray);imagesc(reconst);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Moment Calculation%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [mf,mh,mg]=moments(f,h,order)
mf = double(zeros(order+1,order+1));
mh = double(zeros(order+1,order+1));
mg = double(zeros(order+1,order+1));
for p=0:order
    for q=0:order
        for y=1:size(f,1)
            for x=1:size(f,2)
                mf(p+1,q+1) = mf(p+1,q+1)+(x-1)^p*(y-1)^q*f(x,y);
            end
        end
        for y=1:size(h,1)
            for x=1:size(h,2)
                mh(p+1,q+1) = mh(p+1,q+1)+(x-1)^p*(y-1)^q*h(x,y);
            end
        end
        for r=0:p
            for s=0:q
                mg(p+1,q+1) = mg(p+1,q+1)+nchoosek(p,r)*nchoosek(q,s)*mh(p-r+1,q-s+1)*mf(r+1,s+1);
            end
        end
    end
end

function [image]=iterate_moment_0(mh,mg)
dt=0.0001;
store=mg(1,1);

```

```

for iter = 1:200

    if (iter ==1)

        C(1,1)=mh(1,1)*(mh(1,1)*mg(1,1)-store);

        new_mf(1) = store-2*dt*C;

    else

        C(1,1)=mh(1,1)*(mh(1,1)*new_mf(iter-1)-store);

        new_mf(iter) = new_mf(iter-1)-2*dt*C;

    end

end

image=new_mf(end);

end

function [image,plo]=fast_general_moment(mh,mg,ord1,ord2,A)

dt=0.01;

p=ord1;

q=ord2;

% new_mf=zeros(1,order+1);

store=mg(p+1,q+1);

B=0;

    for i=0:p

        %           fprintf('p=%d',p);

        for j=0:q

            %           fprintf(' q=%d\n',q);

            if((i==p)&& (j==q))

                A1=nchoosek(p,i)*nchoosek(q,j)*mh(p-i+1,q-j+1)*store;

            %           fprintf('p=%d,q=%d,i= %d, j=%d , A1= %d \n',p,q,i,j,A1);

            else

                B=B+nchoosek(p,i)*nchoosek(q,j)*mh(p-i+1,q-j+1)*A(i+1,j+1);

                %fprintf('p=%d,q=%d,i= %d , j=%d , B= %d \n',p,q,i,j,B);

            end

        end

    end

    D=(A1+B)-store;

    %           fprintf('D %d\n',D);

    C=mh(1,1)*(D);

    % fprintf('c %d\n',C);

    new_mf(1) = store-2*dt*C;

    B=0; beta=1;

```

```

        for i=0:p
            for j=0:q
                if((i~=p)|| (j~=q))
                    B=B+nchoosek(p,i)*nchoosek(q,j)*mh(p-i+1,q-j+1)*A(i+1,j+1);
                end
            for iter = 2:5000
                if((i==p)&& (j==q))
                    A1=nchoosek(p,i)*nchoosek(q,j)*mh(p-i+1,q-j+1)*new__mf(iter-1);
                end
                D=(A1+B)-store;
                C=mh(1,1)*(D);
%                fprintf('D %d\n',D);
                new__mf(iter) = new__mf(iter-1)-2*dt*C;
            end
        end %                plo(1, :,beta)=new__mf;
%                beta=beta+1;
    end

    image=new__mf(end);
    plo(:,1)=new__mf;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% C Code to evaluate higher order images %%%%%%%%%%

#include<stdio.h>
#include<stdlib.h>
#include<stddef.h>
#include<string.h>
#include<conio.h>
#include<iostream>
#include<math.h>
#include<fstream>
#include<time.h>

#define SP ""
#define LF "\n"
#define pi 3.141592653589793
#define N 1 //number of images
#define NP 16384 //Change this to N*M of image
#define IP 47
// #define YSIZE 64

```

```

// #define XSIZE 64

#include "error_functions.h"

#include "additional_function.h"

#include "Geometric_Moments.h"

#include "Geometric_Moments_1.h"

#include "image_operations.h"

#include "PDE_moment_0.h"

#include "PDE_moments.h"

void main()

{

int i,j,order,count=0,count_img;

long int FILESIZE,FILESIZE_PSF;

float ***B,**a,**D;

long double **Mf,**Mh,**Mg,**IMAGE,*temp,**restore_image,*New_mf,**restore_image1;

//FILESIZE=YSIZE*XSIZE;

//FILESIZE_PSF=49;

//cout<<"FILESIZE="<<FILESIZE<<LF;

temp=Doubvector(0,NP);

IMAGE=Doubmatrix(0,500,0,500); //Size of the image is 0 to 64

B=D3fmatrix(0,40,40,4096);

Mf=Doubmatrix(0,1000,0,1000); //why increasing this solves large order problem

D=fmatrix(0,500,0,500);

Mh=Doubmatrix(0,500,0,500);

Mg=Doubmatrix(0,1000,0,1000); //why increasing this solves large order problem

New_mf=Doubvector(0,2000);

restore_image=Doubmatrix(0,500,0,500);

// Calculate moments of Image

cout<<"Enter order";

cin>>order;

char name[50];

ifstream infile;

infile.open("lena_128.txt");

//infile.open("Lena_128.txt");

//infile.open("original_image.txt");

if(!infile)

{

cout<<"Cannot open file 1"<<LF;

```

```

exit(0);
}

    count_img=1;
    for(i=0;i<NP;i++)

{
infile>>temp[i];

}

vector2matrix(temp,IMAGE,128,128); // Change this to N,M of image
moments(IMAGE,order,Mf);          // change inside the moments function limits of image
    cout<<"Mf done";
free_Doubvector(temp,0);
// free_ivector(New_mf,0);
// free_fmatrix(Mf,0,500,0);
free_Doubmatrix(IMAGE,0,500,0);
// free_fmatrix(Z,0,64,0);
free_D3fmatrix(B,0,40,40);
infile.close();
// Calculation of Moments of PSF
temp=Doubvector(0,NP);
IMAGE=Doubmatrix(0,500,0,500); //Size of the image is 0 to 64
B=D3fmatrix(0,40,40,4096);
ifstream infile2;
infile2.open("psf.txt");
if(!infile)
{
    cout<<"Cannot open file 1"<<LF;
    exit(0);
}

    count_img=1;
    for(i=0;i<NP;i++)

{
infile2>>temp[i];
}

vector2matrix(temp,IMAGE,5,5);
moments_h(IMAGE, order, Mh);
cout<<"Mh done";

```

```

b_moments(Mf,Mh,Mg,order); // Blurred Image Moments

cout<<"Mg done";

free_Doubvector(temp,0);

// free_ivector(New_mf,0);
// free_fmatrix(Mh,0,500,0);
free_Doubmatrix(IMAGE,0,500,0);
// free_fmatrix(Z,0,64,0);
free_D3fmatrix(B,0,40,40);
infile2.close();
cout<<"ÑND"<<LF;
// Program for calculation
long double vijeta=0.0;
for(int j3=0;j3<order;j3++)
{
    for(int k3=0;k3<order;k3++)
    {
        if((j3==0)&&(k3==0))
        {
            iterate_moment1(Mh, Mg, restore_image);
            //cout<<"Hello";
            //cout<<"Image is ";
            //cout<<restore_image[0][0];
        }
        else
        {
            restore_image[j3][k3]=iterate_moment(Mh, Mg, j3, k3, restore_image,New_mf,vijeta);
            //cout<<"GOLA";
        }
    }
}

//Write_Into_disc(New_mf);
for(i=0;i<order;i++)
{
    for(j=0;j<order;j++)
    {
        //restore_image2[i][j]=Mf[i][j]-restore_image[i][j];
        //restore_image3[i][j]=restore_image2[i][j]/Mf[i][j];
    }
}

```



```

cout<<" "<<restore_image[i][j];

}

cout<<"\n";

}

ofstream myfile("orginal_moments.txt");

if(myfile.is_open())
{
for( i=0;i<order;i++)
{
for(j=0;j<order;j++)
{
myfile<<Mf[i][j]<<"\n";
}
cout<<"\t";
}
myfile.close();
}
else
cout<<"Unable to open file lah";
ofstream myfile1("restored_moments.txt");
if(myfile1.is_open())
{
for( i=0;i<order;i++)
{
for(j=0;j<order;j++)
{
myfile1<<restore_image[i][j]<<"\n";
}
cout<<"\t";
}
myfile1.close();
}
else
cout<<"Unable to open file lah";
}

/*
free__Doubmatrix(Mf,0,1000,0);

```

```

free__Doubmatrix(Mg,0,1000,0);

free__Doubmatrix(Mh,0,500,0);

free__Doubmatrix(restore_image,0,500,0);

/*
for(i=0;i<200;i++)
{
cout<<New__mf[i]<<"\n";
}
*/

#include<stdio.h>
#include<stdlib.h>
#include<stddef.h>
#include<string.h>
#include<conio.h>
#include<iostream.h>
#include<math.h>
#include<fstream.h>
#include<time.h>

#define SP ""
#define LF "\n"
#define pi 3.141592653589793
#define N 1 //number of images
#define NP 16384 //Change this to N*M of image
#define IP 47
// #define YSIZE 64
// #define XSIZE 64

#include"error_functions.h"
#include "additional_function.h"
#include"Geometric_Moments.h"
#include"Geometric_Moments_1.h"
#include"image_operations.h"
#include"PDE_moment_0.h"
#include"PDE_moments.h"

void main()
{
int i,j,order,count=0,count_img;

long int FILESIZE,FILESIZE_PSF;

```

```

float ***B,**a,**D;

long double **Mf,**Mh,**Mg,**IMAGE,*temp,**restore_image,*New_mf,**restore_image1;

//FILESIZE=YSIZE*XSIZ;

//FILESIZE__PSF=49;

//cout<<"FILESIZE="<<FILESIZE<<LF;

temp=Doubvector(0,NP);

IMAGE=Doubmatrix(0,500,0,500); //Size of the image is 0 to 64

B=D3fmatrix(0,40,40,4096);

Mf=Doubmatrix(0,1000,0,1000);          //why incresing this solves large order problem

D=fmatrix(0,500,0,500);

Mh=Doubmatrix(0,500,0,500);

Mg=Doubmatrix(0,1000,0,1000); //why incresing this solves large order problem

New_mf=Doubvector(0,2000);

restore_image=Doubmatrix(0,500,0,500);

// Calculate moments of Image

cout<<"Enter order";

cin>>order;

char name[50];

ifstream infile;

infile.open("original_image.txt");

//infile.open("original_image.txt");

if(!infile)

{

cout<<"Cannot open file 1"<<LF;

exit(0);

}

count_img=1;

for(i=0;i<NP;i++)

{

infile>>temp[i];

}

vector2matrix(temp,IMAGE,128,128); // Change this to N,M of image

moments(IMAGE,order,Mf);          // change inside the moments function limits of image

cout<<"Mf done";

free__Doubvector(temp,0);

// free_ivector(New_mf,0);

```

```

// free_fmatrix(Mf,0,500,0);

free_Doubmatrix(IMAGE,0,500,0);

// free_fmatrix(Z,0,64,0);

free_D3fmatrix(B,0,40,40);

infile.close();

// Calculation of Moments of PSF

temp=Doubvector(0,NP);

IMAGE=Doubmatrix(0,500,0,500); //Size of the image is 0 to 64

B=D3fmatrix(0,40,40,4096);

ifstream infile2;

infile2.open("psf.txt");

if(!infile)

{

cout<<"Cannot open file 1"<<LF;

exit(0);

}

        count_img=1;

        for(i=0;i<NP;i++)

{

infile2>>temp[i];

}

vector2matrix(temp,IMAGE,5,5);

moments_h(IMAGE, order, Mh);

cout<<"Mh done";

b_moments(Mf,Mh,Mg,order); // Blurred Image Moments

        cout<<"Mg done";

        free_Doubvector(temp,0);

// free_ivector(New_mf,0);

// free_fmatrix(Mh,0,500,0);

free_Doubmatrix(IMAGE,0,500,0);

// free_fmatrix(Z,0,64,0);

free_D3fmatrix(B,0,40,40);

infile2.close();

cout<<"ÑND"<<LF;

// Program for calculation

long double vijeta=0.0;

```

```

for(int j3=0;j3<order;j3++)
{
    for(int k3=0;k3<order;k3++)
    {
        if((j3==0)&&(k3==0))
        {
            iterate_moment1(Mh, Mg, restore_image);
            //cout<<"Hello";
            //cout<<"Image is ";
            //cout<<restore_image[0][0];
        }
        else
        {
            restore_image[j3][k3]=iterate_moment(Mh, Mg, j3, k3, restore_image,New_mf,vijeta);
            //cout<<"GOLA";
        }
    }
}

//Write_Into_disc(New_mf);
for(i=0;i<order;i++)
{
    for(j=0;j<order;j++)
    {
        //restore_image2[i][j]=Mf[i][j]-restore_image[i][j];
        //restore_image3[i][j]=restore_image2[i][j]/Mf[i][j];
        cout<<" "<<restore_image[i][j];
    }
    cout<<"\n";
}

free_Doubmatrix(Mf,0,1000,0);
free_Doubmatrix(Mg,0,1000,0);
free_Doubmatrix(Mh,0,500,0);
free_Doubmatrix(restore_image,0,500,0);

/*
for(i=0;i<200;i++)
{
    cout<<New_mf[i]<<"\n";
}

```

```

}

*/

}

// Convert data to matrix form

void vector2matrix(long double *TEMP,long double **IMAGE,int XSIZE,int YSIZE)
{
    int i,j,k;

    float vector_k;

    for (i=0;i<YSIZE;i++)
    {
        for(j=0;j<XSIZE;j++)
        {
            IMAGE[i][j]=0;
        }
    }

    k=0;

    for (i=0;i<YSIZE;i++)
    {
        for(j=0;j<XSIZE;j++)
        {
            vector_k=TEMP[k];

            IMAGE[i][j]=vector_k;

            k=k+1;
        }
    }
}

// Writing the file to output

void Write_Into_disc(float **Output)
{
    int i,j;

    char process_data[15];

    cout<<"Writing to disc\t"<<LF;

    cout<<Output[0][1]<<"writ\n";

    char filename[25],*s=".txt";

    cout<<"Enter file name"<<LF;

    cin>>process_data;

    strcpy(filename,"E:");

```

```

strcat(filename,process_data);

strcat(filename,s);

puts(filename);

getch();

ofstream outfile(filename);

if(!outfile)

cout<<"cannot open file"<<LF;

for(int i=0; i<IP;i++)

{

//for(int j=1;j<N;j++)

//{

outfile<<Output[i][1]<<SP;

//}

outfile<<LF;

}

outfile.close();

}

void moments( long double **f,int order, long double **mf)

{

//long double mf[32][32];

int i,j,p,q,y,x;

long double temp;

for(i=0;i<order;i++)

{

for(j=0;j<order;j++)

{

mf[i][j] = 0;

}

}

// Calculation of Moments

for(p=0;p<order;p++)

{

for(q=0;q<order;q++)

{

for(y=0;y<128;y++)

{

for(x=0;x<128;x++)

```

```

{
    if(x==0 || y==0)
    {
        mf[p][q]=mf[p][q]+ f[x][y];
    }
else
{
    mf[p][q]=mf[p][q]+ (long double) (pow(double(x),double(p))* pow(double(y),double(q)) * f[x][y]);
}
}

    }
// cout<<mf[p][q]<<" \n";
}

//cout<<"counter"<<p<<"\n";
//ak=ak+1;
}

// Transpose
for(p=0;p<order;p++)
{
    for(q=0;q<order;q++)
    {
        temp=mf[q][p];
        mf[p][q]=temp;
    }
}

}

// For Blurred Image Calculation form Mg and Mf
void b__moments( long double **Mf, long double **Mh, long double **Mg, int order)
{
    int i,j,p,q,y,x;
    long double alpha=0,beta=0,alpha1,beta1;
    for(i=0;i<order;i++)
    {
        for(j=0;j<order;j++)
        {
            Mg[i][j] = 0;
        }
    }
}

```



```

}

for(p=0;p<order;p++)
{
for(q=0;q<order;q++)
{
for(y=0;y<=p;y++)
{
for(x=0;x<=q;x++)
{

alpha1=nchoosek(p,y,alpha);
beta1=nchoosek(q,x,beta);

Mg[p][q]+=alpha1*beta1*Mh[p-y][q-x]*Mf[y][x];
}
}

//cout<<Mg[p][q]<<" ";
}

//cout<<"\n";
}

}

void moments_h( long double **f,int order, long double **mf)
{

//long double mf[32][32];
int i,j,p,q,y,x;
for(i=0;i<order;i++)
{
for(j=0;j<order;j++)
{

mf[i][j] = 0;

}

}

// Calculation of Moments

for(p=0;p<order;p++)
{
for(q=0;q<order;q++)
{

```

```

    for(y=0;y<5;y++)
    {
for(x=0;x<5;x++)
    {
        if(x==0 || y==0)
        {
            mf[p][q]=mf[p][q]+ f[x][y];
        }
    }
else
    {
        mf[p][q]=mf[p][q]+ (long double) (pow(double(x),double(p))* pow(double(y),double(q)) * f[x][y]);
    }
    }
    }
    }

// Transpose
for(p=0;p<order;p++)
{
    for(q=0;q<order;q++)
    {
        mf[p][q]=mf[q][p];
    }
}

#include<math.h>

void Stirling__Unsigned(double S[32][32],int order);
void Stirling__Signed(double S[32][32],int order)
{
    int n,k;

    Stirling__Unsigned(S,order);

    for(n=0;n<=order;n++)
    {
        for(k=0;k<=order;k++)
        {
            S[n][k] = S[n][k]*pow(-1,(n-k));

```

```

    }

}

}

// #include <gmpxx.h>

#include <iostream>

#include <math.h>

#include "psf.h"

using namespace std;

double nchoosek(int n,int k);

double factorial(int a);

void Stirling__Unsigned(double S[32][32],int order);

void Stirling__Signed(double S[32][32],int order);

void moments(long double mf[200][200],double f[256][256],int order, int N);

int main()

{

    int i,j,x,y,p,q,n,order;

    double f[32][32],a,b,c;

    long double mf[200][200];

    double f1[256][256];

    double S[32][32];

    //N=8;

    order=N-1;

    Stirling__Signed(S,order);

    //Stirling__Unsigned(S,order);

    /*for(i=0;i<=N-1;i++)

    {

        for(j=0;j<=N-1;j++)

        {

            cout<<S[i][j]<<"\t";

        }

        cout<<"\n";

    }*/

    //Reading an Image

    for(i=0;i<=N-1;i++)

    {

        for(j=0;j<=N-1;j++)

```

```

    {
        f1[i][j] = (double)I[i][j];
    }

}

for(i=0;i<=N-1;i++)
{
    for(j=0;j<=N-1;j++)
    {
        f[i][j] = 0;
    }

}

// Calculation of Moments
//cout<<"Enter the order of moments put (N-1";
for (int i=0;i<=order;i++)
{
    for(int j=0;j<=order;j++)
    {
        cout<<f1[i][j]<<"\t";
    }
    cout<<"\n";
}
cout<<"\n";
moments(mf,f1,order,N);
for(i=0;i<=N-1;i++)
{
    for(j=0;j<=N-1;j++)
    {
        cout<<mf[i][j]<<"\t";
    }
    cout<<"\n";
}

// Reconstruction
for (x=0;x<=N-1;x++)
{
    for(y=0;y<=N-1;y++)

```

```

{
    for(p=0;p<=N-1;p++)
    {
        for(q=0;q<=N-1;q++)
        {
            for(i=x;i<=N-1;i++)
            {
                for(j=y; j<=N-1;j++)
                {
                    //f[x][y]=f[x][y]+pow(-1.0,(double)(i+j-x-y))/(factorial(i)*factorial(j))*nchoosek(i,x)
                    *nchoosek(j,y)*S[i][p]*S[j][q]*mf[p][q];

                    a=pow(-1.0,(double)(i+j-x-y))/(factorial(i)*factorial(j));

                    // cout<<a<<endl;

                    b=nchoosek(i,x)*S[i][p];
                    //cout<<b<<endl;

                    c=nchoosek(j,y)*S[j][q];
                    //cout<<c<<endl;

                    f[x][y]=f[x][y]+a*b*c*mf[p][q];

                    cout<<b*c*mf[p][q]<<"\t";
                }
            }
        }
    }

    cout<<"\nValue of f is "<<endl;

    for (i=0;i<=N-1;i++)
    {
        for (j=0;j<=N-1;j++)
        {
            cout<<f[i][j]<<"\t";
        }

        cout<<"\n";
    }

    // Exporting to Matlab Code

    /*
    FILE *outfile;

```

```

        outfile = fopen("Ahlad.txt","w");

// let's say your variable is F and of float type
fprintf(outfile,"%d\n",N);
for(int i=0;i<N;i++){
for(int j=0;j<N;j++){
if(f[i][j]>255)
f[i][j] = 255;
if(f[i][j]<0)
f[i][j] = 0;
fprintf(outfile,"%0Ff\n",f[i][j]);
}
}
fclose(outfile);

*/
return(0);
}

%%%%%%%%%%%%Blind restoration using Alternate Minimization%%%%%%%%%%%%

clear all;

clc;

%For Original Input Images
y=imread('camera256.bmp');
%y = imresize(y,[541 541]);

%Check for the image for gray scale or not and convert it to double
[m,n,num_colors]=size(y);
if num_colors > 1
y=rgb2gray(y);
end
y=double(y);

sigma=4;
patch=7;

psf_ideal=(fspecial('gaussian',[patch patch],sigma));
blurr_image_ideal=imfilter(y,psf_ideal,'conv','replicate');
blurr_model=blurr_image_ideal;

l=1;

[image_sharp,psf_estimated,z]=Blind_De(blurr_model,patch);

subplot(1,3,1,colormap(gray));imagesc(blurr_model);

subplot(1,3,2,colormap(gray));imagesc(image_sharp);

```

```

subplot(1,3,3),surf(psf_estimated./max(max(psf_estimated)));

function [im,psf,z] = Blind_De(blurr_model,patch)

dt=0.1;

dt2=0.01;

alpha1=0.01;

alpha2=5;

ep_U = 0.00001;

ep_k = 0.01;

stepmax=4000;

[row, col]=size(blurr_model);

%Blurring the Image

blurred_ornal=blurr_model;

psf_ornal(patch,patch)=0;

psf_ornal(ceil(patch/2),ceil(patch/2))=1;

blurred_final=blurred_ornal;

psf_final=psf_ornal;

blurred_previous=0;

psf_previous=0;

for j=1:3

    m=0;

    fprintf('PSF step: %d Overall: %d \n',m,j);

    while((mean(mean(abs(psf_final-psf_previous)))>ep_k*dt2) &&(m~=stepmax))

        m=m+1;

    %    fprintf('PSF step: %d Overall: %d \n',m,j);

    psf_previous=psf_final;

    [reg]=FV(psf_final);

    blurred_norm=blurred_final./sum(sum(blurred_final));

    psf_increment=conv2(blurred_norm,psf_final,'same');

    psf_derivative=conv2(psf_previous,(psf_increment-blurred_norm),'same');

    psf_step=psf_final+dt2.*(alpha2*reg-psf_derivative);

    %    fprintf('psf step %d \n',psf_step);

    psf_final=psf_step;

    %    subplot(133); imagesc(psf_final); title('Original');

    %    colormap gray;

    %    drawnow;

    %assure positive filter

    for r=1:patch

```

```

        for c=1:patch
            psf_final(r,c)=max(psf_final(r,c),0);
        end
    end
end

i=0;

fprintf('Image step: %d Overall: %d \n',i,j);

while((mean(mean(abs(blurred_final-blurred_previous))))>ep_U*dt) &&(i~=stepmax))

    i=i+1;

%     fprintf('Image step: %d Overall: %d \n',i,j);

    z(i)=mean(mean(abs(blurred_final-blurred_previous)));

    fprintf('Image step: %d \n',i);

    blurred_previous=blurred_final;

    %blurred_replicate=padarray(blurred_final,[1 1],'replicate');

    [reg]=FV(blurred_previous);

    blurred_increment=imfilter(blurred_final,psf_final,'replicate','conv');

    derivative=imfilter((blurred_increment-blurred_orignal),psf_final,'replicate','conv');

    %Update Step

    blurred_step = blurred_final+dt.*(alpha1.*reg-derivative);

    blurred_final=blurred_step;

%     subplot(131); imagesc(blurred_orignal); title('Original');

%     subplot(132); imagesc(blurred_final); xlabel(['i=',num2str(i)]);

%     colormap gray;

%     drawnow;

%     subplot(133);

    if(j==3)

        hold on;

        plot(i,z(i),'--mo','linewidth',1);

        drawnow;

    end

    for r=1:row

        for c=1:col

            blurred_final(r,c)=max(blurred_final(r,c),0);

        end

    end

end

end

end

```



```

im=blurred_final;

psf=psf_final;

end

function [reg]=FV(u)

a = 0.01; %Fudge factor to avoid division by zero.

[m,n] = size(u);

u_x = (u(:,[2:n,n]) - u(:,[1,1:n-1]))/2;
u_y = (u([2:m,m],:) - u([1,1:m-1],:))/2;
u_xx = u(:,[2:n,n]) - 2*u + u(:,[1,1:n-1]);
u_yy = u([2:m,m],:) - 2*u + u([1,1:m-1],:);
u_xy = ( u([2:m,m],[2:n,n]) + u([1,1:m-1],[1,1:n-1]) - u([1,1:m-1],[2:n,n]) - u([2:m,m],[1,1:n-1]) ) / 4;

%TV term = Num/Den

Num = u_xx.*(1+u_y.^2) - 2*u_x.*u_y.*u_xy + u_yy.*(1+u_x.^2);
Den = (1+u_x.^2 + u_y.^2).^(3/2);

reg = Num./Den;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Sigma Estimation using WGM %%%%%%%%%%

clear all; close all; clc;

loc = 'laptop';

load T.mat

addpath('D:\Simga_estimation_using_Tchebichef_moments_\CSIQ')
addpath('D:\Simga_estimation_using_Tchebichef_moments_\CIDIQ');

Nh = 7;

sig_start = 0.3;
sig_step = 0.1;
sig_stop = 4.0;
psf=[3:2:17];

image_count=23;

%image_count=30;

Lsig = length(sig_start:sig_step:sig_stop);

F = double(zeros(image_count*Lsig,4)); %change 6 to the number of feature vectors used

Target = repmat((sig_start:sig_step:sig_stop)',[image_count 1]);

t = tchebichef_polynomial(8,7);

filename = {'final01','final02','final03','final04','final05','final06','final07',
'final08','final09','final10','final11','final12','final13','final14','final15',
'final16','final17','final18','final19','final20','final21','final22','final23'};

```

```

count = 1;

for k=1:image_count

    Icolor = imread([filename{k} '.bmp']);

    %Icolor = imread([filename{k} '.png']);

    Iori = rgb2gray(Icolor);

    [M N] = size(Iori);

    N1=max(M,N);

    display(['Image: ' filename{k}])

    for sig = sig_start:sig_step:sig_stop

        F1 = [];

    index=randi([1,8],[1,1]);

        mask=psf(1,index);

        h = fspecial('gaussian',[mask mask],sig);

        Iblur = imfilter(Iori,h,'same');

        h1 = fspecial('gaussian',[Nh Nh],1.0);

        Iblur1 = imfilter(Iblur,h1,'same');

        edge_no=1;

        for i=1:8:M-7

            for j=1:8:N-7

                if(i==1 || j==1 || i==M-7 || j==N-7)

                    else

                        Ib = double(Iblur(i:i+7,j:j+7));

                        GM2=T(1:8,1:8)*Ib*T(1:8,1:8)';

                        GM2=GM2';

                        Tblur(1,2)=(3*((3-N1)*GM2(1,1)+2*GM2(1,2)))/(N1*(N1^2-1));

                        Tblur(2,1)=(3*((3-N1)*GM2(1,1)+2*GM2(2,1)))/(N1*(N1^2-1));

                        Tblur(1,3)=(5*((N1-1)*(N1-2)*GM2(1,1)+6*(1-N1)*GM2(1,2)

+6*GM2(1,3)))/((N1^2-4)*(N1^2-1));

                        Tblur(3,1)=(5*((N1-1)*(N1-2)*GM2(1,1)+6*(1-N1)*GM2(2,1)

+6*GM2(3,1)))/((N1^2-4)*(N1^2-1));

                        Tblur(1,4)=(7*N1*(-1*(N1-1)*(N1-2)*(N1-3)*GM2(1,1)+

2*(N1-3)*(6*N1-53)*GM2(1,2)+(150-30*N1)*GM2(1,3)+

20*GM2(1,4)))/((N^2-9)*(N1^2-4)*(N1^2-1));

                        Tblur(4,1)=(7*N1*(-1*(N1-1)*(N1-2)*(N1-3)*GM2(1,1)+

2*(N1-3)*(6*N1-53)*GM2(2,1)+(150-30*N1)*GM2(3,1)+

20*GM2(4,1)))/((N^2-9)*(N1^2-4)*(N1^2-1));

                        Tblur(1,5)=(9*N1^2*((N1-1)*(N1-2)*(N1-3)*(N1-4)*GM2(1,1)+(5*(N1-3)*(17-4*N1)-

```

```

70*(4*N1-10))*GM2(1,2)+(45*(N1-3)*(N1-4)-420*(N1-1)+770)*GM2(1,3)+(560-140*N1)*GM2(1,4)+
70*GM2(1,5))/((N1^2-16)*(N1^2-9)*(N1^2-4)*(N1^2-1));

Tblur(5,1)=(9*N1^2*((N1-1)*(N1-2)*(N1-3)*(N1-4)*GM2(1,1)+(5*(N1-3)*(17-4*N1)-
70*(4*N1-10))*GM2(2,1)+(45*(N1-3)*(N1-4)-420*(N1-1)+770)*GM2(3,1)+(560-140*N1)*GM2(4,1)+
70*GM2(5,1)))/((N1^2-16)*(N1^2-9)*(N1^2-4)*(N1^2-1));

[Blc_type,Tblur,edgebox_count] = TM_Black_Classification(Ib,t,edge_no);

if Blc_type ==2

Ireblur1 = double(Iblur1(i:i+7,j:j+7));

GM=T(1:8,1:8)*Ireblur1*T(1:8,1:8)';

%GM=moments_2D(Ireblur1,4);

GM=GM';

Treblur(1,2)=(3*((3-N1)*GM(1,1)+2*GM(1,2)))/(N1*(N1^2-1));

Treblur(2,1)=(3*((3-N1)*GM(1,1)+2*GM(2,1)))/(N1*(N1^2-1));

Treblur(1,3)=(5*((N1-1)*(N1-2)*GM(1,1)+6*(1-N1)*GM(1,2)+6*GM(1,3)))/((N1^2-4)*(N1^2-1));

Treblur(3,1)=(5*((N1-1)*(N1-2)*GM(1,1)+6*(1-N1)*GM(2,1)+6*GM(3,1)))/((N1^2-4)*(N1^2-1));

Treblur(1,4)=(7*N1*(-1*(N1-1)*(N1-2)*(N1-3)*GM(1,1)+2*(N1-3)*(6*N1-53)*GM(1,2)+
(150-30*N1)*GM(1,3)+20*GM(1,4)))/((N1^2-9)*(N1^2-4)*(N1^2-1));

Treblur(4,1)=(7*N1*(-1*(N1-1)*(N1-2)*(N1-3)*GM(1,1)+2*(N1-3)*(6*N1-53)*GM(2,1)+
(150-30*N1)*GM(3,1)+20*GM(4,1)))/((N1^2-9)*(N1^2-4)*(N1^2-1));

Treblur(1,5)=(9*N1^2*((N1-1)*(N1-2)*(N1-3)*(N1-4)*GM(1,1)+
(5*(N1-3)*(17-4*N1)-70*(4*N1-10))*GM(1,2)+
(45*(N1-3)*(N1-4)-420*(N1-1)+770)*GM(1,3)
+(560-140*N1)*GM(1,4)+70*GM(1,5)))/((N1^2-16)*(N1^2-9)*(N1^2-4)*(N1^2-1));

Treblur(5,1)=(9*N1^2*((N1-1)*(N1-2)*(N1-3)*(N1-4)*GM(1,1)+
(5*(N1-3)*(17-4*N1)-70*(4*N1-10))*GM(2,1)+
(45*(N1-3)*(N1-4)-420*(N1-1)+770)*GM(3,1)+
(560-140*N1)*GM(4,1)+70*GM(5,1)))/((N1^2-16)*(N1^2-9)*(N1^2-4)*(N1^2-1));

Freblur = [sqrt(Treblur(2,1)^2+Treblur(1,2)^2) sqrt(Treblur(3,1)^2
+Treblur(1,3)^2) sqrt(Treblur(4,1)^2+Treblur(1,4)^2)
sqrt(Treblur(5,1)^2+Treblur(1,5)^2)];

Fblur = [sqrt(Tblur(2,1)^2+Tblur(1,2)^2) sqrt(Tblur(3,1)^2+Tblur(1,3)^2)
sqrt(Tblur(4,1)^2+Tblur(1,4)^2) sqrt(Tblur(5,1)^2+Tblur(1,5)^2)];

F1 = [F1; abs(abs(Fblur)-abs(Freblur))];

end

end

end

end

```

```

        F(count,:) = mean(F1);

        count = count+1;

    end

end

save TM_Not_Normalized.mat F Target

%%ELM

function [TrainingTime, TestingTime, TrainingAccuracy, TestingAccuracy, TY] = elm(TrainingData_File,
TestingData_File, Elm_Type, NumberofHiddenNeurons, ActivationFunction,C)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Macro definition

REGRESSION=0;

CLASSIFIER=1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Load training dataset

train_data=TrainingData_File;

T=train_data(:,1)';

P=train_data(:,2:size(train_data,2))';

clear train_data;                                % Release raw training data array

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Load testing dataset

test_data=TestingData_File;

TV.T=test_data(:,1)';

TV.P=test_data(:,2:size(test_data,2))';

clear test_data;                                % Release raw testing data array

NumberofTrainingData=size(P,2);

NumberofTestingData=size(TV.P,2);

NumberofInputNeurons=size(P,1);

if Elm_Type~=REGRESSION

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Preprocessing the data of classification

    sorted_target=sort(cat(2,T,TV.T),2);

    label=zeros(1,1);                            % Find and save in 'label' class label from training and testing data sets

    label(1,1)=sorted_target(1,1);

    j=1;

    for i = 2:(NumberofTrainingData+NumberofTestingData)

        if sorted_target(1,i) ~= label(1,j)

            j=j+1;

            label(1,j) = sorted_target(1,i);

        end

    end

end

number_class=j;

```

```

NumberOfOutputNeurons=number_class;

%%%%%%%%%%%% Processing the targets of training

temp_T=zeros(NumberOfOutputNeurons, NumberOfTrainingData);

for i = 1:NumberOfTrainingData

    for j = 1:number_class

        if label(1,j) == T(1,i)

            break;

        end

    end

    temp_T(j,i)=1;

end

T=temp_T*2-1;

%%%%%%%%%%%% Processing the targets of testing

temp_TV_T=zeros(NumberOfOutputNeurons, NumberOfTestingData);

for i = 1:NumberOfTestingData

    for j = 1:number_class

        if label(1,j) == TV.T(1,i)

            break;

        end

    end

    temp_TV_T(j,i)=1;

end

TV.T=temp_TV_T*2-1;

end                                     % end if of Elm_Type

%%%%%%%%%%%% Calculate weights & biases

start_time_train=cputime;

%%%%%%%%%%%% Random generate input weights InputWeight (w_i) and biases

% BiasofHiddenNeurons (b_i) of hidden neurons

InputWeight=rand(NumberOfHiddenNeurons,NumberOfInputNeurons)*2-1;

BiasofHiddenNeurons=rand(NumberOfHiddenNeurons,1);

tempH=InputWeight*P;

clear P;                               % Release input of training data

ind=ones(1,NumberOfTrainingData);

BiasMatrix=BiasofHiddenNeurons(:,ind);    % Extend the bias matrix BiasofHiddenNeurons to

% match the demention of H

tempH=tempH+BiasMatrix;

```

```

%%%%%%%%%%%%%% Calculate hidden neuron output matrix H

switch lower(ActivationFunction)

    case {'sig','sigmoid'}

        %%%%%%%%%%% Sigmoid

        H = 1 ./ (1 + exp(-tempH));

    case {'sin','sine'}

        %%%%%%%%%%% Sine

        H = sin(tempH);

    case {'hardlim'}

        %%%%%%%%%%% Hard Limit

        H = double(hardlim(tempH));

    case {'tribas'}

        %%%%%%%%%%% Triangular basis function

        H = tribas(tempH);

    case {'radbas'}

        %%%%%%%%%%% Radial basis function

        H = radbas(tempH);

        %%%%%%%%%%% More activation functions can be added here

end

clear tempH; % Release the temporary array for calculation of hidden neuron output matrix H

%%%%%%%%%%%%%% Calculate output weights OutputWeight (beta_i)

% OutputWeight=pinv(H') * T';

OutputWeight=(eye(size(H,1))/C+H * H') \ H * T';

end_time_train=cputime;

TrainingTime=end_time_train-start_time_train;

% Calculate CPU time (seconds) spent for training ELM

%%%%%%%%%%%%%% Calculate the training accuracy

Y=(H' * OutputWeight)'; % Y: the actual output of the training data

if Elm_Type == REGRESSION

    %TrainingAccuracy=sqrt(mse(T - Y))

    % Calculate training accuracy (RMSE) for regression case

    Zd= corrcoef(T,Y);

    TrainingAccuracy= abs(Zd(1,2));

end

clear H;

%%%%%%%%%%%%%% Calculate the output of testing input

start_time_test=cputime;

```

```

tempH_test=InputWeight*TV.P;

clear TV.P;          % Release input of testing data

ind=ones(1,NumberOfTestingData);

BiasMatrix=BiasofHiddenNeurons(:,ind);

% Extend the bias matrix BiasofHiddenNeurons to match the dimentions of H

tempH_test=tempH_test + BiasMatrix;

switch lower(ActivationFunction)

    case {'sig','sigmoid'}

        %%%%%%%%% Sigmoid

        H_test = 1 ./ (1 + exp(-tempH_test));

    case {'sin','sine'}

        %%%%%%%%% Sine

        H_test = sin(tempH_test);

    case {'hardlim'}

        %%%%%%%%% Hard Limit

        H_test = hardlim(tempH_test);

    case {'tribas'}

        %%%%%%%%% Triangular basis function

        H_test = tribas(tempH_test);

    case {'radbas'}

        %%%%%%%%% Radial basis function

        H_test = radbas(tempH_test);

        %%%%%%%%% More activation functions can be added here

end

TY=(H_test' * OutputWeight)';          % TY: the actual output of the testing data

end_time_test=cputime;

TestingTime=end_time_test-start_time_test;          % Calculate CPU time (seconds) spent

% by ELM predicting the whole testing data

if Elm_Type == REGRESSION

    %TestingAccuracy=sqrt(mse(TV.T - TY))          % Calculate testing accuracy (RMSE) for regression case

    Zd2= median(corrcoef(TV.T,TY));          %my chnage

    TestingAccuracy= abs(Zd2(1,2));          %my change

end

if Elm_Type == CLASSIFIER

    %%%%%%%%% Calculate training & testing classification accuracy

    MissClassificationRate_Training=0;

    MissClassificationRate_Testing=0;

```

```

for i = 1 : size(T, 2)

    [x, label_index_expected]=max(T(:,i));

    [x, label_index_actual]=max(Y(:,i));

    if label_index_actual~=label_index_expected

        MissClassificationRate_Training=MissClassificationRate_Training+1;

    end

end

TrainingAccuracy=1-MissClassificationRate_Training/size(T,2)

for i = 1 : size(TV.T, 2)

    [x, label_index_expected]=max(TV.T(:,i));

    [x, label_index_actual]=max(TY(:,i));

    if label_index_actual~=label_index_expected

        MissClassificationRate_Testing=MissClassificationRate_Testing+1;

    end

end

TestingAccuracy=1-MissClassificationRate_Testing/size(TV.T,2)

end

%%%%%%%%%%%%%%Sigma Estimation using TM

%VLC with LIVE

clear all; close all; clc;

% loc = 'home';

loc = 'laptop';

addpath('D:\Sigma_estimation_using_Tchebichef_moments_Ahlab\VLC');

addpath('D:\Sigma_estimation_using_Tchebichef_moments_Ahlab\Caltech');

Nh = 7;

sig_start = 0.3;

sig_step = 0.1;

sig_stop = 4.0;

Lsig = length(sig_start:sig_step:sig_stop);

psf=3:3:17;

images=23;

F = double(zeros(images*Lsig,6)); %change 6 to the number of feature vectors used

Target = repmat((sig_start:sig_step:sig_stop)',[images 1]);

t = tchebichef_polynomial(8,7);

dir_image=dir('D:\Sigma_estimation_using_Tchebichef_moments_Ahlab\Berkely\test');

[im_num,im_n]=size(dir_image);

count=1;

```



```

imc=1;
for k=3:23
    im_route=['D:\Sigma_estimation_using_Tchebichef_moments_Ahlab\Berkely\test\'dir_image(k).name];
    Icolor=imread(im_route);

    [m1,n1,num_colors]=size(Icolor);
    if num_colors > 1
        Iori = rgb2gray(Icolor);
    end
    [M N] = size(Iori);
    fprintf('Image: %d \n', imc)
    imc=imc+1;
    for sig = sig_start:sig_step:sig_stop
        F1 = [];
        index=randi([1,8],[1,1]);
        mask=psf(1,index);
        h = fspecial('gaussian',[mask mask],sig);
        Iblur = imfilter(Iori,h,'same');
        h1 = fspecial('gaussian',[Nh Nh],0.5);
        Iblur1 = imfilter(Iblur,h1,'same');
        edge_no=1;
        for i=1:8:M-7
            for j=1:8:N-7
                if(i==1 || j==1 || i==M-7 || j==N-7)
                    else
                        Ib = double(Iblur(i:i+7,j:j+7));
                        [Blc_type,Tblur,edgebox_count] = TM_Black_Classification(Ib,t,edge_no);
                        if Blc_type ==2
                            Ireblur1 = double(Iblur1(i:i+7,j:j+7));
                            Treblur = t*Ireblur1*t';
                            Freblur = [sqrt(Treblur(2,1)^2+Treblur(1,2)^2) sqrt(Treblur(3,1)^2+
Treblur(1,3)^2) sqrt(Treblur(4,1)^2+Treblur(1,4)^2) sqrt(Treblur(5,1)^2+
Treblur(1,5)^2) sqrt(Treblur(6,1)^2+
Treblur(1,6)^2) sqrt(Treblur(7,1)^2+Treblur(1,7)^2)];
                            Fblur = [sqrt(Tblur(2,1)^2+Tblur(1,2)^2) sqrt(Tblur(3,1)^2+Tblur(1,3)^2) sqrt(Tblur(4,1)^2+
Tblur(1,4)^2) sqrt(Tblur(5,1)^2+Tblur(1,5)^2)
sqrt(Tblur(6,1)^2+Tblur(1,6)^2) sqrt(Tblur(7,1)^2+Tblur(1,7)^2)];

```

```

        F1 = [F1; abs(Freblur)];

    end

end

end

end

F(count,:) = mean(F1);

count = count+1;

end

end

save TM_Not_Normalized1.mat F Target

Nh = 7;

sig_start = 0.3;

sig_step = 0.1;

sig_stop = 4.0;

Lsig = length(sig_start:sig_step:sig_stop);

images=23;

F2 = double(zeros(images*Lsig,6)); %change 6 to the number of feature vectors used

Target2 = repmat((sig_start:sig_step:sig_stop)',[images 1]);

t = tchebichef_polynomial(8,7);

dir_image=dir('D:\Sigma_estimation_using_Tchebichef_moments_Ahlab\Caltech');

[im_num,im_n]=size(dir_image);

count=1;

imc=1;

for k=3:23

    im_route=['D:\Sigma_estimation_using_Tchebichef_moments_Ahlab\Caltech\'',dir_image(k).name];

    Icolor=imread(im_route);

    [m1,n1,num_colors]=size(Icolor);

    if num_colors > 1

        Iori = rgb2gray(Icolor);

    end

    [M N] = size(Iori);

    fprintf('Image: %d \n', imc)

    imc=imc+1;

    for sig = sig_start:sig_step:sig_stop

        F1 = [];

        index=randi([1,8],[1,1]);

```

```

mask=psf(1,index);

h = fspecial('gaussian',[mask mask],sig);

Iblur = imfilter(Iori,h,'same');

h1 = fspecial('gaussian',[Nh Nh],0.5);

Iblur1 = imfilter(Iblur,h1,'same');

edge_no=1;

for i=1:8:M-7

    for j=1:8:N-7

        if(i==1 || j==1 || i==M-7 || j==N-7)

            else

                Ib = double(Iblur(i:i+7,j:j+7));

                [Blc_type,Tblur,edgebox_count] = TM_Blck_Classification(Ib,t,edge_no);

                if Blc_type ==2

                    Ireblur1 = double(Iblur1(i:i+7,j:j+7));

                    Treblur = t*Ireblur1*t';

                    Freblur = [sqrt(Treblur(2,1)^2+Treblur(1,2)^2) sqrt(Treblur(3,1)^2

+Treblur(1,3)^2) sqrt(Treblur(4,1)^2+

Treblur(1,4)^2) sqrt(Treblur(5,1)^2+Treblur(1,5)^2)

sqrt(Treblur(6,1)^2+Treblur(1,6)^2) sqrt(Treblur(7,1)^2+Treblur(1,7)^2)];

                    Fblur = [sqrt(Tblur(2,1)^2+Tblur(1,2)^2)

sqrt(Tblur(3,1)^2+Tblur(1,3)^2) sqrt(Tblur(4,1)^2+

Tblur(1,4)^2) sqrt(Tblur(5,1)^2+Tblur(1,5)^2)

sqrt(Tblur(6,1)^2+Tblur(1,6)^2) sqrt(Tblur(7,1)^2+Tblur(1,7)^2)];

                    F1 = [F1; abs(Freblur)];

                end

            end

        end

    end

    F2(count,:) = mean(F1);

    count = count+1;

end

end

save TM__Not__Normalized2.mat F2 Target2

%%%%%%MATHEMATICA CODE FOR SUBTRACTOR

input3 = Import[

    "D:\\Dropbox\\Mathematica Codes_Subtractor_PhD\\Final_3rd \\

version\\lena256.png"];

```

```

input4 = Floor[255*ImageData[input3]];

M1 = 256;

M2 = 256;

lev = 0;

y = ConstantArray[0.0, {M1, M2, M2}];

T = ConstantArray[0.0, {M1, M2, M2}];

inte = ConstantArray[0.0, {M1, M2}];

Dig = ConstantArray[0.0, {M1, M2}];

input = ConstantArray[0.0, {M1, M2}];

Dignew = ConstantArray[0.0, {M1, M2}];

secinput = ConstantArray[0.0, {1, M2}];

newinput = ConstantArray[0.0, {M1, M2}];

level = ConstantArray[0, {M2, M2, M2}];

level2 = ConstantArray[0, {M2, M2, M2 + 1}];

level3 = ConstantArray[0, {M2, M2, M2}];

level4 = ConstantArray[0, {M2, M2}];

thirdinput = ConstantArray[0, {M2, M2, M2}];

xval = ConstantArray[0, {1, (M1 - 1)*(M1 - 2)/2}];

yval = ConstantArray[0, {1, (M1 - 1)*(M1 - 2)/2}];

xval2 = ConstantArray[0, {1, M1}];

yval2 = ConstantArray[0, {1, M1}];

temp = ConstantArray[0, {1, M1}];

For[i1 = 1, i1 <= M1, i1++,
  For[j1 = 1, j1 <= M1, j1++,
    input[[i1, j1]] = input4[[M1 - i1 + 1, M1 - j1 + 1]]
  ]
]

Image[input4/255]

For[horiz = 1, horiz <= M2, horiz++,
  Print[horiz]

  If[horiz == 1,
    For[i = 1, i <= M1, i++,
      For[j = 1, j <= M2, j++,
        If[j == 1,
          y[[horiz, i, j]] = input[[i, j]],
          y[[horiz, i, j]] = input[[i, j]] + y[[horiz, i, j - 1]]
        ]
      ]
    ]
  ]
]

```



```

For[i1 = 1, i1 <= M1, i1++,
  Dig[[i1, 1 ;; M1]] = T[[i1, M1, 1 ;; M1]]
]

Dignew = Transpose[Dig];

For[i1 = 1, i1 <= M1, i1++,
  For[j1 = 1, j1 <= M1, j1++,
    newinput[[i1, j1]] = Dignew[[M1 - i1 + 1, M1 - j1 + 1]]
  ]
]

"Second stage completed"

For[i = 1, i <= M1, i++,
  secinput[[1, 1 ;; M1]] = newinput[[i, 1 ;; M1]];
  lev = M1 - 1;
  level[[i, 1, 1 ;; M1]] = secinput[[1, 1 ;; M1]];
  For[j1 = 2, j1 <= M1, j1++,
    For[j2 = 1, j2 <= M1 - (j1 - 1), j2++,
      level[[i, j1, j2]] =
        level[[i, j1 - 1, j2]] - level[[i, j1 - 1, j2 + 1]];
    ]
  ]
]

For[i = 1, i <= M1, i++,
  level[[i, M1, 1 ;; M1]] = level[[i, M1, 1]]
]

count = 1;

For[j = 1, j <= M1, j++,
  For[i = 1, i <= M1, i++,
    If[level[[1, i, j]] == 0,
      xval[[1, count]] = i;
      yval[[1, count]] = j;
      count = count + 1;
    ]
  ]
]

range = (M1 - 1)*(M1 - 2)/2;

For[k = 1, k <= M1, k++,
  For[i = 1, i <= range - 1, i++,

```

```

If[xval[[1, i]] != M1 && yval[[1, i]] != 1,
    level[[k, xval[[1, i]], yval[[1, i]]] =
    level[[k, xval[[1, i]], yval[[1, i]] - 1]] -
    level[[k, xval[[1, i]] + 1, yval[[1, i]] - 1]]
]
]
]
For[k = 1, k <= M1, k++,
    If[xval[[1, range]] != M1 || yval[[1, range]] != M1,
        level[[k, xval[[1, range]], yval[[1, range]]] =
        level[[k, xval[[1, range]], yval[[1, range]] - 1]] -
        level[[k, xval[[1, range]] + 1, yval[[1, range]] - 1]]
    ]
]
xzeros = ConstantArray[0, {M1, 1}];
For [k = 1, k <= M1, k++,
    For[i = 1, i <= M1, i++,
        For[j = 1, j <= M1, j++,
            level2[[k, i, j]] = level[[k, i, j]]
        ]
    ]
]
count = 1;
For[j = 1, j <= M1 + 1, j++,
    For[i = 1, i <= M1 + 1, i++,
        If[level2[[1, i, j]] == 0,
            xval2[[1, count]] = i;
            yval2[[1, count]] = j;
            count = count + 1;
        ]
    ]
]
For[k = 1, k <= M1, k++,
    level2[[k, M1, M1 + 1]] = level2[[k, M1, 1]]
]
For[k = 1, k <= M1, k++,
    For[i = 1, i <= M1 - 1, i++,

```

```

        level2[[k, i, M1 + 1]] =
        level2[[k, i, M1]] - level2[[k, i + 1, M1]]
    ]
]
For[j = 1, j <= M1, j++,
    cnt = M1;
    For[i = 1, i <= M1, i++,
        level3[[i, j]] = level2[[j, cnt, M1 + 1]];
        cnt = cnt - 1;
    ]

]

"Third stage completed"
For[i = 1, i <= M1, i++,
    secinput[[1, 1 ;; M1]] = level3[[i, 1 ;; M1]];
    lev = M1 - 1;
    level[[i, 1, 1 ;; M1]] = secinput[[1, 1 ;; M1]];
    For[j1 = 2, j1 <= M1, j1++,
        For[j2 = 1, j2 <= M1 - (j1 - 1), j2++,
            level[[i, j1, j2]] =
            level[[i, j1 - 1, j2]] - level[[i, j1 - 1, j2 + 1]];
        ]
    ]
]

For[i = 1, i <= M1, i++,
    level[[i, M1, 1 ;; M1]] = level[[i, M1, 1]]
]

count = 1;

For[j = 1, j <= M1, j++,
    For[i = 1, i <= M1, i++,
        If[level[[1, i, j]] == 0,
            xval[[1, count]] = i;
            yval[[1, count]] = j;
            count = count + 1;
        ]
    ]
]

```



```

range = (M1 - 1)*(M1 - 2)/2;
For[k = 1, k <= M1, k++,
  For[i = 1, i <= range - 1, i++,
    If[xval[[1, i]] != M1 && yval[[1, i]] != 1,
      level[[k, xval[[1, i]], yval[[1, i]]] =
        level[[k, xval[[1, i]], yval[[1, i]] - 1]] -
        level[[k, xval[[1, i]] + 1, yval[[1, i]] - 1]];
    ]
  ]
]
For[k = 1, k <= M1, k++,
  If[xval[[1, range]] != M1 || yval[[1, range]] != M1,
    level[[k, xval[[1, range]], yval[[1, range]]] =
      level[[k, xval[[1, range]], yval[[1, range]] - 1]] -
      level[[k, xval[[1, range]] + 1, yval[[1, range]] - 1]];
  ]
]
xzeros = ConstantArray[0, {M1, 1}];
For [k = 1, k <= M1, k++,
  For[i = 1, i <= M1, i++,
    For[j = 1, j <= M1, j++,
      level2[[k, i, j]] = level[[k, i, j]];
    ]
  ]
]
count = 1;
For[j = 1, j <= M1 + 1, j++,
  For[i = 1, i <= M1 + 1, i++,
    If[level2[[1, i, j]] == 0,
      xval2[[1, count]] = i;
      yval2[[1, count]] = j;
      count = count + 1;
    ]
  ]
]
For[k = 1, k <= M1, k++,
  level2[[k, M1, M1 + 1]] = level2[[k, M1, 1]];

```

```

    ]
    For[k = 1, k <= M1, k++,
        For[i = 1, i <= M1 - 1, i++,
            level2[[k, i, M1 + 1]] =
                level2[[k, i, M1]] - level2[[k, i + 1, M1]];
        ]
    ]
    For[k = M1, k >= 1, k--,
        For [t = 1, t <= M1, t++,
            temp[[1, t]] = level2[[k, t, M1 + 1]];
        ]
        For[j = 1, j <= M1, j++,
            level4[[M1 - k + 1, j]] = temp[[1, j]];
        ]
    ]
    Image[level4/255]
    Clear[y];
    Clear[T];
    Clear[intel];
    Clear[Dig];
    Clear[Dignew];
    Clear[secinput];
    Clear[newinput];
    Clear[thirdinput];
    Clear[level];
    Clear[level2];
    Clear[level3];
    Clear[level4];
    Clear[xval];
    Clear[yval];
    Clear[xval2];
    Clear[yval2];
    Clear[temp];
    TimeUsed[]

%%Subtractor in C++

//2 dimension complete cycle (Input-DigitalFilter-Output-GeometricMoment)

#include <iostream>

```

```

#include <math.h>

#include <gmp.h>

#include <time.h>

#include "lena64.h"

#include "invcf_64.h"

void mpf_array_clear(mpf_t*,int,int);

void mpf_array_init(mpf_t*,int,int);

void mpz_array_clear(mpz_t*,int,int);

void mpf_array_clear(mpf_t *input, int a, int b){
    for(int i=0;i<a;i++){
        for(int j=0;j<b;j++){
            mpf_set_d(*(input+j+i*b),0);
        }
    }
}

void mpf_array_init(mpf_t *input, int a, int b){
    for(int i=0;i<a;i++){
        for(int j=0;j<b;j++){
            mpf_init(*(input+j+i*b));
        }
    }
}

void mpz_array_clear(mpz_t *input, int a,int b){
    for(int i=0;i<a;i++){
        for(int j=0;j<b;j++){
            mpz_clear(*(input+j+i*b));
        }
    }
}

int main(){
    clock_t t_start,t_end;

    int order = N-1;

    mpf_t sz;

    mpf_init(sz);

    mpf_set_ui(sz,(unsigned short int)N);

    std::cout<<"\n\n\tImage size: " << (sz) << " by " << (sz);

    std::cout<<"\n\t";

```

```

std::cout<<"\n\n\t*****";

//=====

//Initialization

//=====

mpz_t y[N][N];          //for 1D output and Wong&Siu matrix
mpz_array_init(y[0][0],N*N,4096);
mpz_t yy[N][N];         //for 2D output and 2D input
mpz_array_init(yy[0][0],N*N,4096);

mpz_t cf;
mpz_init(cf);

mpz_t M[N][N];          //for GM
mpz_array_init(M[0][0],N*N,4096);
mpz_t MM[N][N];         //for GM and 1D input
mpz_array_init(MM[0][0],N*N,4096);
mpf_t MMZ[N][N];        //for GM in mpf
mpf_array_init(*MMZ,N,N);
mpf_t EE[N][N];
mpf_array_init(*EE,N,N);
mpf_t AA,BB,CC;
mpf_init(AA);
mpf_init(BB);
mpf_init(CC);
mpf_t RMM[N][N];
mpf_array_init(*RMM,N,N);
mpf_array_clear(*RMM,N,N);
mpz_t xoo[N][N+1];      //for subtractor
mpz_array_init(xoo[0][0],N*(N+1),4096);

//=====

//2-dimensional adder filter

//=====

for(int i=0;i<N;i++){
    for(int j=0;j<N;j++){
        mpz_add_ui(y[i][0],y[i][0],(unsigned long int)I[N-i-1][N-j-1]);
        for(int k=1;k<order+1;k++){
            mpz_add(y[i][k],y[i][k],y[i][k-1]);
        }
        for(int j=0;j<order+1;j++){

```

```

        mpz_add(yy[0][j],yy[0][j],y[i][j]);

        for(int k=1;k<order+1;k++)

            mpz_add(yy[k][j],yy[k][j],yy[k-1][j]);

    }

}

//*****

//Showing the output

// std::cout<<"\n\n\tDigital Filter Output:";

// std::cout<<"\n\t";

// for(int i=0;i<N;i++)

// {std::cout<<"\n\t";

//     for(int j=0;j<N;j++)

//         std::cout<<yy[i][j]<<" ";}

//=====

//The coefficient for Wong & Siu filter

//=====

for(int i=0;i<N;i++){

    {for(int j=0;j<N;j++)

        mpz_init(y[i][j]);}

    mpz_set_d(y[0][0],1.0);

    for(int i=1;i<order+1;i++){

        mpz_set_d(cf,-1.0);

        mpz_pow_ui(y[i][1],cf,(unsigned long int)(i-1));

        for(int j=2;j<i+1;j++){

            mpz_sub(cf,y[i-1][j-1],y[i-1][j]);

            mpz_mul_ui(y[i][j],cf,(unsigned long int)j);

        }

    }

}

//=====

//Geometric Moments calculation

//=====

for(int i=0;i<order+1;i++){

    for(int j=0;j<order+1;j++){

        for(int r=0;r<i+1;r++){

            {

                for(int s=0;s<j+1;s++)

                    mpz_addmul(M[i][j],y[j][s],yy[s][r]);

            }

        }

    }

}

```

```

        mpz_mul(M[i][j],y[i][r],M[i][j]);

        mpz_add(MM[i][j],MM[i][j],M[i][j]);

        mpz_init(M[i][j]);
    }

}

}

//*****

//Showing the geometric moment

// std::cout<<"\n\n\tGeometric Moments: ";

// std::cout<<"\n\t";

// for(int i=0;i<N;i++){

//     {std::cout<<"\n\t";

//         for(int j=0;j<N;j++){

//             std::cout<<MM[j][i]<<" ";}

//         =====

//Converting mpz to mpf - decimal to float

//=====

for(int i=0;i<N;i++){

    for(int j=0;j<N;j++){

        mpf_set_z(MMZ[i][j],MM[i][j]);}

//*****

//RECONSTRUCTION STARTS HERE =====

t_start = clock();

//=====

//Inverse Matrix Coefficient of Wong & Siu

//=====

for(int i=0;i<N;i++){

    for(int j=0;j<N;j++){

        mpf_set_d(EE[i][j],E[i][j]);}

//=====

//Digital Filter Outputs Calculation

//=====

for(int i=0;i<N;i++){

    for(int j=0;j<N;j++){

        for(int r=0;r<i+1;r++){

            mpf_set_d(BB,0);

            for(int s=0;s<j+1;s++){

```

```

        mpf_mul(AA,EE[j][s],MMZ[r][s]);

        mpf_add(BB,BB,AA);
    }

    mpf_mul(CC,EE[i][r],BB);
    mpf_add(RMM[i][j],RMM[i][j],CC);
}

}

//Showing the output
// std::cout<<"\n\n\tDigital Filter Output:";
// std::cout<<"\n\t";
// for(int i=0;i<N;i++)
// {std::cout<<"\n\t";
//     for(int j=0;j<N;j++)
//         std::cout<<RMM[j][i]<<" ";}

for(int i=0;i<N;i++)
    {for(int j=0;j<N;j++)
        mpz_init(M[i][j]);}

for(int i=0;i<N;i++)
    {for(int j=0;j<N;j++)
        mpz_init(MM[i][j]);}

//=====

//Converting mpf to mpz for digital filter output - float to decimal
//=====

for(int i=0;i<N;i++){
    for(int j=0;j<N;j++)
        mpz_set_f(M[i][j],RMM[i][j]);

    for(int i=0;i<N;i++){
        for(int j=0;j<N;j++)
            mpz_get_d(M[i][j]);}

// std::cout<<"\n\n\tDecimal";
// std::cout<<"\n\t";
// for(int i=0;i<N;i++)
// {std::cout<<"\n\t";
//     for(int j=0;j<N;j++)
//         std::cout<<M[j][i]<<" ";}

```

```

//=====

//Subtractor - Y direction
//=====

for(int cl=0;cl<N;cl++){
    for(int j=0;j<N;j++){
        mpz_set(xoo[0][j],yy[order-j][cl]);

        //upper triangle
        for(int i=1;i<N;i++){
            {for(int k=0;k<N-i;k++)
                mpz_sub(xoo[i][k],xoo[i-1][k],xoo[i-1][k+1]);}

            //last row
            {for(int p=1;p<N+1;p++)
                mpz_set(xoo[order][p],xoo[order][0]);}

            //lower triangle
            for(int p=2;p<N+1;p++){
                {for(int q=0;q<N-1;q++)
                    mpz_sub(xoo[q][p],xoo[q][p-1],xoo[q+1][p-1]);

                    {for(int n=0;n<N;n++)
                        mpz_set(MM[n][order-cl],xoo[n][order+1]);

                    }}}
        }
    }
    for(int i=0;i<N;i++){
        {for(int j=0;j<N;j++)
            mpz_init(xoo[i][j]);}
    }
    for(int i=0;i<N;i++){
        {for(int j=0;j<N;j++)
            mpz_init(y[i][j]);}
    }
    for(int i=0;i<N;i++){
        {for(int j=0;j<N;j++)
            mpz_init(yy[i][j]);}
    }
}

//=====

//Subtractor - X direction
//=====

for(int cl=0;cl<N;cl++){
    {for(int j=0;j<N;j++)
        mpz_set(xoo[0][j],MM[cl][j]);

        //upper triangle
        for(int i=1;i<N;i++){

```



```

        {for(int k=0;k<N-i;k++)

mpz_sub(xoo[i][k],xoo[i-1][k],xoo[i-1][k+1]);}

        //last row

        {for(int p=1;p<N+1;p++)

mpz_set(xoo[order][p],xoo[order][0]);}

//lower triangle

for(int p=2;p<N+1;p++)

    {for(int q=0;q<N-1;q++)

mpz_sub(xoo[q][p],xoo[q][p-1],xoo[q+1][p-1]);

        {for(int n=0;n<N;n++)

mpz_set(yy[cl][n],xoo[n][order+1]);

        }}}

t_end = clock();

//*****

//Showing the output

std::cout<<"\n\n\tThe Digital Filter Input:";

std::cout<<"\n\t";

for(int i=0;i<N;i++)

{std::cout<<"\n\t";

    for(int j=0;j<order+1;j++)

        std::cout<<yy[j][i]<<" ";}

//Timing of the process

std::cout<<"\n\n\tImage size: " << (sz) << " by " << (sz);

std::cout<<"\n\n\tTime taken to compute the 2-D Digital Filter Input is "

<< (double)(t_end-t_start)/(double)CLOCKS_PER_SEC << " seconds\n";

}

%%%%Split Brergman iterations%%%%

function [ deblur1,deblur2 ] = deconvbregman( img , kernel , numIterations)

%DECONVBREGMAN Summary of this function goes here

% Detailed explanation goes here

[h w c] = size(img);

opts.kernel_size = size(kernel,1);

opts.gamma_correct = 1.0;

opts.nb_lambda = 3000;

opts.nb_alpha = 1.0;

opts.use_ycbcr = 1;

yorig = img;

```

```

b = zeros(opts.kernel_size);

bhs = floor(size(b, 1)/2);

padsizes = bhs;

dx = [1 -1];

dy = dx';

if (opts.use_ycbcr)

    if (size(yorig, 3) == 3)

        ybcr = rgb2ycbcr(yorig);

    else

        ybcr = yorig;

    end;

    opts.nb_alpha = 1;

end;

if (nargout == 2)

    kernels = [];

    kernels(:, :, 1) = kernel;

    kernels(:, :, 2) = rot90(kernel, 2);

    numKernels = size(kernels, 3);

    ppm = ParforProgMon( 'Deconvolution' , numKernels*numIterations);

else

    kernels = kernel;

    ppm = [];

end

numKernels = size(kernels, 3);

deblurImages = cell(numKernels, 1);

parfor q=1:size(kernels, 3)

    kernel = kernels(:, :, q);

    deblur = zeros(h, w, c);

    if (opts.use_ycbcr == 1)

        ypad = padarray(ybcr(:, :, 1), [padsizes padsizes], 'replicate', 'both');

        for a = 1:4

            ypad = edgetaper(ypad, kernel);

        end;

    tmp = fast_deconv_bregman(ypad, kernel, opts.nb_lambda, opts.nb_alpha, numIterations, ppm);

    deblur(:, :, 1) = tmp(bhs + 1 : end - bhs, bhs + 1 : end - bhs);

    if (size(ybcr, 3) == 3)

        deblur(:, :, 2:3) = ybcr(:, :, 2:3);

```

```

        deblur = ycbcr2rgb(deblur);

    end;

else
    for j = 1:3

        ypad = padarray(yorig(:, :, j), [1 1] * bhs, 'replicate', 'both');

        for a = 1:4

            ypad = edgetaper(ypad, kernel);

        end;

        tmp = fast_deconv_bregman(ypad, kernel, opts.nb_lambda, opts.nb_alpha, numIterations, ppm);

        deblur(:, :, j) = tmp(bhs + 1 : end - bhs, bhs + 1 : end - bhs);

    end;

end;

deblurImages{q} = deblur;

end

deblur1 = deblurImages{1};

if (nargout == 2)

    deblur2 = deblurImages{2};

end

end

function [g] = fast_deconv_bregman(f, k, lambda, alpha, outiter_max, progress)

beta = 400;

initer_max = 1;

[m n] = size(f);

% initialize

g = f;

% make sure k is odd-sized

if ((mod(size(k, 1), 2) ~= 1) || (mod(size(k, 2), 2) ~= 1))

    fprintf('Error - blur kernel k must be odd-sized.\n');

    return;

end;

ks = floor((size(k, 1)-1)/2);

dx = [1 -1];

dy = dx';

dxt = fliplr(flipud(dx));

dyt = fliplr(flipud(dy));

[Ktf, KtK, DtD, Fdx, Fdy, FN] = computeConstants(f, k, dx, dy);

gx = conv2(g, dx, 'valid');

```

```

gy = conv2(g, dy, 'valid');

fx = conv2(f, dx, 'valid');

fy = conv2(f, dy, 'valid');

ks = size(k, 1);

ks2 = floor(ks / 2);

% store some of the statistics

lcost = [];

pcost = [];

outiter = 0;

bx = zeros(size(gx));

by = zeros(size(gy));

wx = gx;

wy = gy;

totiter = 1;

gk = conv2(g, k, 'same');

lcost(totiter) = (lambda / 2) * norm(gk(:) - f(:))^2;

pcost(totiter) = sum((abs(gx(:)) .^ alpha));

pcost(totiter) = pcost(totiter) + sum((abs(gy(:)) .^ alpha));

for outiter = 1:outiter_max

%   fprintf('Outer iteration %d\n', outiter);

    initer = 0;

    for initer = 1:initer_max

        totiter = totiter + 1;

        if (alpha == 1)

            tmpx = beta * (gx + bx);

            betax = beta;

            tmpx = tmpx ./ betax;

            tmpy = beta * (gy + by);

            betay = beta;

            tmpy = tmpy ./ betay;

            betay = betay;

            wx = max(abs(tmpx) - 1 ./ betax, 0) .* sign(tmpx);

            wy = max(abs(tmpy) - 1 ./ betay, 0) .* sign(tmpy);

        else

            wx = solve_image_bregman(gx + bx, beta, alpha);

            wy = solve_image_bregman(gy + by, beta, alpha);

```

```

end

bx = bx - wx + gx;

by = by - wy + gy;

wx1 = conv2(wx - bx, dxt, 'full');

wy1 = conv2(wy - by, dyt, 'full');

tmp = zeros(size(g));

gprev = g;

gxprev = gx;

gyprev = gy;

num = lambda * Ktf + beta * fft2(wx1 + wy1);

denom = lambda * KtK + beta * DtD;

Fg = num ./ denom;

%   Fg = num .* FN ./ denom;

g = real(iff2(Fg));

gx = conv2(g, dx, 'valid');

gy = conv2(g, dy, 'valid');

gk = conv2(g, k, 'same');

%   imagesc(k);

%   pause(0.01);

lcost(totiter) = (lambda / 2) * norm(gk(:) - f(:))^2;

pcost(totiter) = sum((abs(gx(:)) .^ alpha));

pcost(totiter) = pcost(totiter) + sum((abs(gy(:)) .^ alpha));

if (~isempty(progress))

    progress.increment();

end

end;

end;

function [Ktf, KtK, DtD, Fdx, Fdy, FN] = computeConstants(f, k, dx, dy)

szK = size(k);

szR = floor(szK/2);

exponent = 2.5;

[xI,yI] = meshgrid(-szR(1):szR(1),-szR(2):szR(2));

D = 1+sqrt(xI.^2 + yI.^2);

FN = 1./(D.^exponent);

FN = FN / sum(FN(:));

FN = psf2otf(FN,size(f));

sizef = size(f);

```

```
otfk = psf2otf(k, sizef);  
Ktf = conj(otfk) .* fft2(f);  
KtK = abs(otfk) .^ 2;  
Fdx = abs(psf2otf(dx, sizef)).^2;  
Fdy = abs(psf2otf(dy, sizef)).^2;  
DtD = Fdx + Fdy;
```

University of Malaya