Development of Some Spatial-domain Pre-processing and Post-processing Algorithms for Better 2-D Up-scaling

Aditya Acharya

Department of Electronics and Communication Engineering

National Institute of Technology Rourkela (India)

Development of Some Spatial-domain Pre-processing and Post-processing Algorithms for Better 2-D Up-scaling

Dissertation submitted in partial fulfillment

of the requirements of the degree of

Doctor of Philosophy

in

Electronics and Communication Engineering

by

Aditya Acharya

(Roll Number: 510EC308)

based on research carried out

under the supervision of

Prof. Sukadev Meher

July, 2018

Department of Electronics and Communication Engineering

National Institute of Technology Rourkela

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela

July 21, 2018

Certificate of Examination

Roll Number: *510EC308*

Name: *Aditya Acharya*

Title of Dissertation: *Development of Some Spatial-domain Pre-processing and Post-processing Algorithms for Better 2-D Up-scaling*

We the below signed, after checking the dissertation mentioned above and the official record book (s) of the student, hereby state our approval of the dissertation submitted in partial fulfillment of the requirements of the degree of *Doctor of Philosophy* in *Electronics and Communication Engineering* at *National Institute of Technology Rourkela*. We are satisfied with the volume, quality, correctness, and originality of the work.

Sukadev Meher
Principal Supervisor

Samit Ari
Member, DSC

Ajit Kumar Sahoo
Member, DSC

Pankaj Sa
Member, DSC

Niladri Bihari Puhan
External Examiner

Dipti Patra
Chairperson, DSC

Tarun Kumar Dan
Head of the Department

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Prof. Sukadev Meher
Professor

July 21, 2018

Supervisor's Certificate

This is to certify that the work presented in the dissertation titled, *Development of Some Spatial-domain Pre-processing and Post-processing Algorithms for Better 2-D Up-scaling*, submitted by *Aditya Acharya*, Roll Number 510EC308, is a *bona fide* record of original research carried out by him under my supervision and guidance in partial fulfillment of the requirements of the degree of *Doctor of Philosophy* in *Electronics and Communication Engineering*. To the best of my knowledge, neither this dissertation nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

Sukadev Meher
Professor

Dedication

Dedicated
to
The Mother and Sri Aurobindo

Declaration of Originality

I, *Aditya Acharya*, Roll Number *510EC308* hereby declare that this dissertation entitled *Development of Some Spatial-domain Pre-processing and Post-processing Algorithms for Better 2-D Up-scaling* presents my original work carried out as a doctoral student of NIT Rourkela and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the sections "Reference" or "Bibliography". I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

July 21, 2018

NIT Rourkela

Aditya Acharya

Acknowledgment

This dissertation would not have been possible without the guidance and the help of several individuals who in one way or other contributed and extended their valuable assistance in course of this study.

I wish to express my sincere gratitude to my supervisor, Prof. Sukadev Meher, for his guidance, encouragement and support throughout this research work. His impressive knowledge, technical skills and human qualities have been a source of inspiration and a model for me to follow.

I am grateful to Prof. A. Biswas, Director NIT Rourkela for providing me adequate infrastructure to carry out my research. I am also thankful to Prof. Kamalakanta Mahapatra, Dean Academics NIT Rourkela for his constant support for the doctoral research.

I am very much thankful to Prof. Tarun Kumar Dan, Head of the Department, Electronics Communication Engineering, for his constant support. I also gratefully thank my Doctoral Scrutiny Members, Prof. Samit Ari, Prof. Dipti Patra, Prof. Pankaj Kumar Sa, Prof. Ajit Kumar Sahoo for their valuable suggestions on this dissertation.

I would also like to thank fellow research colleagues Deepak, Bibhu, Jyoti, Sonia, Dhiren for their accompaniment. It gives me a sense of happiness to be with them. I would also like to thank Silicon Institute of Technology Bhubaneswar for all the support and help for this research work. Finally, I would like to thank my parents, Sri Bijoy Kumar Acharya and Smt. Jyotshna Acharya, my wife Subhashree and daughter Adyashree, whose faith and patience had always been a great source of inspiration to me.

July 21, 2018 *Aditya Acharya*
NIT Rourkela Roll Number: 510EC308

Abstract

Image super-resolution is an area of great interest in recent years and is extensively used in applications like video streaming, multimedia, internet technologies, consumer electronics, display and printing industries. Image super-resolution is a process of increasing the resolution of a given image without losing its integrity. Its most common application is to provide better visual effect after resizing a digital image for display or printing. One of the methods of improving the image resolution is through the employment of a 2-D interpolation.

An up-scaled image should retain all the image details with very less degree of blurring meant for better visual quality. In literature, many efficient 2-D interpolation schemes are found that well preserve the image details in the up-scaled images; particularly at the regions with edges and fine details. Nevertheless, these existing interpolation schemes too give blurring effect in the up-scaled images due to the high frequency (HF) degradation during the up-sampling process. Hence, there is a scope to further improve their performance through the incorporation of various spatial domain pre-processing, post-processing and composite algorithms.

Therefore, it is felt that there is sufficient scope to develop various efficient but simple pre-processing, post-processing and composite schemes to effectively restore the HF contents in the up-scaled images for various online and off-line applications. An efficient and widely used Lanczos-3 interpolation is taken for further performance improvement through the incorporation of various proposed algorithms.

The various pre-processing algorithms developed in this thesis are summarized here. The term *pre-processing* refers to processing the low-resolution input image prior to image up-scaling. The various pre-processing algorithms proposed in this thesis are: Laplacian of Laplacian based global pre-processing (LLGP) scheme; Hybrid global pre-processing (HGP); Iterative Laplacian of Laplacian based global pre-processing (ILLGP); Unsharp masking based pre-processing (UMP); Iterative unsharp masking (IUM); Error based up-sampling (EU) scheme.

The proposed algorithms: LLGP, HGP and ILLGP are three spatial domain pre-processing algorithms which are based on 4^{th}, 6^{th} and 8^{th} order derivatives to alleviate non-uniform blurring in up-scaled images. These algorithms are used to obtain the high frequency (HF) extracts from an image by employing higher order derivatives and perform precise

sharpening on a low resolution image to alleviate the blurring in its 2-D up-sampled counterpart.

In case of unsharp masking based pre-processing (UMP) scheme, the blurred version of a low resolution image is used for HF extraction from the original version through image subtraction. The weighted version of the HF extracts are superimposed with the original image to produce a sharpened image prior to image up-scaling to counter blurring effectively.

IUM makes use of many iterations to generate an unsharp mask which contains very high frequency (VHF) components. The VHF extract is the result of signal decomposition in terms of sub-bands using the concept of analysis filter bank. Since the degradation of VHF components is maximum, restoration of such components would produce much better restoration performance.

EU is another pre-processing scheme in which the HF degradation due to image up-scaling is extracted and is called prediction error. The prediction error contains the lost high frequency components. When this error is superimposed on the low resolution image prior to image up-sampling, blurring is considerably reduced in the up-scaled images.

Various post-processing algorithms developed in this thesis are summarized in following. The term *post-processing* refers to processing the high resolution up-scaled image. The various post-processing algorithms proposed in this thesis are: Local adaptive Laplacian (LAL); Fuzzy weighted Laplacian (FWL); Legendre functional link artificial neural network (LFLANN).

LAL is a non-fuzzy, local based scheme. The local regions of an up-scaled image with high variance are sharpened more than the region with moderate or low variance by employing a local adaptive Laplacian kernel. The weights of the LAL kernel are varied as per the normalized local variance so as to provide more degree of HF enhancement to high variance regions than the low variance counterpart to effectively counter the non-uniform blurring. Furthermore, FWL post-processing scheme with a higher degree of non-linearity is proposed to further improve the performance of LAL. FWL, being a fuzzy based mapping scheme, is highly nonlinear to resolve the blurring problem more effectively than LAL which employs a linear mapping.

Another LFLANN based post-processing scheme is proposed here to minimize the cost function so as to reduce the blurring in a 2-D up-scaled image. Legendre polynomials are used for functional expansion of the input pattern-vector and provide high degree of nonlinearity. Therefore, the requirement of multiple layers can be replaced by single layer LFLANN architecture so as to reduce the cost function effectively for better restoration performance. With single layer architecture, it has reduced the computational complexity and hence is suitable for various real-time applications.

There is a scope of further improvement of the stand-alone pre-processing and post-processing schemes by combining them through composite schemes. Here, two spatial domain composite schemes, CS-I and CS-II are proposed to tackle non-uniform blurring in an up-scaled image. CS-I is developed by combining global iterative Laplacian (GIL) pre-processing scheme with LAL post-processing scheme. Another highly nonlinear composite scheme, CS-II is proposed which combines ILLGP scheme with a fuzzy weighted Laplacian post-processing scheme for more improved performance than the stand-alone schemes.

Finally, it is observed that the proposed algorithms: ILLGP, IUM, FWL, LFLANN and CS-II are better algorithms in their respective categories for effectively reducing blurring in the up-scaled images.

Keywords: image up-sampling, image super-resolution, image deblurring, 2-D interpolation, Nearest-neighbour interpolation, Bilinear interpolation, Bicubic interpolation, Lanczos interpolation, DCT interpolation, Laplacian of Laplacian, fuzzy logic, FLANN, fuzzy Laplacian, iterative unsharp masking, low resolution, high resolution

Contents

Certificate of Examination **iii**

Supervisors' Certificate **iv**

Dedication **v**

Declaration of Originality **vi**

Acknowledgment **vii**

Abstract **viii**

List of Acronyms **xiv**

List of Symbols **xvi**

1. Introduction **1**

1.1 Introduction to image up-scaling. 2

1.2 Fundamental of image interpolation 3

1.3 Some basic interpolation schemes 6

1.3.1 Nearest–neighbor interpolation……………………………………… 6

1.3.2 Bilinear interpolation……………………………………………… 7

1.3.3 Bicubic interpolation……………………………………………… 10

1.4 Advantages and disadvantages of various existing interpolation schemes. . 12

1.5 Problem statement……………………………………………………… 15

1.6 Methodology adopted…………………………………………………….. 16

1.7 Performance measure…………………………………………………….. 17

1.8 Chapter wise organization of the thesis…………………………………… 19

1.9 Conclusion………………………………………………………………… 21

2. Literature Review **22**

2.1 Image up-scaling using 2-D interpolation……………………………… 23

2.2 Review of polynomial based interpolation………………………………… 23

2.3 Review of edge-directed interpolation…………………………………… 28

2.4 Review of transform-domain interpolation……………………………… 30

2.5 Study of some existing interpolation schemes…………………………… 36

2.5.1 Lanczos-3 interpolation…………………………………………… 36

2.5.2 DCT interpolation ……………………………………………… 37

2.5.3 New edge directed interpolation…………………………………. 38

2.5.4 Image super-resolution based on ………………………………… 39

2.5.5 Image resolution enhancement using …………………………….… 40

2.6 Comparative analysis…………………………………………………… 42

2.7 Conclusion………………………………………………………………… 51

3. Spatial-domain pre-processing algorithms using higher order derivatives **52**
3.1 Laplacian of Laplacian based global pre-processing. 53
3.1.1 Development of a 5×5 LOL filter kernel …………………………. 54
3.1.2 LLGP algorithm…………………………………………………… 56
3.2 Hybrid global pre-processing scheme 57
3.2.1 HGP algorithm 58
3.2.2 Development of high-pass filter kernel based on 6th order ……… 59
3.3 Iterative Laplacian of Laplacian based global pre-processing 62
3.3.1 ILLGP algorithm 63
3.4 Experiment and simulation 63
3.4.1 Estimation of weight factor………………………………….. 64
3.5 Results and discussion………………………………………………… 80
3.6 Conclusion ………………………………………………………….. 83

4. Pre-processing algorithms using unsharp masking **85**
4.1 Unsharp masking based pre-processing scheme……………………… 87
4.1.1 UMP algorithm………………………………………………. 89
4.2 Iterative unsharp masking scheme ……………………………………. 90
4.2.1 Estimation of number of iterations and weight factor………….. 95
4.2.2 IUM algorithm………………………………………………… 100
4.3 Error based up-sampling scheme…………………………………….. 101
4.3.1 EU algorithm…………………………………………………… 103
4.4 Experiment and simulation……………………………………… 103
4.5 Results and discussion……………………………………………. 116
4.6 Conclusion……………………………………………………… 119

5. Post-processing algorithms using soft-computing techniques **121**
5.1 Local adaptive Laplacian based post-processing………………………. 123
5.1.1 LAL algorithm ……………………………………………….. 124
5.2 Fuzzy weighted Laplacian based post-processing……………………….. 126
5.2.1 Fuzzy based mapping …………………………………………… 126
5.2.2 FWL algorithm ……………………………………………….. 133
5.2.3 Weight factor estimation …………………………………….. 133
5.3 LFLANN based post-processing ……………………………………. 133
5.3.1 LFLANN algorithm …………………………………………….. 138
5.4 Results and discussion ………………………………………………… 152
5.5 Conclusion…………………………………………………………. 155

6. **Development of some spatial domain composite algorithms** **157**
6.1 Composite scheme using iterative Laplacian and local adaptive ………. 159
6.1.1 HF enhancement using iterative Laplacian………………………. 160
6.1.2 CS-I algorithm ………………………………………………….. 162
6.1.3 Estimation of weight factors …………………………………… 163
6.2 Composite scheme CS-II using iterative Laplacian of Laplacian…….. 166
6.2.1 Fuzzy mapping technique employed in CFWL …………………. 168
6.2.2 CS-II algorithm…………………………………………………. 173
6.2.3 Estimation of weight factors……………………………………. 173
6.3 Results and discussion …………………………………………………. 188
6.4 Conclusion ……………………………………………………………… 190

7. Conclusion **192**
7.1 Comparative analysis ………………………………………………….. 193
7.2 Computational complexities of the proposed algorithms ………………. 198
7.3 Conclusion ……………………………………………………………….. 203
7.4 Scope for future work……………………………………………………. 204

References **205**

Dissemination **214**

List of Acronyms

HDTV	High definition television
2-D	Two dimension
HR	High resolution
LR	Low resolution
SVC	Scalable video coding
IR	Infrared
1-D	One dimension
NEDI	New edge directed interpolation
SAI	Soft-decision adaptive interpolation
BSAI	Bilateral soft-decision adaptive interpolation
DCT	Discrete cosine transform
IDCT	Inverse discrete cosine transform
LLGP	Laplacian of Laplacian based global pre-processing
HGP	Hybrid global pre-processing
ILLGP	Iterative Laplacian of Laplacian based global pre-processing
UMP	Unsharp masking based pre-processing
IUM	Iterative unsharp masking
EU	Error based up-sampling
LAL	Local adaptive Laplacian
FWL	Fuzzy weighted Laplacian
LFLANN	Legendre functional link artificial neural network
CS	Composite scheme
MSE	Mean square error
RMSE	Root mean square error
MAE	Mean absolute error
PSNR	Peak signal to noise ratio
UQI	Universal quality index
HVS	Human visual system
FIR	Finite impulse response
EDI	Edge directed interpolation
HMT	Hidden Markov tree
SWT	Stationary wavelet transform
DWT	Discrete wavelet transform

CR	Compression ratio
LOL	Laplacian of Laplacian
LPF	Low pass filter
CIF	Common intermediate format
HF	High frequency
VHF	Very high frequency
ANN	Artificial neural network
\ANFIS	Adaptive neural based fuzzy inference system
SISO	Single input single output
FIS	Fuzzy inference system
MLP	Multi layer perceptron
LMS	Least mean square
GIL	Global iterative Laplacian
CFWL	Composite fuzzy weighted Laplacian

List of Symbols

$f(x,y)$	Original image; image pixel at co-ordinate (x,y)
$g(x,y)$	Sub-sampled, low resolution image
$\hat{f}(x,y)$	Reconstructed image using interpolation
x_k	Interpolation node in 1-D
δ	Sampling increment
δ_x	Sampling increment in x direction
$h(x)$, $h(y)$	1-D interpolation kernel
$h(x,y)$	2-D interpolation or filter kernel
$e(x,y)$	Error image
$L(x)$	1-D Lanczos interpolation kernel
∇^2	2nd order derivative operator
∇^4	4th order derivative operator
$h_{LOL}(x,y)$	Laplacian of Laplacian filter kernel
K	Weight factor
$g_s(x,y)$	Sharpened, sub-sampled image
$f(x,y,n)$	Video sequence; n representing frame number
$Lanczos3(x)$	Lanczos-3 interpolation
$h_{La}(x,y)$	Laplacian kernel
$h_{Lav}(x,y)$	Averaging low-pass filter kernel
$g_{UM}(x,y)$	Unsharp mask
$g_{LL}(x,y)$	LL sub-band of image $g(x,y)$
$g_{LH}(x,y)$	LH sub-band of image $g(x,y)$
$g_{HH}(x,y)$	HH sub-band of image
$h_{Lwa}(x,y)$	Weighted average filter kernel
$*$	2-D convolution
ΔK	Weight factor deviation
w	3×3 window
P	Number of rows of the original image
Q	Number of columns of the original image
$\sigma^2(x,y)$	Local variance in a 3×3 neighborhood
m	Local mean
$\sigma^2_{\mathbf{max}}$	Maximum local variance of an image
σ^2_N	Normalized local variance
$h_{LAL}(x,y)$	Local adaptive Laplacian filter kernel

$\psi(x,y)$	HF extracts due to high pass filtering
μ_i	Input fuzzy membership function
μ_o	Output fuzzy membership function
min	Minimum or AND operator
$R(w_t)$	Response of fuzzy rule
w_{tO}	De-fuzzified output
$h_{FWL}(x,y)$	Fuzzy weight Laplacian filter kernel
N	Degree of functional expansion
μ	Learning rate parameter
max	Maximun or union operator

Chapter 1

Introduction

Preview

Image up-scaling is a promising area of research in the field of digital image processing. The image up-scaling or resolution enhancement is a process of improving the resolution of a given image without losing its integrity. Image up-scaling is widely used in numerous applications such as video streaming, high resolution display (HDTV), digital camera, multimedia, consumer electronics, video surveillance, satellite imaging, image correction, quality control, medical image processing and printing industries. In many such applications, it is required to increase the size of a low resolution image without losing much image details meant for better visual quality. Various 2-D, spatial-domain and transform-domain interpolation schemes are employed for this purpose. However, most of the interpolation schemes produce blurring in the up-scaled images due to high frequency degradation during the up-sampling process. Hence, it is very much essential to up-scale an image and to preserve the edges and fine details. Therefore, it is felt that there is sufficient scope to further improve the performance of the existing 2-D interpolation schemes. In the present research work, efforts are made to develop various efficient spatial-domain pre-processing and post-processing schemes to alleviate blurring in up-scaled images.

The following topics are covered in this introductory chapter.

- *Introduction to Image Up-scaling*
- *Fundamentals of Image Interpolation*
- *Some Basic Interpolation Schemes*
- *Brief Literature Review on Image Interpolation Schemes*
- *Problem Statement*
- *Methodologies Adopted*
- *Performance Measures*
- *Chapter-wise Organization of the Thesis*
- *Conclusion*

1.1 Introduction to Image Up-scaling

Image up-scaling is a topic of great interest in recent years and is used for generating a high resolution (HR) image from a low resolution (LR) image data. An efficient image up-scaling scheme must preserve the high frequency information, texture, geometrical regularities and smoothness of the original LR image while producing its corresponding HR counterpart. The most common application of image up-scaling is to provide an enhanced visual effect after resizing a digital image for display and printing [1]. There are many applications of image up-scaling techniques; some of them are described below.

There is a great need to facilitate flexible image/video format conversion among various multimedia terminals such as digital cameras, cellular phones, computers and HDTV. Generally, the low resolution image captured by mobile phones and digital camera can't be displayed in a high definition television (HDTV). Hence, the resolution of input signal coming from low resolution source is converted to high resolution through a 2-D interpolation technique prior to display in a high definition screen. So, image super-resolution or up-scaling is used to enhance the ability of the existing viewing devices for displaying signal from a low resolution source [2].

Moreover, image super-resolution is used in streaming video websites, which often store low resolution videos for bandwidth constraint. If the user wishes to enhance the resolution to watch full screen, then the process is accomplished through 2-D interpolation [2]. Now-a-days, the up-sampling technologies are also employed in scalable video coding (SVC) to provide spatial scalability which can provide several video resolutions as desired by various consumer applications.

Image up-scaling or super-resolution has a wide range of applications in numerous fields such as medical image processing, military applications, satellite image enhancement, video surveillance etc. Zooming or rotating medical images after their acquisition is often desired for proper diagnosis and treatment. Therefore, interpolation methods are incorporated into the systems for computer aided diagnosis [3].

Currently, satellite images are used in many applications such as geosciences studies, remote sensing, weather forecasting, geographical information system and military applications. In such applications, very often it is desired to improve the native resolution offered by imaging hardware using interpolation for proper visibility and quality enhancement for subsequent analysis and interpretation.

Now-a-days digital cameras are abundantly available that produce high resolution images. However, there are many existing LR cameras as well as low-grade sensors found in existing mobile devices and surveillance systems for cost effectiveness. Hence, to improve the quality and resolution of such images at a lower cost, 2-D interpolation is employed. This ensures proper video surveillance at a low cost. Likewise, infra-red (IR) cameras generally have low resolution because of high cost of the IR sensors. Hence, the quality and resolution of IR images can be enhanced using interpolation in a cost-effective way. Up-scaling techniques are also used in digital camera to vary the resolution of the captured image according to the zooming criteria as desired by the user for proper visibility.

Various spatial domain interpolation schemes are used in many image up-scaling applications. For most of the real-time applications, various conventional, spatial-domain interpolation schemes such as nearest-neighbor, bilinear, bicubic and cubic-spline interpolation are used for their simplicity and much reduced computational complexities. Fundamentals of interpolation along with some basic interpolation schemes are discussed in the next section.

1.2 Fundamentals of Image Interpolation

Image interpolation is the process of estimating the intermediate values of a spatially continuous image from a set of its discrete samples. It typically estimates an unknown pixel value within a neighborhood from the known pixel neighbors. To determine the intensity at a particular position in between the known pixels, the intensities of the neighboring pixels and the distance between the estimating point and the neighboring pixels are incorporated into the estimation process. Interpolation is widely used in various fundamental digital image processing operations such as re-sampling, translation, scaling, rotation and geometric correction. These general operations require image values at locations for which no samples are available. Typically, the interpolated values at these locations are computed as a weighted average or convolution of the neighboring image samples. The weighting function used in local convolution is called the kernel. So, interpolation is the process of determining the values of a function at positions lying between its samples. It achieves this process by fitting a continuous function through the discrete input samples. The interpolation function is a special type of approximating function. A fundamental property of interpolation function is that they must coincide with the sampled data at the interpolation nodes or sample point. In between the sample points; it estimates the function value using the convolution of the neighboring samples.

If $g(x_k)$ is a sampled function and $\hat{f}(x)$ is the corresponding interpolated function, then $\hat{f}(x_k) = g(x_k)$ whenever x_k is an interpolation node. For equally spaced 1-D sample data $g(x_k)$, many 1-D interpolation techniques can be used. The interpolated function $\hat{f}(x)$ can be determined in 1-D form by

$$\hat{f}(x) = \sum_k c_k \, h\left(\frac{x - x_k}{\delta}\right) \tag{1.1}$$

where δ represents the sampling increment and x_k's are the interpolation nodes. h is the interpolation kernel, and $\hat{f}$ is the interpolated function. The c_k's are parameters which depend on the sampled data and are also called interpolation coefficients. They are selected so that the interpolation condition, $\hat{f}(x_k) = g(x_k)$ for each x_k is satisfied [4]. The interpolation kernel, h converts discrete data into a continuous function by an operation similar to convolution. Generally, h is a symmetric kernel, i.e., $h(-x) = h(x)$. The interpolation kernel, weighted by coefficients c_k is applied to k data samples, x_k to obtain the functional value at an unknown location, x.

An image interpolation attempts to reconstruct a two-dimensional continuous signal $\hat{f}(x, y)$ from its discrete image samples $g(x_j, y_k)$ and hence is an extension of 1-D interpolation in both x and y directions. This process can be described as the 2-D convolution of the discrete image samples $g(x_j, y_k)$ with a 2-D reconstruction filter or interpolation kernel $h(x, y)$ meant for estimating the intensity value at any position (x, y) from its discrete neighboring pixels. Let us consider a point (x, y) in a rectangular sub-division $[x_j, x_{j+1}] \times [y_k, y_{k+1}]$. The pixel intensity at that point using 2-D interpolation [4] is given by,

$$\hat{f}(x, y) = \sum_{l=-1}^{2} \sum_{m=-1}^{2} c_{j+l, k+m} \, h\left(\frac{x - x_{j+l}}{\delta_x}\right) h\left(\frac{y - y_{k+m}}{\delta_y}\right) \tag{1.2}$$

where, h is the interpolation kernel, δ_x and δ_y are sampling increments of the x and y co-ordinates. For interior grid points, the c_{jk}'s are given by $c_{jk} = f(x_j, y_k)$. The range of l, m defines the size of the neighborhood for pixel estimation. Their values range from -1 to 2 for a 4×4 neighborhood.

For interpolating a point in a 2-D signal such as image, 1-D interpolation can also be employed separately in both x and y directions at the given location. So, the 2-D interpolation kernel, $h(x, y)$ can be represented as the product of two separable 1-D interpolation kernel in x and y direction. Mathematically, the representation of a 2-D interpolation kernel $h(x, y)$ in terms of 1-D is given by,

$$h(x, y) = h(x).\, h(y)$$

where, $h(x)$ and $h(y)$ represent 1-D interpolation kernel in x and y direction respectively.

For image re-sampling, the interpolation step must reconstruct a two dimensional continuous signal $\hat{f}(x, y)$ from its discrete samples $g(x_j, y_k)$. Thus the amplitude at (x, y) position is estimated from its discrete neighbors. This process is analogous to the convolution of discrete samples with the continuous 2-D impulse response $h(x, y)$ of a 2-D reconstruction filter. Therefore (1.2) can also be written in the following form [3].

$$\hat{f}(x, y) = \sum_j \sum_k g(x_j, y_k)\, h(x - x_j, y - y_k) \tag{1.3}$$

Sometimes, it is necessary to down-sample an image for display or for transmission through a channel with bandwidth constraints. This sampling process generates an infinite bandwidth signal out of a band limited signal. An interpolation scheme performs inverse operation on the discrete signal by reducing its bandwidth by filtering through a low-pass filter. So, interpolation reconstructs the signal lost in the sampling process by smoothing the data samples with an interpolation function or kernel. Hence, there occurs high frequency degradation during the interpolation process resulting in blurring of the 2-D data samples.

There are some basic criteria for an efficient interpolation scheme. The interpolation method should preserve the geometry and relative sizes of objects in an image so that the subject matter doesn't change under interpolation. The method should show contrast invariance by preserving the luminance values of objects in an image and by maintaining the overall contrast of the image. It must not add noise and other artifacts to the image such as ringing artifacts near the boundaries. It should preserve edges and boundaries without significant blurring. Jagged artifacts or “staircase” edges generated because of aliasing ought to be prevented by the interpolation scheme. The textured region must be well preserved without blurring or smoothing during interpolation. The interpolated image should be free from blocking artifacts.

1.3 Some Basic Interpolation Schemes

There are several interpolation techniques. Some of the basic interpolation schemes such as nearest-neighbor, bilinear and bicubic are discussed in this section meant for 2-D interpolation. The performance and computational complexity of interpolation algorithms are related to their interpolation kernel. So, the development of efficient interpolation kernels is the objective of design and analysis. A tradeoff between accuracy and efficiency is one of the influencing factors while designing an efficient interpolation kernel. In this section, we will discuss various 1-D and 2-D interpolation schemes.

1.3.1 Nearest-neighbor Interpolation

Nearest-neighbor interpolation is the simplest interpolation technique in which the interpolated pixel value is determined by the nearest neighbor in the proximity. It is called *nearest neighbor interpolation* because it assigns to each new location the intensity of its nearest neighbor in the original image. Hence, this interpolation is also called pixel replication. In case of 1-D interpolation, it takes two pixels into consideration while, in 2-D interpolation, it takes four pixels into account for estimating the pixel intensity at any point within the neighborhood of the original image. This interpolation is also called point-shift algorithm as the interpolated image is shifted with respect to original image by the difference between positions of co-ordinate location. The interpolated value at a given location using nearest-neighbor interpolation is given by the following expression.

$$\hat{f}(x) = g(x_k), \quad \frac{x_{k-1} + x_k}{2} < x \le \frac{x_k + x_{k+1}}{2} \tag{1.4}$$

where, $g(x_k)$ represents a discrete 1-D function and $\hat{f}(x)$ represents the interpolated function. The interpolated value in one-dimension at a given point using nearest-neighbor interpolation can be estimated by convolving the input data sample with the following interpolation kernel in spatial domain. The interpolation kernel for nearest-neighbor algorithm is given by,

$$h(x) = \begin{cases} 1, & 0 \le |x| < 0.5 \\ 0, & 0.5 \le |x| \end{cases} \tag{1.5}$$

The interpolation in 2-D can be performed by convolving the input image in both x and y directions using the above kernel [3].

This method requires minimum computational time because it uses the square pulse as kernel function and hence is the fastest among the basic interpolation schemes. This approach is simple but has the tendency to produce undesirable artifacts, such as distortion at the straight edges. An image under high magnification looks blocky using nearest-neighbor interpolation. Hence, these artifacts are called blocking artifacts. For this reason it is infrequently used in practice.

The convolution in spatial domain with the rectangular function, $h(x)$ is equivalent to frequency domain multiplication with a sinc function. The spectrum of a rectangular function is a sinc function which represents a poor frequency response for-low pass filter due to prominent side-lobes and infinite extent [3]. The problems encountered in nearest-neighbor interpolation are to some extent resolved using bilinear interpolation at the cost of a little computational complexity which is discussed in the next section.

1.3.2 Bilinear Interpolation

For interpolating a 1-D function, linear interpolation technique is used. However, to interpolate a 2-D function bilinear interpolation is employed. The bilinear interpolation is a 2-D interpolation scheme which makes use of the linear interpolation in horizontal and vertical directions for determining pixel intensity at a desired location within a 2×2 neighbourhood. It uses the weighted average of the four neighboring pixels to arrive at the final interpolated value at the desired location.

In case of bilinear interpolation, the pixel intensity at a given location is estimated as a linear combination of the four neighboring pixel with weights inversely proportional to their distance from the estimating location. The weighted combination of neighboring pixel simply represents a low- pass filtering operation which blurs the fine details and edges of an image due to high frequency attenuation.

Bilinear interpolation is more complex and computationally demanding than the nearest neighbor interpolation. It eliminates the blocking artifacts produced by the nearest neighbor interpolation up to a certain level at the cost of blurring the fine details and edges of the original image. The impulse response in case of linear interpolation is a triangle function. Linear interpolation is a first degree method or a first order polynomial because it passes a straight line through every two consecutive points of the input signal [3]. The expression of the impulse response of a linear interpolation is given by,

$$h(x)=\begin{cases}1-|x|, & 0\le |x|<1\\ 0, & 1\le |x|\end{cases} \tag{1.6}$$

The frequency response of the linear interpolation function is the square of a sinc function whose side lobes are less prominent than the side lobes of the sinc function and hence is having an improved frequency response over nearest-neighbor interpolation. It attenuates the harmonics of the input signal near its cut-off frequency and moderately attenuates the pass-band resulting in smoothing of an image. The side lobes are far less prominent, indicating improved performance in the stop band [3]. However, a significant amount of spurious high frequency components continue to leak the pass-band, contributing to some aliasing.

Point Estimation in 2-D using Bilinear Interpolation:

Suppose the value of an unknown function f at a point (x,y) is to be determined. It is assumed that the value of f at the four points Q_{11}, Q_{12}, Q_{21} and Q_{22} is known. The locations of the four points are given by, $Q_{11}=(x_1,y_1)$, $Q_{12}=(x_1,y_2)$, $Q_{21}=(x_2,y_1)$ and $Q_{22}=(x_2,y_2)$ as shown in Fig. 1.1. To estimate the pixel intensity an unknown location (x,y), linear interpolation is carried out initially in x - direction. This yields,

$$f(R_1)\cong\frac{x_2-x}{x_2-x_1}f(Q_{11})+\frac{x-x_1}{x_2-x_1}f(Q_{21}) \tag{1.7}$$

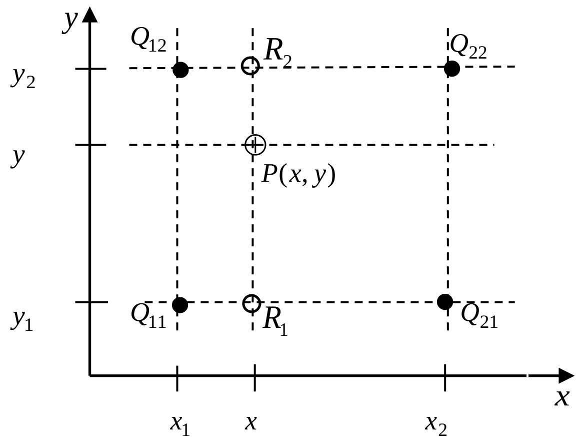


Fig. 1.1 Estimation of pixel intensity at an unknown location (x, y) using bilinear interpolation

where, R_1 is located at (x, y_1). Similarly, the function value at R_2 is given by,

$$f(R_2) \cong \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \tag{1.8}$$

The point R_2 is located at (x, y_2). After performing linear interpolation in x-direction, linear interpolation is done in y-direction to determine the functional value, $\hat{f}(P)$ at the desired location (x, y) and is given by,

$$\hat{f}(P) \cong \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2) \tag{1.9}$$

Substituting (1.7) and (1.8) into (1.9) we have,

$$\begin{aligned}
\hat{f}(P) &\cong \frac{x_2 - x}{x_2 - x_1}\left[\frac{(y_2 - y) f(Q_{11}) + (y - y_1) f(Q_{12})}{(y_2 - y_1)}\right] + \frac{x - x_1}{x_2 - x_1}\left[\frac{(y_2 - y) f(Q_{21}) + (y - y_1) f(Q_{22})}{(y_2 - y_1)}\right] \\
&\cong \frac{f(Q_{11})\,(x_2 - x)(y_2 - y)}{(x_2 - x_1)\,(y_2 - y_1)} + \frac{f(Q_{12})\,(x_2 - x)\,(y - y_1)}{(x_2 - x_1)\,(y_2 - y_1)} \\
&\quad + \frac{f(Q_{21})\,(x - x_1)\,(y_2 - y)}{(x_2 - x_1)\,(y_2 - y_1)} + \frac{f(Q_{22})\,(x - x_1)\,(y - y_1)}{(x_2 - x_1)\,(y_2 - y_1)}
\end{aligned} \tag{1.10}$$

where $f(Q_{11})$, $f(Q_{12})$, $f(Q_{21})$ and $f(Q_{22})$ represent pixel intensities at four different locations as shown in Fig. 1.1. The result of bilinear interpolation is independent of the order of interpolation [5].

Bilinear interpolation performs much better than nearest-neighbor interpolation by reducing blocking artifacts. However, it shows much blurring at the edges and fine details and can be alleviated to a considerable extent using bicubic interpolation which is discussed in the next section.

1.3.3 Bicubic Interpolation

For interpolating a 1-D function, cubic interpolation is used. Likewise, for interpolating a 2-D function, cubic interpolation is employed in both x and $y-$directions which is otherwise called as bicubic interpolation. The interpolated surface is smoother than nearest-neighbor interpolation but preserves fine detail and edge information more than bilinear interpolation. In image processing, bicubic interpolation is preferred over bilinear interpolation or nearest neighbor interpolation when speed is not an issue. In addition, this interpolation has fewer interpolation artifacts than nearest-neighbor and bilinear interpolation.

A bicubic interpolation involves the sixteen nearest neighbors of a point while determining the intensity of a particular point (x, y) and hence is computationally more complex than bilinear interpolation. The pixel intensity at an unknown location (x, y) is determined by the linear combination of its sixteen neighboring pixels using the following expression [6].

$$\hat{f}(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j \tag{1.11}$$

where the sixteen coefficients are determined from the sixteen equations in sixteen unknowns that can be written using sixteen nearest-neighbors of the point of estimation.

Generally, bicubic interpolation does a better job [6] of preserving fine detail than its bilinear counterpart. Bicubic interpolation is the standard used in commercial image editing program such as Adobe Photoshop and Corel Photo-point.

Bicubic interpolation employs cubic convolution in x and y directions which is a third-degree interpolation algorithm and produces good resizing results. The impulse response of cubic interpolation is composed of piecewise cubic polynomial defined on the subintervals (-2,-1), (-

1,0), (0,1) and (1,2). Outside the interval (-2, 2), the impulse response is zero. As a result, each interpolated point is a weighted sum of four consecutive input points. This has the desirable symmetry property of retaining two input points on each side of the interpolation region. It gives rise to a symmetric, space-invariant property of the interpolation kernel.

It is required to solve eight linear equations with seven unknowns to derive the cubic interpolation impulse response. The performance of the interpolation impulse response depends on parameter, a and the frequency content of an image which is to be interpolated. For different images, different values of a yield the best performance. The impulse response of the cubic convolution is of the form:

$$h(x)=\begin{cases}(a+2)|x|^3-(a+3)|x|^2+1, & 0\le|x|<1\\ a|x|^3-5a|x|^2+8a|x|-4a, & 1\le|x<2|\\ 0, & 2\le|x|\end{cases} \tag{1.12}$$

The value of the parameter a determines the kernel performance. Choice for value of a between -3 and 0 approximates the sinc function kernel. If the value of $a=-1$, the kernel will amplify the high frequency components in the pass band resulting in image sharpening. If the interpolation function is to agree with the first three terms of the Taylor series expansion for the given function, then the parameter a must be equal to -0.5. Hence, putting $a=-0.5$ in (1.12) the expression becomes:

$$h(x)=\begin{cases}(3/2)|x|^3-(5/2)|x|^2+1, & 0\le|x|<1\\ (-1/2)|x|^3+(5/2)|x|^2-4|x|+2, & 1\le|x|<2\\ 0, & 2\le|x|\end{cases} \tag{1.13}$$

Two-dimensional cubic interpolation, otherwise known as bicubic interpolation is accomplished by performing one-dimensional cubic interpolation with respect to each co-ordinate as shown in Fig. 1.2. Therefore, the 2-D cubic convolution interpolation function is a separable extension of the 1-D interpolation. Let us consider a point (x,y) in a rectangular sub-division $[x_j,x_{j+1}]\times[y_k,y_{k+1}]$. The pixel intensity at that point using the bicubic interpolation [4] is given by,

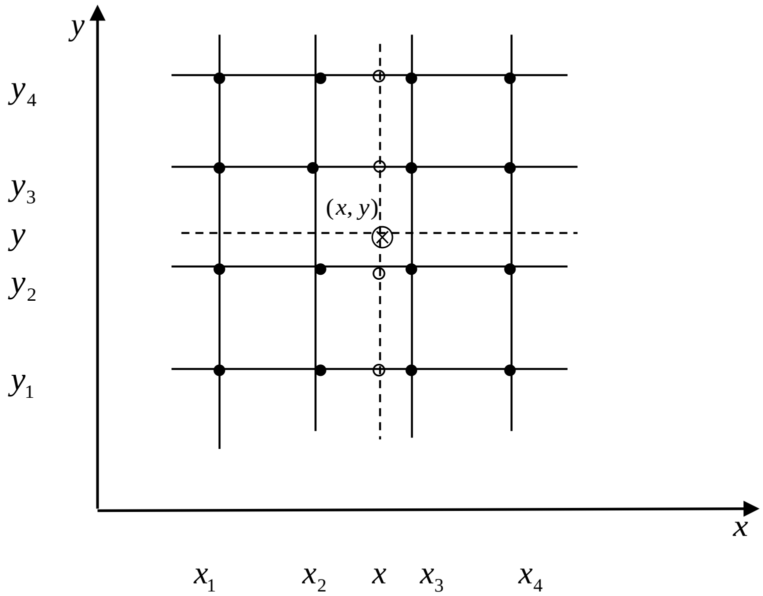


Fig. 1.2 Estimation of pixel intensity at an unknown location (x, y) employing bicubic interpolation

$$\hat{f}(x, y) = \sum_{l=-1}^{2} \sum_{m=-1}^{2} c_{j+l,k+m}\, h\left(\frac{x - x_{j+l}}{\delta_x}\right) h\left(\frac{y - y_{k+m}}{\delta_y}\right) \tag{1.14}$$

where, h is the interpolation kernel of (1.13) and δ_x and δ_y are the x and y co-ordinate sampling increments. For interior grid points, the c_{jk} 's are given by $c_{jk} = f(x_j, y_k)$.

The frequency response in case of cubic interpolation is better than the bilinear interpolation where high frequency components are almost attenuated at the stop band resulting in reduced aliasing and blurring as compared to bilinear interpolation [3].

1.4 Advantages and Disadvantages of Various Existing Interpolation Schemes

Image interpolation is the process of estimating the intermediate values of a spatially continuous image from a set of its discrete samples. It typically estimates an unknown pixel value within a neighborhood from the known pixel neighbors. Typically, the interpolated value at a particular location is computed by the weighted average or convolution of the neighboring image samples. The weighting function used in local convolution is called the interpolation kernel. Many image interpolation schemes, developed so far, are broadly classified into the following three categories. They are namely,

- Polynomial based interpolation schemes [14-36]
- Edge directed interpolation schemes [37-65]
- Transform-domain interpolation schemes [66-82]

The **polynomial based interpolation** schemes are based on the convolution of sampled data with an interpolation kernel. In these cases, the pixel intensity at an unknown location is directly estimated based on convolution without considering the local statistical features of an image and hence are not adaptive techniques. Most of such techniques are simple, easy to implement and computationally less complex and so are suitable for various real-time applications. However, they suffer from aliasing, blurring at the edges in the reconstructed image. In general, the perfect reconstruction of a band limited 2-D signal such as image from a set of its sample requires 2-D convolution in spatial domain. In case of polynomial based interpolation, the interpolated value at an estimating point is computed as a weighted average or convolution of the neighboring image samples. Hence, interpolation is analogous to a low pass filtering operation resulting in blurring at the edges and fine details. On the other hand, the key concern with spatial aliasing in images are the introduction of artifacts such as jaggedness in line features, spurious highlights and the appearance of frequency patterns not present in the original image. Spatial aliasing is due to under-sampling. For instance, a continuous 2-D function of two continuous variables can be band limited only if it extends infinitely in both co-ordinate directions. The very act of spatially limiting the 2-D function introduces frequency components extending to infinity in frequency domain. Since we cannot sample a function infinitely, aliasing is always present in digital images. The effect of aliasing can be reduced by band limiting or slightly blurring an image to be re-sampled so that high frequencies are attenuated. Various polynomial based interpolation schemes are nearest-neighbor, bilinear, bicubic, cubic-spline, lanczos-3 etc. The performance of the polynomial based interpolation schemes can be enhanced through incorporation of some adaptive and / or hybrid techniques. Those interpolation schemes are advance polynomial based interpolation schemes and perform better than conventional interpolation schemes [32-36]. Various advance polynomial based interpolation schemes are image interpolation using adaptive fast B-spline filtering, adaptive least-square bilinear interpolation, parametric cubic convolution scaler, sub-pixel edge localization and interpolation, adaptive interpolation based Gaussian function etc.

Various **edge directed interpolation** schemes eliminate the unwanted artifacts to a certain extent that are frequently encountered in polynomial based interpolation schemes at the cost of additional computational complexities. The conventional polynomial based techniques only preserve the low frequency details but fail to preserve the high frequency details resulting in undesirable blurring at the edges. This problem can be alleviated using various edge-directed algorithms which preserve high frequency information in an up-scaled image for a better visual quality. Although the edge directed interpolation schemes are efficient in preserving fine details and edge information in an image while up-scaling, they are computationally more complex than polynomial interpolation schemes because of the employment of adaptive and local based techniques. Hence, these schemes are not suitable for real-time applications. Various edge directed interpolation schemes [37-65] are new edge directed interpolation (NEDI), soft-decision adaptive interpolation (SAI), bilateral soft-decision adaptive interpolation (BSAI) etc.

Many **transform domain** techniques [66-82] for image resizing have been developed. Up-sampling in DCT domain is implemented by padding zero coefficients to the high frequency side and then by taking the inverse DCT (IDCT) to get back the up-scaled image in spatial domain. Image resizing in DCT domain shows very good result in terms of scalability and image quality. However, these techniques although improved, suffer through undesirable ringing and blocking artifacts and are computationally more complex than various spatial domain interpolation techniques. Ringing artifacts appear as spurious signals near sharp transitions in an image. The main cause of ringing is due to the band limiting of an image in frequency domain by padding zero coefficients to the high frequency side or truncating image coefficients in frequency domain. The DCT domain image resizing methods make use of a truncation that discards HF coefficients to down-scale and zero padding to up-scale an image. However, these truncation and zero padding in frequency domain generate ringing artifacts near object boundaries in spatial domain. Furthermore, DCT is computationally more complex than polynomial based interpolation schemes because of the requirement of forward transform and inverse transform during the up-sampling process as depicted in Table 2.1.

Likewise, various wavelet domain resolution enhancement schemes have been developed [79-82] so as to preserve the high frequency contents in up-scaled images. Although the wavelet domain interpolation schemes preserve the HF information quite effectively in the up-scaled images, they demand more computational time as compared to conventional interpolation schemes. Since wavelet is a transform domain interpolation scheme, it consumes more computation time like DCT interpolation. The forward discrete wavelet transform decomposes an image to LL, LH, HL and HH components. These components are individually up-scaled to higher dimensions using bicubic interpolation followed by inverse discrete wavelet transform to obtain the up-scaled image in spatial domain. Therefore, wavelet requires more computation time than polynomial interpolation schemes as depicted in Table 2.1.

Hence, there is a requirement to develop some better interpolation schemes which are not only computationally efficient but also would give better image quality by preserving HF information in the up-scaled images. The research problem taken up is presented in the next section.

1.5 Problem Statement

From the literature review, it is apparent that the polynomial based interpolation schemes are computationally efficient but produce undesirable artifacts such as blurring and aliasing. Though edge-directed and transform domain interpolation schemes preserve the edge information and fine details effectively than polynomial based interpolation schemes, they are computationally more complex. Hence, it is felt that there is further scope to develop efficient up-scaling schemes which are not only computationally efficient but also produce better visual quality by preserving the fine details and edge information. Hence, in this current research work, efforts are made to improve the performance of the existing 2-D polynomial based interpolation schemes through the incorporation of various spatial domain pre-processing, post-processing and composite techniques so as to obtain a better up-scaled image quality along with reduced computational complexity. The polynomial based interpolation schemes are taken-up for up-gradation because of their reduced computational complexities than transform-domain schemes and their suitability for real-time applications.

Hence, the objective of the research work is to develop various efficient but simple pre-processing, post-processing and composite schemes using Lanczos-3 interpolation to effectively restore the HF contents in the up-scaled images for various online and off-line applications.

Although all the polynomial based interpolation schemes show improvement through incorporation of various pre-processing and post-processing schemes, Lanczos-3 interpolation scheme is preferred because of its better HF restoration performance with less blurring. In addition, it is free from artifacts such as ringing which is found in DCT interpolation scheme. Furthermore, it is computationally less complex than transform-domain schemes such as DCT and Wavelet. Hence, Lanczos-3 interpolation is preferred among various polynomial and transform-domain up-scaling schemes because of its overall better performance.

1.6 Methodologies Adopted

In order to overcome the challenges posed by various interpolation schemes, many pre-processing, post-processing and composite algorithms are proposed in this thesis.

The **pre-processing algorithms** are based on higher order derivatives and unsharp masking. The term *pre-processing* refers to processing the low-resolution input image prior to image up-scaling. The various pre-processing algorithms proposed in this thesis are :

- *Laplacian of Laplacian (**LLGP**) based Global Pre-processing Scheme*
- *Hybrid Global Pre-processing (**HGP**) Scheme*
- *Iterative Laplacian of Laplacian based Global Pre-processing (**ILLGP**) Scheme*
- *Unsharp Masking based Pre-processing (**UMP**)*
- *Iterative Unsharp Masking (**IUM**)*
- *Error based Up-sampling (**EU**) Scheme*

The **post-processing schemes** are local adaptive schemes and work on high resolution, up-scaled images. The various post-processing algorithms proposed in this thesis are:

- *Local Adaptive Laplacian (**LAL**) based Post-processing Algorithm*
- *Fuzzy Weighted Laplacian (**FWL**)based Post-processing Algorithm*
- *Legendre Functional Link Artificial Neural Network (**LFLANN**) based Post-processing Algorithm*

There is a scope of further improvement of the stand-alone pre-processing and post-processing schemes by combining them through **composite schemes**. Two spatial-domain composite schemes, CS-I and CS-II are proposed in this thesis to tackle non-uniform blurring in an up-scaled image and are given below.

- *Composite Scheme **(CS-I)** using Iterative Laplacian and Local Adaptive Laplacian*
- *Composite Scheme **(CS-II)** using Iterative Laplacian of Laplacian and Fuzzy Weighted Laplacian*

1.7 Performance Measures

The performances of different algorithms are evaluated by objective and subjective measures. The objective performance of an image is evaluated by determining error and error-related parameters mathematically. However, for subjective evaluation, the image has to be observed by several observers [7, 8].

There are various metrics used to measure the objective performance of an image. The commonly used metrics are mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), peak signal to noise ratio (PSNR) and universal quality index (UQI). However, objective performance metrics like PSNR and UQI are taken here for objective evaluation. The mean square error (MSE) and peak-signal-to-noise-ratio (PSNR) of the up-sampled, restored image are computed and compared with other existing algorithms for the objective evaluation. Universal quality index (UQI) is another quality metric to measure the image quality by combining correlation, average luminance and contrast level similarity and hence becomes a good performance measure. It takes care of human visual system (HVS) and hence its performance is quite similar to what a human expert observes. Therefore, it corresponds to subjective evaluation to some extent [8].

Let the original and restored images be represented by $f(x,y)$ and $\hat{f}(x,y)$ respectively. Let them be of size $M \times N$. Then **MSE** is given by,

$$MSE = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}(\hat{f}(x,y) - f(x,y))^2}{M \times N} \tag{1.15}$$

The **PSNR** is defined in logarithmic scale, and is expressed in dB. It is a ratio of peak signal power to noise power. The PSNR is defined as:

$$PSNR = 10\log_{10}\left[\frac{\sum_{x=1}^{M}\sum_{y=1}^{N}1^2}{\sum_{x=1}^{M}\sum_{y=1}^{N}(\hat{f}(x,y)-f(x,y))^2}\right] \tag{1.16a}$$

$$=10\log_{10}\left[\frac{MN}{\sum_{x=1}^{M}\sum_{y=1}^{N}(\hat{f}(x,y)-f(x,y))^2}\right]$$

$$=10\log_{10}\left[\frac{1}{\frac{\sum_{x=1}^{M}\sum_{y=1}^{N}(\hat{f}(x,y)-f(x,y))^2}{MN}}\right]=10\log_{10}\left[\frac{1}{MSE}\right]dB$$

$$PSNR = 10\log_{10}(1/MSE)\ dB \tag{1.16b}$$

provided the signal lies in the range [0, 1]. On the other hand, if the signal is represented in the range of [0,255], the numerator in (1.16) will be $(255)^2$ instead of 1.

The universal quality index (UQI) [8] is used to measure the quality of an image. The universal quality is modeled by considering three different factors and is defined by:

$$UQI = \frac{\sigma_{f\hat{f}}}{\sigma_f\,\sigma_{\hat{f}}}\cdot\frac{2\,\bar{f}\,\bar{\hat{f}}}{\bar{f}^2+\bar{\hat{f}}^2}\cdot\frac{2\sigma_f\,\sigma_{\hat{f}}}{\sigma_f^2+\sigma_{\hat{f}}^2} \tag{1.17a}$$

where,

$$\bar{f} = \frac{1}{M\times N}\sum_{x=1}^{M}\sum_{y=1}^{N}f(x,y) \tag{1.17b}$$

$$\bar{\hat{f}} = \frac{1}{M\times N}\sum_{x=1}^{M}\sum_{y=1}^{N}\hat{f}(x,y) \tag{1.17c}$$

$$\sigma_f^2 = \frac{1}{MN-1}\sum_{x=1}^{M}\sum_{y=1}^{N}(f(x,y)-\bar{f}(x,y))^2 \tag{1.17d}$$

$$\sigma_{\hat{f}}^2 = \frac{1}{MN-1}\sum_{x=1}^{M}\sum_{y=1}^{N}(\hat{f}(x,y)-\bar{\hat{f}}(x,y))^2 \tag{1.17e}$$

$$\sigma_{f\hat{f}} = \frac{1}{MN-1}\sum_{x=1}^{M}\sum_{y=1}^{N}(f(x,y)-\bar{f}(x,y))(\hat{f}(x,y)-\bar{\hat{f}}(x,y)) \tag{1.17f}$$

The UQI consists of three components. The first component is the correlation between the original image, f and the restored image, $\hat{f}$. It measures the degree of linear correlation between them and its dynamic range is [-1, 1]. The second component, with the range of [0, 1], measures the closeness between the average luminance of f and $\hat{f}$. It reaches the maximum value of 1 if and only if $\bar{f}$ equals $\bar{\hat{f}}$. The standard deviations of these two images, σ_f and $\sigma_{\hat{f}}$ are also regarded as estimates of their contrast levels. The value of contrast level ranges from 0 to 1 and the optimum value of 1 is achieved only when $\sigma_f = \sigma_{\hat{f}}$. Hence, combining the three parameters: correlation, average luminance similarity and contrast level similarity, the universal quality index (UQI) becomes a very good performance measure.

Error image is an important tool to study the HF restoration performance of various algorithms. The error is calculated by taking the absolute value of the difference between the restored image and the original image. For display purpose, the error is scaled up by an appropriate scaling factor for proper visibility. Let $f(x, y)$ and $\hat{f}(x, y)$ be the original and restored image. The error $e(x, y)$ for different algorithms is computed for performance analysis and is given by,

$$e(x, y) = \left| \hat{f}(x, y) - f(x, y) \right| \tag{1.18}$$

Computational complexity plays a major role to define the effectiveness and suitability of the proposed algorithms for a specific application. Execution time (T_E) is related to the complexity of an algorithm. As complexity of an algorithm increases, so does the execution time. For simulating the algorithms, a digital computing platform having an Intel® core™ i3-3217U processor running at 1.8 GHz clock with 3.89 GB usable RAM and 64 bit Window-10 operating system has been employed. MATLAB R2010a software is used for simulation of all these algorithms in the above mentioned digital computing platform. To understand the computational complexity of the algorithms, simulation studies are carried out on this platform to find the execution time for various algorithms.

1.8 Chapter-wise Organization of the Thesis

The chapter-wise organization of the thesis is outlined here.

Chapter 1: Introduction

Preview; Introduction to image up-scaling; Fundamentals of image interpolation; Some basic interpolation schemes; Advantages and disadvantages of various existing interpolation schemes; Problem statement; Methodologies adopted; Performance measures; Chapter-wise organization of the thesis; Conclusion.

Chapter 2: Literature Review

Preview; image up-scaling using 2-D interpolation; Review on polynomial, edge directed and transform-domain interpolation schemes; Study of some existing interpolation schemes; Performance analysis; Conclusion.

Chapter 3: Spatial-domain Pre-processing Algorithms using Higher Order Derivatives

Preview; Laplacian of Laplacian based global pre-processing (LLGP); Hybrid global pre-processing (HGP); Iterative Laplacian of Laplacian based global pre-processing (ILLGP); Experiment and simulation; Results and discussion; Conclusion.

Chapter 4: Pre-processing Algorithms using Unsharp Masking

Preview; Unsharp masking based pre-processing (UMP); Iterative unsharp masking (IUM); Error based up-sampling (EU); Experiment and simulation; Results and discussion; Conclusion.

Chapter 5: Post-processing Algorithms using Soft-computing Techniques

Preview; Local adaptive Laplacian; Fuzzy weighted Laplacian; Legendre functional link artificial neural network (LFLANN); Results and discussion; Conclusion.

Chapter 6: Development of Some Spatial Domain Composite Algorithms

Preview; Composite scheme (CS-I) using iterative Laplacian and local adaptive Laplacian; Composite scheme (CS-II) using iterative Laplacian of Laplacian and fuzzy weighted Laplacian; Results and discussion; Conclusion.

Chapter 7: Conclusion

Comparative analysis; Conclusion; Scope for future work

1.9 Conclusion

In this chapter, introduction to image up-scaling, fundamentals of image interpolation and some basic interpolation schemes are discussed. The challenges associated with various polynomial, edge-directed and transform based interpolation schemes are discussed in brief literature review. Based on the challenges, the research problem is clearly stated. Various methodologies adopted based on pre-processing, post-processing and composite schemes to overcome the challenges of the existing schemes are reported. In addition, image metrics associated with performance evaluation of various algorithms are discussed. Finally, the organization of the thesis is presented.

The literature review on various interpolation schemes is described in the subsequent chapter.

Chapter 2

Literature Review

Preview

In this chapter a study of different interpolation schemes has been presented. The advantages and limitations of different interpolation schemes are discussed in the prospective of visual quality and their suitability for various online and offline applications.

Image up-scaling or super-resolution can be accomplished using single image or through multiple images. Image super-resolution using multiple images produces a high resolution (HR) image of better visual quality. But it is computationally more demanding and hence is not preferred for real-time applications. Therefore, the focus is more on super-resolution using single image which requires less processing time. Resolution enhancement using single image can be accomplished through interpolation-based, learning-based and reconstruction-based schemes [9]. Amongst the various single image resolution enhancement methods, interpolation-based techniques are preferred for their simplicity and suitability for various online applications. The interpolation based up-sampling schemes are further classified into polynomial-based, edge-directed and transform-domain techniques. Various 2-D interpolation schemes are discussed in this chapter. The topics covered in this chapter are:

- *Preview*
- *Image Up-scaling using 2-D Interpolation*
- *Review of Polynomial based Interpolation Schemes*
- *Review of Edge Directed Interpolation Schemes*
- *Review of Transform Domain Interpolation Schemes*
- *Study of some Existing Interpolation Schemes*
- *Comparative Analysis*
- *Conclusion*

2.1 Image Up-scaling using 2-D Interpolation

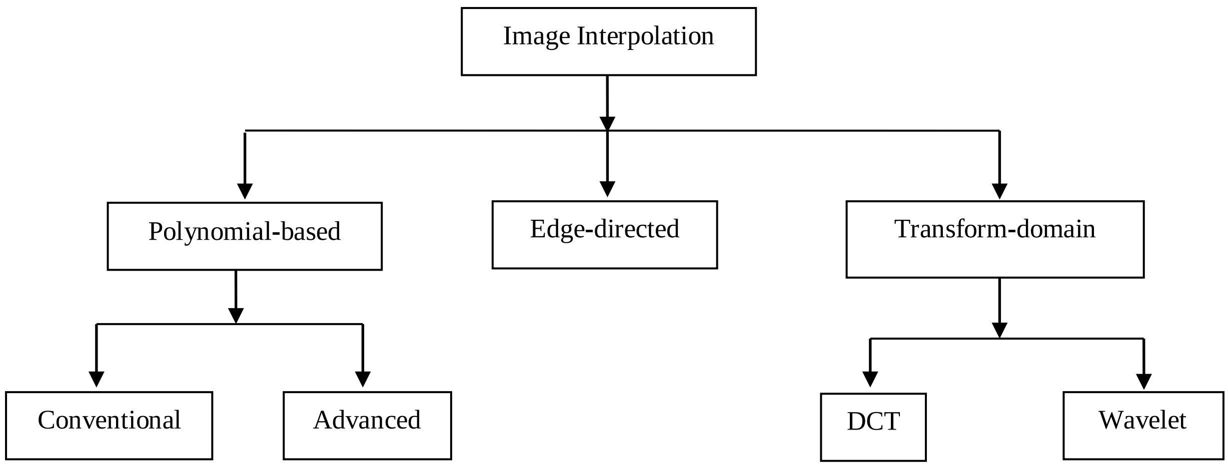


Fig. 2.1 Classification of interpolation schemes

Image interpolation is the process of estimating the intermediate values of a spatially continuous image from a set of its discrete samples. It typically estimates an unknown pixel value within a neighborhood from the known pixel neighbors. 2-D interpolation is used for resolution enhancement of a low resolution image data and is employed in many online applications such as video streaming, HDTV and video surveillance. Many image interpolation schemes, developed so far, are broadly classified into three categories. They are namely polynomial based, edge directed and transform domain interpolation schemes. The polynomial interpolation scheme is classified into conventional and advanced-polynomial based interpolation. Likewise, the transform-domain interpolation schemes are categorized into DCT and wavelet based interpolation. In addition, there are several edge-directed interpolation schemes employed for efficient image up-sampling. The detail classification of interpolation schemes are shown in Fig. 2.1. A review of various interpolation schemes is given in the subsequent sections.

2.2 Review of Polynomial based Interpolation Schemes

The polynomial based interpolation schemes are based on the convolution of sampled data with an interpolation kernel. They are broadly classified into conventional and advanced polynomial based interpolation schemes. In case of conventional polynomial based interpolation schemes [3-5], the pixel intensity at an unknown location is directly estimated based on convolution without considering the local statistical features of an image and hence are not adaptive techniques. Most of such techniques are simple, easy to implement and computationally less complex and so are suitable for various real-time applications. However, they suffer from aliasing, blurring at the edges in the reconstructed image. Various polynomial based interpolation schemes are nearest-neighbor, bilinear, bicubic, cubic-spline [3], lanczos-3 [10-13] etc. Some of the basic polynomial based interpolation schemes such as nearest-neighbor [3], bilinear [5] and bicubic [4] interpolation schemes are already discussed in Chapter-1.

The problems encountered in conventional polynomial based interpolation schemes are to some extent are resolved using various advanced polynomial based techniques at the cost of additional computational complexities [14-36]. The advanced polynomial based schemes modify or hybridize the conventional schemes for performance improvement. These schemes can also be adaptive and can be employed in various real-time applications because of their reduced computational complexities. In this section, some of the advanced polynomial based interpolation schemes are discussed.

Lehman et al. [14] have developed a high-degree B-spline interpolation scheme for medical image processing applications. The high-degree B-spline interpolation has superior spectral characteristics, less interpolation error in comparison to low-degree B-spline interpolation. In addition, the proposed scheme has reduced computational complexity. For this reason, it is employed in numerous medical image processing applications which demand high precision and less interpolation artifacts.

Chung et al. [15] have introduced a fractal based image enhancement technique to reduce the image degradation due to sub-optimal contractive mapping. The proposed technique preserves the details in the edge regions and at the same time maintain the smoothness of the slowly varying or flat regions. This technique shows better performance than conventional bilinear and bicubic interpolation techniques.

Hadhoud et al. [16] have suggested an adaptive image interpolation based on local activity levels. In this case, different conventional interpolation schemes such as bilinear, bicubic, cubic-spline and warped distance technique are modified based on the level of activity in the local regions of an image. This process is accomplished by assigning different weights to the pixels used in the interpolation process according to the degree of local information. The warped distance is based on modifying a distance based on homogeneity or in-homogeneity in a neighbor. The proposed adaptive technique shows better performance than the conventional interpolation and warped distance technique.

Blu et al. [17] have presented an original method to improve piecewise-linear interpolation with uniform knots. In this paper, the sampling knots are shifted by a fixed amount while enforcing the interpolation property. The optimal shift that produces maximum performance improvement is found to be roughly 0.2 for linear interpolation. Experimental results show the performance improvement of the proposed scheme over linear interpolation.

Malvar et al. [18] have introduced a new linear interpolation technique for demosaicing of bayer-patterned color images. Demosaicing is a digital image processing technique used to reconstruct a full color image from in-complete color samples through color filter array interpolation. The proposed technique is simple and show better performance than bilinear interpolation.

Aly et al. [19] have introduced a image up-sampling using total variation regularization with a new observation model. In this paper, a formulation and analysis for the image up-sampling problem employing total-variation regularizer is developed. A new observation and the total variation regularizer are used at the formation level. The formation is set as an optimization problem and is numerically solved by a level-set motion algorithm. The proposed scheme shows better performance than many state-of-art techniques.

Zhen et al. [11] have employed four image interpolation techniques for down-sampled ultrasound breast phantom data acquired using Fischer's full field digital mammography so as to reduce the size of large images for online diagnosis. This scheme improves the processing speed without sacrificing the quality of ultrasound images and hence is suitable for real-time applications. Experimental results show that Lanczos algorithm performs better than bilinear, bicubic and wavelet domain interpolation schemes.

Shen et al. [20] have developed a modified Laplacian filter and an intensity correction technique for image resolution enhancement. The modified Laplacian filter restores the frequency components degraded during the down sampling and averaging process. The intensity correction gives a better visual effect to an image subjected to different degree of resolution enhancement. The proposed technique is computationally efficient and gives better results than bilinear and bicubic interpolation.

Xiangjian et al. [21] have introduced a reversible fast image translation and rotation scheme based on a hexagonal structure. Images are conventionally represented on a square pixel structure. However, the hexagonal structure provides a more flexible and efficient way to perform image translation and rotation without the loss of information. The resolution of the image is also retained by virtual hexagonal structure during image transformation. The hexagon structural mode of representation shows better performance than traditional square structural mode.

Xiangjian et al. [22] have developed an approach to edge detection on a virtual hexagonal structure employing bilinear interpolation that converts an image from square structure to hexagonal structure. The experimental results show better edge detection accuracy.

Gharavi et al. [23] have presented a spatial interpolation algorithm for intra-frame error concealment. In this method, the image regions that are being affected due to loss of packets can be restored using a composite algorithm comprising bilinear interpolation and edge detection. The edge detection technique is based on Hough transform meant for performance improvement of bilinear interpolation.

Bera et al. [24] have proposed multirate scan conversion of ultrasound images using warped distance based adaptive bilinear interpolation. In this method, the regions with fine details and edges are enhanced and at the same time the smoothness of the flat regions are preserved employing adaptive bilinear interpolation with reduced computational complexities. This shows the suitability of the proposed scheme for online medical applications.

Shen et al. [25] have developed a novel interpolation algorithm for nonlinear omni-catadioptric images to overcome the reduced visual accuracy of omni images taken by nonlinear catadioptric camera. The proposed interpolation scheme improves the resolution of omni images to compensate the lack of visual content. The camera property of interpolated images is also preserved by utilizing epipolar geometry constraint of nonlinear images. The proposed scheme shows better performance than bilinear and bicubic interpolation schemes.

Jiang et al. [26] have developed a configurable system for role-specific video imaging during laparoscopic surgery. In this paper, a configurable system is introduced that implements real-time and role specific imaging. The role-specific imaging displays panoramic and close-up views at the same time. Two separate video bit streams are displayed in real-time for this purpose employing real-time interpolation schemes. The user can dynamically adjust the frame rate, interpolation method and zooming factor through real-time configurable setting. The performance of the system is monitored for various real-time interpolation schemes such as nearest, bilinear, bicubic and lanczos interpolation.

Kang at al. [27] have introduced a new real-time super-resolution technique for digital zooming using finite kernel based edge orientation and truncated image restoration. The proposed method minimizes the interpolation artifacts and restores high frequency details using finite impulse response (FIR) filter. In this case, the edge orientation is precisely estimated using steerable filters with edge refinement. The input image is then adaptively interpolated along the estimated edge orientation. The authors claim that the proposed scheme produces better results with reduced jagged edges and other interpolation artifacts than other existing schemes.

Xiao et al. [28] have proposed an image zooming method using hierarchical structure in the year 2013. In this paper, retinex model is employed to decompose image into low frequency and high frequency layers. Corresponding to low frequency layer, a low-pass filter is realized to filter the mirror signals which are introduced during up-sampling. A nonlinear iterative method based on heat equation is used to reconstruct the high frequency layer that contains the fine details and texture information. The resulting zoomed image is obtained by merging both the layers. Experimental results show that the proposed algorithm improves the defects of linear and non-linear interpolation algorithms.

Dai et al. [29] have developed a directionally adaptive Cubic-spline interpolation using optimized interpolation kernel and edge orientation for mobile digital zoom system. To reduce blurring and jagged artifacts in linear and cubic-spline interpolation, the authors have suggested a new directional adaptive cubic-spline interpolation scheme. The proposed scheme is based on cubic-spline interpolation using edge orientation and the employment of an optimized interpolation kernel using kernel map. Experimental results show better zooming performance with reduced interpolation artifacts than various existing schemes.

Zhong et al. [30] have proposed an adaptive image amplification method with integer multiples for superior zooming. Generally, a pixel in a low-resolution (LR) image corresponds to a block of unknown pixels in the corresponding high-resolution (HR) image. For each pixel in the LR belonging to an edge region, weighted least square estimation algorithm is adopted to get its eight orientation parameters for estimating its associated block of pixels in the HR image. Subsequently, each LR pixel is substituted by linear weighted summation of its eight surrounding HR pixels so that the model parameters are obtained. Finally, correction is done on each block to refine each HR pixel. Experimental results show superiority of the proposed scheme over other zooming methods.

Hung et al. [31] have introduced a modified bicubic interpolation scheme and employed it for frame enhancement in a camera based traffic monitoring. The proposed technique modifies the bicubic interpolation by considering edge map on the interpolated images for reducing the blurring artifacts while preserving the edge information effectively. The proposed method proves a promising pre-processing stage to perform traffic analysis on the acquired video at low resolution.

In addition, there are many advanced polynomial based interpolation schemes which are adaptive [32-36] and provide superior resolution enhancement than conventional polynomial based interpolation schemes.

The next section presents various edge directed interpolation schemes.

2.3 Review of Edge-directed Interpolation Schemes

Edges are visually attractive to human visual system. The interpolation quality is considered to be better if the edges of an interpolated image are sharp and free from blurring and other artifacts. The basic objective of various edge-directed interpolation schemes is to preserve sharpness during the up-sampling process. Various edge-directed interpolation schemes are discussed below.

Li et al. [37] have introduced a new edge-directed interpolation (NEDI) scheme for natural images. In this method, the local covariance coefficients are estimated from an LR image. These coefficients are used to adaptively control the interpolation at higher resolution based on duality between LR covariance and HR covariance. The proposed scheme performs better than the conventional linear interpolation schemes.

Loung et al. [38] have proposed a method for interpolating images with repetitive structures. In this method, an unknown pixel is estimated based on information of the entire image unlike other conventional interpolation schemes and shows performance improvement over those conventional schemes.

Tam et al. [39] have developed a modified edge-directed interpolation scheme which is the modified version of new edge directed interpolation (NEDI). The prediction error accumulation problem in NEDI is eliminated by adopting a modified training window structure. The proposed scheme further extends covariance matching to multiple directions for suppressing covariance mismatch and shows much better subjective performance over existing schemes.

Shi et al. [40] have proposed context based adaptive image resolution up-conversion in which an LR image patch is used as a context in which missing HR pixels are estimated. The context is quantized into classes and for each class an adaptive linear filter is designed using a training set. The training set incorporates the prior knowledge of point spread function, edges, texture, smooth shades, etc. into the up-conversion filter design. Experimental results show improved performance of the algorithm over the existing schemes.

Dung et al. [41] have proposed a selective data pruning-based compression scheme to improve the rate distortion relation of compressed images and video sequences. The original frames are pruned to a smaller size before compression. After decoding, they are interpolated back to their original size by an edge-directed interpolation method. The data pruning phase is optimized to obtain the minimal distortion in the interpolation phase. Furthermore, a novel high-order interpolation is proposed to adapt the interpolation to several edge directions in the current frame. This high-order filtering uses more surrounding pixels in the frame than the fourth-order edge-directed method and it is more robust. The algorithm is also considered for multi-frame based interpolation by using spatio-temporally surrounding pixels coming from the previous frame. Simulation results are shown for both image interpolation and coding applications to validate the effectiveness of the proposed methods.

Mishiba et al. [42] have proposed an edge adaptive image interpolation to estimate a HR image from its LR counterpart using constrained least squares. The adaptive image interpolation makes use of edge-directed smoothness filter and constrains the interpolated image to have edge-directed smoothness and fidelity to the original image based on observation model.

Shaode et al. [43] have proposed an edge-directed interpolation (EDI) method and applied it on a group of fetal spine MR images to evaluate its feasibility and performance. This method extracts edge information using canny edge detector and perform further pixel modification based on the edge information. Initially, low-resolution (LR) images of fetal spine are interpolated into high-resolution (HR) images with targeted scaling factor using bilinear interpolation. Later on, the edge information from LR and HR images is put into a twofold strategy to sharpen or soften edge structures. Finally, a HR image with well-preserved edge structures is generated. The proposed method provides proper resolution enhancement for accurate medical diagnosis.

There are many more edge directed interpolation schemes [44-65] which not only provide better objective and subjective performance than the conventional interpolation schemes but are also employed for various up-sampling applications. However, these algorithms demand more processing time due to their adaptive and complex structures.

In the next section, various transform domain interpolation schemes are discussed.

2.4 Review of Transform-domain Interpolation Schemes

DCT and wavelet are widely used transform-domain interpolation schemes. Image resizing in DCT domain shows very good result in terms of scalability and image quality. However, these techniques suffer through undesirable ringing artifacts and computationally more complex than conventional spatial-domain interpolation techniques. Likewise, various wavelet domain resolution enhancement schemes have been developed to effectively preserve the high frequency contents in up-scaled images but demand more computational time as compared to conventional interpolation schemes. Some of the DCT and wavelet based interpolation schemes are discussed in this section.

Alkachouh et al. [66] have developed a technique using DCT based interpolation of blocks in images to restore a block in an image, using a reduced set of border pixels. In case of DCT transformed blocks, most of the high frequency coefficients are likely to be negligible and can be dropped. Generally, the block to be transformed is made of the missing block and the border pixels. It is by setting as many DCT coefficients to zero, a system of linear equation results, whose solution yields approximations for the values of the lost pixels**.** The solution of the system is actually an interpolation operation which yields approximations of the unknown values. Dugad et al. [67] have proposed a fast scheme for image size change in compressed domain. The proposed algorithm for down-sampling and up-sampling in DCT domain is computationally faster and produces visually sharper images. Various other algorithms [68, 69] are developed for image resizing in compressed domain and show performance improvement over various existing schemes. Shu et al. [70] proposed an efficient down-scaling algorithm for video transcoding employing DCT-based interpolation. In this method, the spatial resolution of a video signal is reduced using DCT so as to transmit over a bandwidth constrained channel. The proposed scheme gives satisfactory down-scaling performance compared to existing methods.

Park et al. [71] have developed a fast arbitrary-ratio image resizing method for recovery of compressed images. The downscaling process in the DCT-domain can be implemented by truncating high-frequency coefficients, whereas the up-scaling process is implemented in the DCT domain by padding zero coefficients to the high-frequency part. The proposed method combines a fast inverse and forward DCT of composite length for arbitrary-ratio up-scaling or down-scaling. According to the resizing ratio, truncating the high-frequency coefficients and padding zeros are appropriately considered by combining the inverse DCT and forward DCT. The proposed method shows a good peak-signal-to-noise-ratio and less computational complexity compared with the spatial-domain and previous DCT-domain image resizing methods. Similar image resizing techniques are proposed with better scalability and improved up-scaled image quality [72, 73].

Shin et al. [74] have proposed an adaptive up-sampling method using DCT for spatial scalability of scalable video coding (SVC) using type-II DCT for H.264 SVC up-scaling. The proposed scheme is based on a combination of the forward and backward type-II discrete cosine transform (DCT). Here, a fast algorithm of type-II DCT-based up-sampling method is also proposed. For further improvement of the up-sampling performance, an adaptive filtering method in the type-II DCT up-sampling is introduced, which applies different weighting parameters to DCT coefficients. The proposed adaptive up-sampling method shows a much improved PSNR in comparison with the recent H.264 SVC up-sampling. Wu et al. [75] have introduced a new hybrid DCT-wiener-based interpolation scheme for video intra-frame up-sampling. The proposed scheme exploits the advantages of DCT-domain and spatial-domain interpolation schemes to develop an improved up-sampling filter for better resolution enhancement over the existing schemes. Some modified version of DCT-wiener based algorithms are proposed for more improved performance [76, 77]. Lim et al. [78] have developed a DCT based up-scaling scheme to counter ringing artifacts in the up-scaled image.

Wavelet-domain interpolation is a type of transform-domain interpolation scheme which is widely used in image up-scaling applications. Some of the wavelet domain interpolation schemes are discussed below. Zhao et al. [79] have introduced wavelet image super-resolution using wavelet domain Hidden Markov Tree (HMT) model. In this paper, image super-resolution is formulated as a constrained optimization problem using HMT. Cycle spinning technique is employed to suppress the artifacts in the HR image.

Temizel et al. [80] have developed an image resolution enhancement algorithm using cycle –spinning. In the proposed scheme, an HR image is generated by wavelength domain zero padding of a LR image followed by inverse wavelet transform. The cycle-spinning technique is used for quality improvement of HR image.

Wu et al. [81] have developed a wavelet-based image resolution enhancement technique in which the edges are enhanced by introducing an intermediate stage of stationary wavelet transform (SWT). Experimental results show the superiority of the proposed scheme over conventional 2-D up-sampling scheme.

Demiral. et al. [82] have proposed an image super-resolution employing discrete and stationary wavelet decomposition. The authors propose an image resolution enhancement technique based on interpolation of the high frequency sub-band images obtained by discrete wavelet transform (DWT) and the input image. The edges are enhanced by introducing an intermediate stage by using stationary wavelet transform (SWT). DWT is applied in order to decompose an input image into different sub-bands. Afterwards, the high frequency sub-bands as well as the input image are interpolated. The estimated high frequency sub-bands are being modified by using high frequency sub-band obtained through SWT. Finally, all these sub-bands are combined to generate a new high resolution image employing inverse DWT (IDWT).

Experimental results show that there is a improvement in the performance of the proposed technique compared to the conventional and state-of-art image resolution enhancement techniques.

Besides the polynomial, edge-directed and transform-domain interpolation schemes, there are many other schemes which are used for image super-resolution. They are categorized as learning-based, reconstruction-based, multi-frame based image super-resolution [83-117]. A recent study on image super-resolution reveals that the sparse and deep-learning based super-resolution schemes [120-128] show better performance under various constraints. Some of the recent sparse and deep-learning based super-resolution techniques are discussed here.

Dictionaries are crucial in sparse coding based algorithms for image super-resolution. Sparse coding is a typical unsupervised learning method to study the relationship between the patches of high- and low-resolution images. However, when an LR image and its corresponding HR image are represented in their feature spaces, the two sets of dictionaries and the obtained coefficients have intrinsic links which have not yet been studied. Motivated by the development on nonlocal self similarity and manifold learning, Lu et al. [120] proposed a novel sparse coding method to preserve the geometrical structure of the dictionary and the sparse coefficients of the data to reduce the super-resolution reconstruction artifacts. The purpose of incorporating nonlocal self similarity and manifold learning is to have good reconstruction and discrimination properties which can enhance the learning performance.

Dong et al. [121] have developed an effective image interpolation scheme by nonlocal autoregressive modeling (NARM) and incorporated it in the sparse representation model (SRM). The conventional SRM method becomes less effective in the image interpolation problem because the data fidelity term fails to impose structural constraint on the missing pixels. This problem is properly addressed by exploiting the image non-local self similarity with NARM. The NARM acts as a new structural data fidelity term in SRM by connecting a missing pixel with its nonlocal neighbors. It reduces much coherence between the sampling matrix and the sparse representation dictionary so that SRM becomes more effective for image interpolation. Experimental results reveal that the proposed scheme achieve better performance than various state-of-the-art image interpolation schemes.

M. Nazzal and H. Ozkaramanli [122] have introduced a single image super-resolution approach that employs dictionary learning and signal reconstruction in wavelet domain. The proposed algorithm makes use of DWT and classification to design structured dictionaries which are compact and effectively directional since they inherit the structural and directional properties of the respective wavelet subband. The diagonal wavelet details are classified as diagonal and anti-diagonal. For each horizontal and vertical subband, a pair of dictionaries (LR and HR) is designed. Since wavelet transform is unable to separate diagonals and anti diagonal orientations, two pairs of dictionaries are designed for the diagonal detail subbands. Moreover, the proposed algorithm reduces the dictionary learning computational complexity by designing compactly sized structural dictionaries and shows better performance than the existing schemes.

Various existing sparse coding based super-resolution techniques partition the image into overlapped patches and process each patch separately. These methods however ignore the consistency of pixels in overlapped patches which is a strong constraint for image reconstruction. To resolve this issue, Gu et al. [123] proposed a convolutional sparse coding based super-resolution. This method decomposes the whole image by filtering which naturally takes the consistency of pixels in overlapped patches into consideration. This process involves decomposition of LR images into LR sparse feature map using a set of filters, prediction of HR feature maps from the LR using a mapping function and reconstruction of HR images from the predicted HR feature maps through convolution operations using a set of filters. The proposed algorithm doesn’t need to divide the image into overlap patches and exploits the image global correlation for better reconstruction of image local structures.

Ahmed et al. [124] have presented a selective sparse coding algorithm with a directionally structured dictionary learnt through a coupled K-singular value decomposition (K-SVD) algorithm for single image super-resolution. In this algorithm, the LR and HR patches are assumed to have the same sparse coefficients. Furthermore, the learning of dictionaries is performed using K-SVD algorithm which represents HR and LR patches by enforcing the coefficients of one on the other alternately. The algorithm is made more efficient by introducing selective sparse coding using clustering of data by creating a set of vertical, horizontal and one non-directional template. These templates have a directional structure and are helpful in creating more directional and compact dictionaries. Two sets of HR and LR dictionaries are generated along with one non-directional dictionary. A given LR image can be up-scaled by using these HR dictionaries based on its LR counterpart and correlation criteria. Experimental results show the superiority of the proposed scheme over the existing schemes.

Han et al. [125] have proposed a novel sparse based technique to generate high resolution hyper spectral (HR-HS) image from the available low resolution hyper spectral (LR-HS) and high resolution multi spectral (HR-MS) images using data-guided sparse spectral representation. The proposed scheme makes use of HS dictionary from the input LR-HS image and transforms it to RGB dictionary for calculating the sparse representation of all HR pixels in the input HR RGB image. Subsequently, the sparse representation of the spectrum based on RGB dictionary is used to reconstruct hyper spectrum by combining the HS dictionaries for recovering the HR hyper spectral image. Experimental results on two public hyper spectral datasets validate that the proposed method achieves promising performance than the state-of-the-art methods.

Face hallucination is an example of the image super-resolution problem, where the HR face images can be obtained from the LR ones. Pei et al. [126] have proposed a gradient constrained sparse representation based face hallucination algorithm which incorporates gradient information and reweighted constraint into the image super-resolution problem to achieve better performance. The iterative algorithm refines the reconstructed HR images. The experimental results on several face data bases show the better performance of the proposed algorithm than other baseline algorithms.

Recently, various deep learning methods are applied to super-resolution problem and are giving promising results. Guo et al. [127] have introduced a new low-complexity and effective super-resolution algorithm called super-resolution with coupled back propagation (SR-CBP). This algorithm builds two deep neural networks called coupled auto-encoder networks (CAN) that capture the features of HR and LR images. SR-CBP allows joint training of the LR and HR networks to have middle layer representations that agree for a LR and its corresponding HR image. For a LR input image, its middle layer representation obtained through the trained LR network can be used by the HR network to generate a HR image. In addition, SR-CBP has smaller memory and less computation requirement than the state-of-the-art deep learning based super-resolution method.

Sharma et al. [128] have developed an end-to-end deep learning based framework for noise resilient super-resolution algorithm which performs de-noising and super-resolution simultaneously by preserving the textural details of an image without blurring. In this algorithm, a novel deep learning based image super-resolution architecture, termed as coupled deep convolutional auto-encoder (CDCA) is used for better performance. It computes the convolutional features of LR and HR image patches and learns the nonlinear function that maps these convolutional features of LR image patches to their corresponding convolutional features of HR image patches. Initially, stacked sparse de-noising auto encoder (SSDA) is learned for LR image de-noising. The proposed CDCA is learned for image super-resolution. A cascaded deep learning network is developed by combining SSDA and CDCA and is employed as one integral network, where the pre-trained weights are taken as initial weights. In fine tuning, all the layers of the combined end-to-end network are jointly optimized to perform image de-noising and super-resolution simultaneously. Experimental results show the proposed algorithm shows better performance than various state-of-the-art schemes in terms of PSNR and SSIM metrics.

An analytical study of some well-known state-of-art schemes is presented in the next section.

2.5 Study of Some Existing Interpolation Schemes

2.5.1 Lanczos-3 Interpolation Scheme [12]

Lanczos is a spatial domain interpolation technique which is implemented by multiplying a sinc function with a sinc window which is scaled to be wider and truncated to zero outside the main lobe. In case of Lanczos-3 interpolation, the main lobe of the sinc function along with the two subsequent side lobes on either side is used as a sinc window. The Lanczos window is a product of sinc function, $sinc(x)$ with the scaled version of it, $sinc(x/a)$ restricted to the main period $-a \le x \le a$, to form a convolution kernel for up-sampling the input field [10]. In 1-D, the expression for Lanczos interpolation is given by,

$$L(x) = \begin{cases} sinc\,(x)\, sinc\,(x/a), & -a \le x \le a \\ 0, & otherwise \end{cases} \tag{2.1}$$

where 'a' is a positive integer, typically 2 or 3, used for controlling the size of the kernel. The parameter 'a' corresponds to the number of lobes of the sinc function. The parameter a is taken two for Lanczos-2 interpolation. The three lobed Lanczos windowed sinc function (Lanczos-3) is given by,

$$Lanczos\ 3(x) = \begin{cases} \dfrac{\sin(\pi x)}{\pi x} \dfrac{\sin(\pi x/3)}{\pi x/3}, & -3 \le x \le 3 \\ 0, & otherwise \end{cases} \tag{2.2}$$

For a two dimensional low resolution image $g(x, y)$, an interpolated value at an arbitrary point (x_0, y_0) using Lanczos-3 interpolation is given by,

$$\hat{f}(x_0, y_0) = \sum_{i=\lfloor x_0 \rfloor - a+1}^{\lfloor x_0 \rfloor + a} \sum_{j=\lfloor y_0 \rfloor - a+1}^{\lfloor y_0 \rfloor + a} g(i, j)\, L(x_0 - i)\, L(y_0 - j) \tag{2.3}$$

where, $L(\cdot)$ is Lanczos-3 kernel given by (2.2) and $\hat{f}(x_0, y_0)$ denotes the Lanczos-3 interpolated up-sampled image. The Lanczos-3 interpolation in 2-D uses a support region of 6×6=36 pixels from the original low-resolution image $g(x, y)$ to generate the 2-D up-sampled image $\hat{f}(x, y)$ [12].

2.5.2 DCT Interpolation Scheme [74]

The up-sampling process in the DCT domain is performed by padding zero coefficients to the high-frequency part. For this purpose, we need to add N zeros in the high frequency regions, where N is the signal length. Subsequently, type-II $IDCT$ of the extended $2N$ samples is performed to obtain the two-fold up-sampled data. This process is described Fig 2.2. In the case of 2-D video intra frames or image, the twofold up-sampling process in a matrix form can be described as:

$$b^U_{2N\times 2N} = W^T_{2N\times 2N} \times \begin{pmatrix} 2W_{N\times N} b_{N\times N} W^T_{N\times N} & 0 \\ 0 & 0 \end{pmatrix} \times W_{2N\times 2N} \tag{2.4}$$

where, W denotes the 1-D type-II DCT kernel. b and b^U are the down-sized and the up-sampled frame block. 0 denotes a $N \times N$ zero matrix [74].

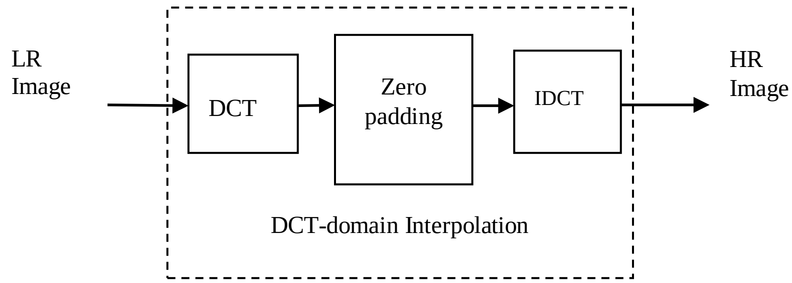


Fig 2.2 DCT interpolation for obtaining HR image from a LR image

2.5.3 New Edge-directed Interpolation (NEDI) [37]

Various 2-D linear interpolation schemes generate blurring artifacts in the up-scaled images due to HF degradation. In the new edge directed interpolation, the covariance based adaptation (CBA) method is used to adjust the predictor which finds the edge pixel values for better reconstruction of an image. This is achieved by finding the geometric duality between the HR covariance and LR covariance which is used to connect the pair of pixels along the same orientation. In NEDI, the estimated HR covariance is used to derive the optimal MMSE interpolation by modeling the image as a locally stationary Gaussian process. The drawback of this method is that the processing time is about two times higher than that of processing time taken in linear interpolation. So the CBA interpolation is used only for the edge pixels. The bilinear interpolation technique is used for the non-edge pixels in the smooth regions due to its simplicity. This hybrid approach is used in NEDI because the edge pixels are very less in comparison to the non-edge pixels and the computation time of this hybrid approach in reconstructing the image will be less than the covariance-based adaptive interpolation for the whole image. The analytical model of NEDI is presented below.

New edge-directed interpolation uses the FIR Wiener filter, equivalently, the linear minimum mean squares error estimator [37] for linear prediction. The 4th order linear estimation model is given by,

$$Y_i = \sum_{k=1}^{4} A_k Y_{i\Delta k} + \varepsilon_i \quad \textit{for } i = 1,2,\ldots,P \tag{2.5}$$

where ε_i is the estimation error, P is the number of data point samples, A_K is the model parameters and each available data point sample (Y_i) has four neighboring data points $Y_{i\Delta k}$. The model parameters A_k can be found by using the least squares estimation:

$$\{\hat{A}_k\} = \underset{\{\hat{A}_k\}}{\arg\min} \sum_{i=1}^{P} \left[Y_i - \sum_{k=1}^{4} A_k . Y_{i\Delta k} \right]^2 \tag{2.6}$$

where the matrix form of (2.6) is given by,

$$\hat{A} = \underset{A}{\arg\min} \|Y - Y_A A\|_2^2 \tag{2.7}$$

and the matrices are defined as

$$Y = \{Y_i\}^T, Y_A = \{Y_{i\Delta k}\}^T, A = \{A_k\}^T \tag{2.8}$$

The sizes of matrices of Y, Y_A and A are $P \times 1$, $P \times 4$ and 4×1, respectively. The close form solution (2.7) is given by,

$$\hat{A} = (Y_A^{\ T}\ Y_A)^{-1}\ Y_A^{\ T} Y \tag{2.9}$$

where, $\hat{A}$ is called as the ordinary least square estimator. Due to the geometry duality, the missing data point X can be interpolated by its four neighboring data points X_K as:

$$X = \sum_{k=1}^{4} \hat{A}_k X_k \tag{2.10}$$

2.5.4 Image Super-resolution based on Interpolation of Wavelet-domain High Frequency Sub-bands [116]

The detail operation of image super-resolution based on interpolation of wavelet-domain high frequency sub-bands scheme is illustrated in the Fig. 2.3. This super-resolution technique makes use of interpolated high frequency sub-bands along with low resolution image to generate a high resolution image counterpart. In this algorithm, discrete wavelet transform (DWT) is used to decompose a low resolution image to different sub-bands such as LL, LH, HL and HH. The sub-bands LH, HL and HH contain high frequency information whereas LL sub-band contains low frequency information of the input image. The up-scaled HF subbands are obtained using bicubic

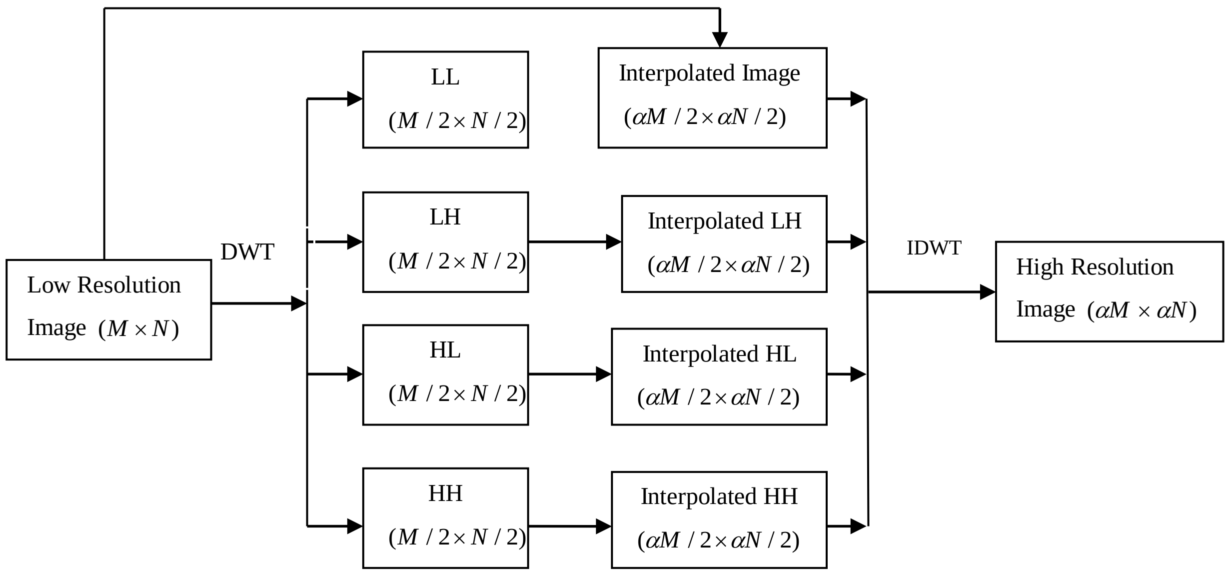


Fig 2.3 Block diagram of Demirel-Anwarjafari super-resolution (DASR) algorithm

interpolation, where α is taken as the interpolation factor. Since the interpolated low resolution image contains more information than LL sub-band, LL sub-band is replaced by low resolution interpolated image in the reconstruction process. Finally, the high resolution image is obtained by taking the inverse discrete wavelet transform of the up-scaled high frequency sub-bands and the interpolated low resolution image. The edges and fine details are preserved in the HR image because of the high frequency enhancement in wavelet domain.

2.5.5 Image Resolution Enhancement using Discrete and Stationary wavelet Decomposition (DSWD) [82]

In this paper, DWT and SWT are employed to preserve the HF information in the super resolved image. The detail block diagrammatic representation of the algorithm is shown in Fig. 2.4. In this algorithm, one level DWT is used to decompose a low resolution input image into different sub-band images. The three high frequency sub-bands: LH, HL and HH contain high frequency information. Down-sampling each of the DWT sub-bands causes information loss. Therefore, SWT is employed to minimize the loss. The interpolated HF sub-bands of DWT and the SWT

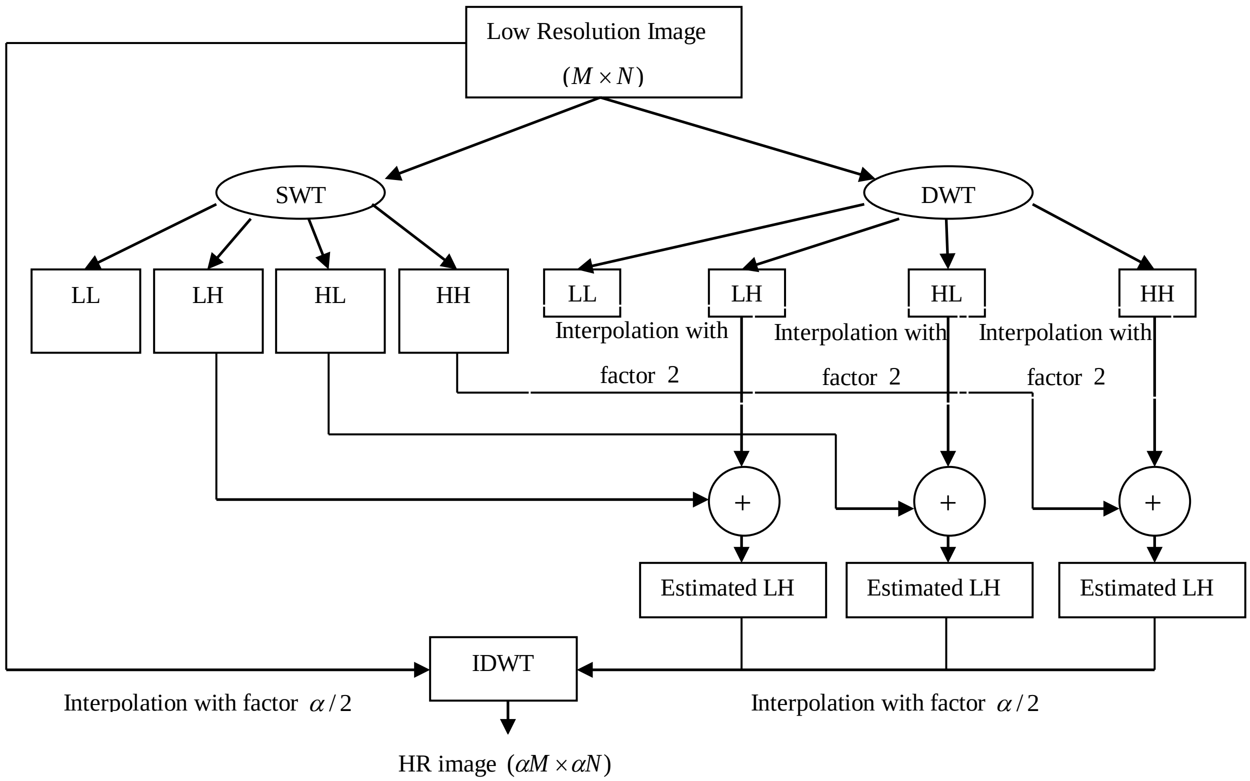


Fig 2.4 Block diagram of image resolution enhancement using DSWD algorithm

high frequency sub-bands have the same size and hence are added correspondingly to generate the estimated HF sub-bands. These HF sub-bands are further interpolated by a factor of $\alpha/2$ meant for further up-conversion in the subsequent process. The LL sub-band is the low resolution version of the original image and contains less information than the original. Therefore, the interpolated version of the original image is used instead of LL sub-band to improve the super-resolution performance. The low resolution image and the DWT high frequency sub-bands are interpolated by $\alpha/2$ and 2 interpolation factor. The estimated HF sub-bands are also interpolated by a factor of $\alpha/2$ so that the dimensions of the estimated sub-bands will be same as that of the dimension of the interpolated input image. Finally, the inverse discrete wavelet transform (IDWT) of all the estimated sub-bands along with the interpolated input image is taken to generate a high resolution image. The edges and fine details of the super resolved image are well preserved because of the correction made by superimposing the SWT high frequency sub-bands with the interpolated DWT sub-bands of the input image.

2.6 Comparative Analysis

To evaluate the HF restoration performance of various interpolation schemes, images of different types are down-sampled at 4:1 compression ratio (CR). Afterwards, the down-sampled images are up-scaled to their original size using various interpolation schemes for comparison with the original images. The down-sampled image is considered as a low-resolution (LR) image while the up-scaled images with various interpolation schemes are considered as a high resolution (HR) image. The computational complexity in terms of CPU execution time of various interpolation algorithms is also computed to determine whether they will be well suited for real-time or off-line applications. In addition, peak-signal-to-noise-ratio (PSNR) in dB is measured to quantify the objective performance of the proposed algorithms. The interpolation algorithms are tested for twenty publicly available 512×512 images for performance analysis.

The execution time comparison of different existing algorithms is given in Table 2.1. Likewise, the objective evaluation at 4:1 and 8:1 compression ratios of various existing algorithms in terms of PSNR (dB) is given in Table 2.2 and Table 2.3. The best performance is shown in bold for clear presentation. In addition, the subjective performance comparisons of these algorithms are given in Fig. 2.5, Fig. 2.6 and Fig. 2.7.

The various existing techniques used for quantitative analysis are: Nearest-neighbor interpolation [3], Bilinear interpolation [5], Bicubic interpolation [4], Lanzos-2 interpolation [12], Lanczos-3 interpolation [12], DCT-based interpolation [74], Demirel-Anbarjafari super-resolution [116], Image resolution enhancement using discrete and stationary wavelet decomposition [82].

It may be observed from Table 2.2 and Table 2.3 that DCT and Lanczos-3 interpolation schemes show better objective performance at 4:1 and 8:1 compression ratios than all other existing schemes. However, DCT is the winner in most of the images though it has undesirable ringing artifacts. In contrast, Lanczos-3 interpolation shows better subjective performance than other existing schemes because of less interpolation artifacts such as blurring and ringing as illustrated in Fig. 2.5, Fig. 2.6 and Fig. 2.7. In addition, Lanczos-3, being a spatial domain polynomial based interpolation scheme, takes less computation time than transform-domain DCT interpolation as shown in Table 2.1. Hence, Lanczos-3 is preferable because of reduced computational complexity and less interpolation artifacts. Since the proposed post-processing and composite algorithms are based on enhancing the HF contents of an interpolated image, the interpolation artifacts will be further enhanced resulting in degradation in image quality. Therefore, it is preferable to use Lanczos-3 interpolation for further improvement.

Furthermore, both the DCT and Lanczos-3 interpolation schemes generate blurring effects at the edge and fast changing regions in the up-scaled images. Therefore, there is sufficient scope for further improvement of these existing algorithms. Hence, efforts are made to improve the performance of existing interpolation algorithms so that blurring can be significantly reduced with a better visual quality. Therefore, Lanczos-3 interpolation scheme is taken as the basic interpolation paradigm, which is further improvised in various pre-processing and post-processing algorithms developed and presented in this research work.

The most commonly found interpolation artifacts in the existing schemes are blurring, ringing and aliasing. The polynomial based interpolation schemes such as bilinear, bicubic and Lanczos suffers from blurring artifacts. These interpolation schemes are based on convolution in which the pixel intensity at a given location is estimated as a linear combination of the neighboring pixels with weights inversely proportional to their distance from the estimating location. The weighted combination of neighboring pixels simply represents a low- pass filtering operation which blurs the fine details and edges of an image due to high frequency attenuation as shown in Fig. 2.5, Fig. 2.6 and Fig. 2.7. In addition, the transform-domain up-sampling schemes such as DCT and wavelet also show blurring due to high frequency degradation.

Table 2.1 Execution time of the existing algorithms at 4:1 CR

Images of different size (M×N)	**Execution time in Seconds for different interpolation schemes**							
	Nearest [3]	**Bilinear [5]**	**Bicubic [4]**	**Lanczos-2 [12]**	**Lanczos3 [12]**	**DCT [74]**	**DASR [116]**	**DSWD [82]**
Clock (200×200)	0.0161	0.0141	0.0150	**0.0054**	0.0151	0.0518	0.2532	1.0770
Lena (256×256)	0.0163	0.0147	0.0152	**0.0062**	0.0157	0.0633	0.2655	1.1235
Fruit (377×321)	0.0164	0.0153	0.0163	**0.0063**	0.0168	0.1729	0.2758	1.1628
Lena (512×512)	0.0174	0.0201	0.0215	**0.0114**	0.0222	0.1673	0.3019	1.2627
Pentagon (1024×1024)	**0.0208**	0.0354	0.0409	0.0295	0.0442	0.6372	0.4953	1.7218

Table 2.2 PSNR (dB) comparison of different existing interpolation schemes at 4:1 compression ratio for various (512×512) images

Image	**Nearest [3]**	**Bilinear [5]**	**Bicubic [4]**	**Lanczos2 [12]**	**Lanczos3 [12]**	**DCT [74]**	**DASR [116]**	**DSWD [82]**
Lena	31.424	32.704	34.148	34.207	34.813	**35.022**	31.220	31.345
Boat	28.388	28.940	29.951	29.995	30.375	**30.466**	27.865	28.446
Livingroom	28.132	28.617	29.557	29.608	29.977	**30.128**	27.459	28.174
Fingerprint	25.374	28.045	30.632	30.753	31.722	**32.133**	26.378	26.411
Goldhill	30.114	30.574	31.405	31.440	**31.725**	31.716	29.609	29.829
Pirate	29.398	30.027	31.058	31.101	31.490	**31.607**	28.955	29.391
Baboon	32.315	33.588	35.014	35.075	35.662	**35.889**	32.145	32.373
Barbara	25.096	24.925	25.352	25.367	**25.428**	25.183	24.764	24.837
Bridge	25.473	25.728	26.504	26.541	26.826	**26.918**	24.910	25.348
Cat	30.441	30.949	31.982	32.024	32.427	**32.562**	29.992	30.464
Crowd	29.522	30.984	32.667	32.732	33.451	**33.768**	29.277	29.582
Cycle	21.227	21.208	21.895	21.926	**22.154**	22.129	20.693	21.173
F16	29.035	30.379	31.543	31.633	32.104	**32.722**	29.689	29.830
House	28.816	29.248	30.314	30.371	30.807	**30.862**	28.049	28.790
Lake	27.881	28.945	30.022	30.080	30.495	**30.793**	27.997	28.549
Cameraman	31.044	33.214	35.757	35.884	37.216	**37.832**	30.972	31.854
Elaine	31.627	32.534	33.117	33.131	**33.309**	33.284	31.617	31.519
Mandrill	23.122	23.045	23.630	23.663	23.859	**23.925**	22.726	23.025
Peppers	29.876	31.180	31.991	32.045	32.329	**32.747**	30.451	30.795
Ruler	12.573	12.335	12.613	12.626	12.673	12.600	14.458	**15.638**

Table 2.3 PSNR (dB) comparison of different existing interpolation schemes at 8:1 compression ratio for various (512×512) images

Image	**Nearest [3]**	**Bilinear [5]**	**Bicubic [4]**	**Lanczos2 [12]**	**Lanczos3 [12]**	**DCT [74]**	**DASR [116]**	**DSWD [82]**
Lena	28.582	30.288	31.267	31.289	31.716	**31.876**	28.408	28.521
Boat	25.981	26.899	27.566	27.585	27.851	**27.905**	25.517	26.049
Livingroom	25.822	26.634	27.202	27.216	27.429	**27.438**	25.215	25.871
Fingerprint	21.948	24.528	26.641	26.707	27.573	**27.757**	22.818	22.847
Goldhill	27.857	28.801	29.368	29.382	29.621	**29.694**	27.390	27.593
Pirate	26.921	28.062	28.745	28.764	29.043	**29.118**	26.515	26.915
Baboon	29.485	31.109	32.122	32.150	32.606	**32.751**	29.329	29.537
Barbara	23.534	24.033	24.261	24.265	24.345	**24.365**	23.231	23.299
Bridge	23.449	24.112	24.593	24.606	24.790	**24.823**	22.937	23.341
Cat	28.103	29.011	29.681	29.696	29.972	**30.035**	27.693	28.129
Crowd	26.406	28.251	29.348	29.376	29.861	**30.002**	26.187	26.460
Cycle	19.338	19.772	20.155	20.165	20.324	**20.369**	18.863	19.301
F16	26.414	27.875	28.708	28.744	29.102	**29.407**	27.014	27.143
House	26.379	27.188	27.778	27.795	**27.989**	27.961	25.686	26.364
Lake	25.380	26.779	27.576	27.598	27.940	**28.097**	25.498	26.001
Cameraman	27.554	29.587	30.999	31.042	31.796	**32.236**	27.506	28.289
Elaine	29.633	31.315	31.908	31.918	32.128	**32.212**	29.632	29.540
Mandrill	21.534	21.786	22.083	22.093	22.198	**22.221**	21.160	21.438
Peppers	27.431	29.203	29.892	29.919	30.184	**30.453**	27.962	28.278
Ruler	11.409	11.484	11.595	11.599	11.624	11.393	13.120	**14.191**

Table 2.4 Computational complexity of existing algorithms for an $N \times N$ image

Algorithm	Number of multiplications	Number of additions
Nearest-neighbor	0	$2N^2$
Bilinear	$4N^2$	$3N^2$
Bicubic	$16N^2$	$15N^2$
Lanczos-3	$36N^2$	$35N^2$
DCT	$5N^2 \log_2 N + 4N^2$	$15N^2 \log_2 N + 2N^2 + 6N$
DASR	$106N^2$	$105N^2$
DSWD	$124N^2$	$126N^2$

Table 2.5 Operation counts of the existing algorithms

Image size $N \times N$	Number of Operations	Nearest [3]	Bilinear [5]	Bicubic [4]	Lanczos-3 [12]	DCT [74]	DASR [116]	DSWD [82]
128×128	Multiplications	0	65356	262144	589824	638976	1736704	2031616
	Additions	32768	49152	245760	573440	1753856	1720320	2064384
256×256	Multiplications	0	262144	1048576	2359296	2883584	6946816	8126464
	Additions	131072	196608	983040	2293760	7996928	6881280	8257536
512×512	Multiplications	0	1048576	4194304	9437184	12845056	27787264	32505856
	Additions	524288	786432	3932160	9175040	35916800	27525120	33030144

(a) (b) (c)
(d) (e) (f)
(g) (h) (i)

Fig. 2.5 Subjective evaluation of Lena (256×256) image using various up-sampling schemes at 4:1 compression ratio: (a) Original; (b) Nearest; (c) Bilinear; (d) Bicubic; (e) Lanczos-2; (f) Lanczos-3; (g) DCT ; (h) DASR; (i) DSWD

Fig. 2.6 Subjective evaluation of Barbara (512×512) image using various up-sampling schemes at 4:1 compression ratio: (a) Original; (b) Nearest; (c) Bilinear; (d) Bicubic; (e) Lanczos-2; (f) Lanczos-3; (g) DCT ; (h) DASR; (i) DSWD

Fig. 2.7 Subjective evaluation of the selected green rectangular region (237×222) of Barbara (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Nearest; (c) Bilinear; (d) Bicubic; (e) Lanczos-2; (f) Lanczos-3; (g) DCT ; (h) DASR; (i) DSWD

Ringing artifacts appear as spurious signals near sharp transitions in an image. Ringing artifacts are more prominent in DCT interpolation and may be observed in Fig 2.7. The main cause of ringing is due to the band limiting of an image in frequency domain by padding zero coefficients to the high frequency side or truncating image coefficients in frequency domain. The DCT domain image resizing methods make use of a truncation that discards high frequency coefficients to down-scale and zero padding to up-scale an image. However, these truncation and zero padding in frequency domain generate ringing artifacts near object boundaries in spatial domain.

Aliasing is a very common phenomenon in various interpolation schemes. The main problem with spatial aliasing in images is the introduction of artifacts such as jaggedness in line features, spurious highlights and the appearance of frequency patterns not present in the original image which is otherwise known as Moiré pattern. The deformation in the original frequency pattern due to aliasing may be observed in Barbara image as depicted in Fig. 2.7. Spatial aliasing is due to under-sampling. For instance, a continuous 2-D function of two continuous variables can be band limited only if it extends infinitely in both coordinate directions. The very act of spatially limiting the 2-D function introduces frequency components extending to infinity in frequency domain. Since we cannot sample a function infinitely, aliasing is present in various up-sampled images as shown in Fig. 2.7. The effect of aliasing can be reduced by band limiting or slightly blurring an image to be up-sampled so that high frequencies are attenuated.

It may be observed from Table 2.2 and Table 2.3 that the PSNR is low for some images such as Barbara, Cycle, Mandrill and Ruler which are rich in HF pattern. The reduction in PSNR is because of the deformation of HF pattern due to aliasing or under-sampling. The deformation of this high frequency pattern generates a new pattern which was absent in the original image and is the cause of reduction in PSNR. The subjective quality degradation of Barbara (512×512) image may also be observed in Fig. 2.7 due to aliasing.

Computational complexity is an important parameter to evaluate an algorithm in terms of its applicability for real-time applications. Complexity of an algorithm is related to the number of multiplications and additions involved to obtain the final output. The number of operations required for a pixel interpolation is a widely accepted method. This number includes the operations required for convolution of the basis function. All the polynomial interpolation schemes are based on convolution. The differences in execution time are due to the difference in size of the convolution kernel. The convolution of $P \times P$ kernel needs P^2 multiplications and $P^2 - 1$ additions [129]. In case of nearest-neighbor interpolation, the number of additions and multiplication required for a pixel estimation is 2 and zero. Therefore, it is the fastest among all the interpolation schemes. Bilinear interpolation operates on 2×2 neighborhood and so requires 4 multiplications and 3 additions per pixel. Likewise, bicubic interpolation requires more number of operations than bilinear interpolation because the pixel estimation is performed on a larger (4×4) neighborhood which requires 16 multiplications and 15 additions. Therefore, bicubic interpolation takes more computation time than bilinear interpolation as shown in Table 2.1. In addition, Lanczos-3 interpolation has a kernel size of 6×6 and hence requires 36 multiplications and 35 additions and takes more computational time than bicubic interpolation.

In case of DCT, there are $N^2 \log_2 N$ multiplications and $3N^2 \log_2 N - 2N^2 + 2N$ additions for N points [130]. Inverse DCT has same number of operations as that of the forward DCT. For the purpose of up-sampling, the number of points becomes $2N$ because of zero padding in frequency domain. The up-sampled image in spatial domain is obtained by taking the IDCT of the zero padded coefficients. Therefore, the total number of multiplications and additions required for DCT up-sampling scheme become $5N^2 \log_2 N + 4N^2$ and $15N^2 \log_2 N + 2N^2 + 6N$ as shown in Table 2.4. Thus, DCT takes more execution time than Lanczos-3 interpolation for images of higher dimensions as depicted in Table 2.1. DCT is computationally more complex than polynomial based interpolation schemes because of the requirement of forward transform and inverse transform during the up-sampling process.

In case of DWT, the number of multiplications and additions is $2KN^2$ for an $N \times N$ image where K is the number nonzero filter coefficients [131]. Considering the number of nonzero filter coefficients to be 9 for a 3×3 filter mask, number of multiplications and additions becomes $18N^2$ for forward DWT. During up-sampling, N becomes $2N$ and therefore, the number of additions and multiplications becomes $72N^2$ during inverse DWT. Up-sampling N samples to $2N$ employs bicubic interpolation which requires $16N^2$ multiplications and $15N^2$ additions. Hence, in case of DASR algorithm proposed by G. Anwarjafari et al. [116] the total number of multiplications and additions becomes $106N^2$ and $105N^2$ due to the employment of DWT. Likewise, the number of multiplications and additions in DSWD algorithm proposed by H. Damirel et al. [82] become $124N^2$ and $126N^2$. Since wavelet is a transform domain interpolation scheme, it consumes more computation time than polynomial based interpolation schemes. The forward discrete wavelet transform decomposes an image to LL, LH, HL and HH components. These components are individually up-scaled to higher dimensions followed by inverse discrete wavelet transform to obtain the up-scaled image in spatial domain. Therefore, wavelet requires more number of operations and high computation time than polynomial interpolation schemes and DCT as depicted in Table 2.1and Table 2.5.

2.7 Conclusion

An elaborate study of different interpolation schemes has been carried out. The existing interpolation schemes are classified into polynomial-based, edge-directed and transform-domain interpolation schemes. The advantages and limitations of these existing schemes in perspective of online and offline applications are discussed. The techniques used for experimental results are lucidly explained. A comparative study of various state-of-the-art interpolation schemes has been presented. The quantitative analysis of various state-of-the-art interpolation schemes is accomplished using accuracy metrics over different publicly available images.

It is observed that Lanczos-3 interpolation scheme is quite an efficient algorithm and hence is taken as the basic interpolation paradigm, on which further improvements are proposed. The developed algorithms are presented in next four contributing chapters.

In the next chapter, various spatial-domain pre-processing techniques based on higher order derivatives are presented for better 2-D up-scaling.

Chapter 3

Spatial-domain Pre-processing Algorithms using Higher Order Derivatives

Preview

Interpolation plays an important role in many 2-D up-sampling applications. Most of the existing interpolation techniques such as nearest-neighbour, bilinear, bicubic, lanczos-3 and DCT produce non-uniform blurring while producing an up-scaled image from a low-resolution image data. The blurring is significant at the edges and fast changing regions and remains low in the slowly varying, flat regions as illustrated in Fig. 2.5, Fig. 2.6 and Fig. 2.7 of the previous chapter. This chapter presents three spatial domain pre-processing algorithms which are based on 4^{th}, 6^{th} and 8^{th} order derivatives to tackle the non-uniform blurring. These algorithms are used to obtain the high frequency (HF) extracts from an image and perform precise sharpening on a low resolution image to alleviate the blurring in its 2-D up-sampled counterpart. So, they are based on inverse modeling approach of high frequency degradation. In this chapter, the term *pre-processing* refers to processing of the low-resolution input image prior to image up-scaling. An attempt has been made here to develop three novel pre-processing schemes for this purpose. This chapter critically compares the capabilities and limitations of these pre-processing algorithms and their relevance in the perspective of adaptability to the varying conditions, computational complexities and visual quality. The simulation results, presented at the end of the chapter, are quite encouraging. The organisation of this chapter is given below.

- *Laplacian of Laplacian based Global Pre-processing (LLGP) Scheme*
- *Hybrid Global Pre-processing (HGP) Scheme*
- *Iterative Laplacian of Laplacian based Global Pre-processing (ILLGP) Scheme*
- *Experiment and Simulation*
- *Results and Discussion*
- *Conclusion*

3.1 Laplacian of Laplacian based Global Pre-processing (LLGP) Scheme [P1]

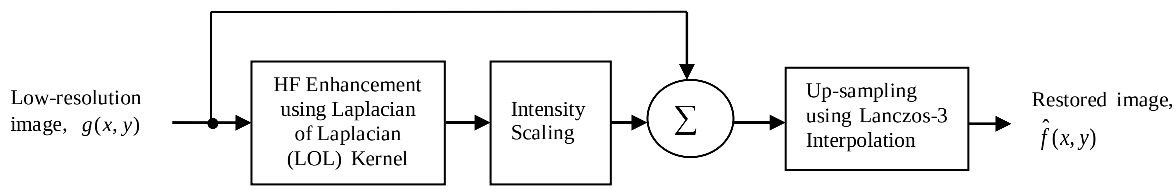


Fig. 3.1a Block Diagram of Laplacian of Laplacian based Global Pre-processing Scheme

The proposed LLGP algorithm is a global pre-processing scheme, used for precise sharpening of a low resolution image to alleviate the blurring in its up-scaled counterpart. In this case, the low resolution image is sharpened using a newly developed 5×5 Laplacian of Laplacian (LOL) kernel which is based on 4^{th} order derivative. This kernel operates on a larger neighborhood (5×5) to extract very high frequency information and works much better than Laplacian kernel to counter blurring more effectively.

The proposed LLGP scheme is typically a no reference, high frequency predictive scheme that predicts and superimposes the HF extracts upon the low resolution image. Since the up-sampling process gives an effect of a low pass filtering (LPF) operation, the high frequency degradation is much more than the medium and low frequency. Hence, the superimposition of the predicted high frequency components prior to Lanczos-3 up-sampling would reduce the extent of blurring more effectively than Laplacian operator. The proposed LLGP scheme is illustrated in Fig. 3.1. The proposed scheme is a global pre-processing scheme and operates on a low resolution image and hence is faster and efficient scheme.

In case of an image, slowly varying and flat regions correspond to low frequencies whereas the fast varying regions with more variations and details correspond to high frequencies. However, the high frequency can be further classified into high frequency (HF) and very high frequency (VHF) sub-bands as illustrated in Fig. 3.1b. Typically, the very high frequency corresponds to the very fine or subtler details of an image and can be considered as VHF sub-band. Generally, the degradation of VHF sub-band is more than the HF sub-band during the up-s

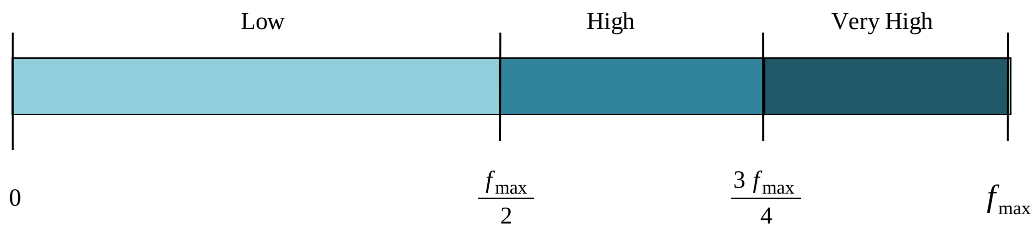


Fig. 3.1b Quantitative explanation to high frequency and very high frequency components

-ampling process which is analogous to an LPF operation. Hence, HF restoration would be more effective if very high frequency details of an image can be restored by making use of higher order derivative operators. The design of one of the higher order derivative operator, Laplacian of Laplacian which is based on 4th order derivative operator is given in the next section.

3.1.1 Development of 5×5 LOL filter kernel

In LLGP scheme, a 5×5 Laplacian of Laplacian (LOL) kernel is developed using 2-D 4th order derivative meant for image sharpening. This approach primarily consists of defining a discrete formulation of the 4th order derivative and then developing a 5×5 filter kernel based on that formulation. In our method, the partial 4th order derivative is taken in x or y direction in view of the fact that it generates five coefficients which perfectly match to the size of a 5×5 filter kernel.

It would be preferable to go for 2nd or 4th order derivatives rather than 3rd or 5th order derivatives. It is because, the even order derivatives such as 2nd or 4th order produce an odd number of coefficients such as 3 or 5 respectively. These odd numbers of coefficients would be suitable for developing an isotropic filter kernel of size 3×3 or 5×5, respectively for spatial domain filtering. In our application, the 5×5 filter kernel is taken since it provides adequate amount of sharpening with moderate computational complexity to alleviate the blurring more effectively than the 3×3 Laplacian kernel while up-sampling an image. The development of 5×5 LOL kernel is based on the 4th order derivative in $x-$, and $y-$ directions. The 4th order derivative operator for a 2-D signal such as a low resolution image $g(x, y)$ is given by,

$$\nabla^4 g(x,y) = \frac{\partial^4 g}{\partial x^4} + \frac{\partial^4 g}{\partial y^4} \tag{3.1}$$

As per the basic definition, the 1st order derivative of a 1-D function $g(x)$ is given by,

$$\frac{\partial g}{\partial x} = g(x+1) - g(x) \tag{3.2}$$

Similarly, the 2nd order derivative of the 1-D function $g(x)$ is given by,

$$\frac{\partial^2 g}{\partial x^2} = g(x+1) - 2g(x) + g(x-1) \tag{3.3}$$

Now, the 4th order derivative of $g(x)$ is obtained by,

$$\frac{\partial^4 g}{\partial x^4} = \frac{\partial^2}{\partial x^2}\left[\frac{\partial^2 g}{\partial x^2}\right]$$

$$= \frac{\partial^2}{\partial x^2}\left[g(x+1) - 2g(x) + g(x-1)\right]$$

$$= \frac{\partial^2}{\partial x^2}\left[g(x+1)\right] - 2\frac{\partial^2}{\partial x^2}\left[g(x)\right] + \frac{\partial}{\partial x^2}\left[g(x-1)\right]$$

$$= g(x+2) - 2g(x+1) + g(x) - 2g(x+1) + 4g(x) - 2g(x-1) + g(x) - 2g(x-1) + g(x-2)$$

$$= g(x+2) - 4g(x+1) + 6g(x) - 4g(x-1) + g(x-2)$$

$$\therefore \quad \frac{\partial^4 g}{\partial x^4} = g(x+2) - 4g(x+1) + 6g(x) - 4g(x-1) + g(x-2) \tag{3.4}$$

Similarly, we have the 4th order derivative in $y-$direction:

$$\frac{\partial^4 g}{\partial y^4} = g(y+2) - 4g(y+1) + 6g(y) - 4g(y-1) + g(y-2) \tag{3.5}$$

In order to represent (3.1) in 2-D discrete form, (3.4) and (3.5) can be extended to 2-D as follows,

$$\frac{\partial^4 g}{\partial x^4} = g(x+2,y) - 4g(x+1,y) + 6g(x,y) - 4g(x-1,y) + g(x-2,y) \tag{3.6}$$

$$\frac{\partial^4 g}{\partial y^4} = g(x,y+2) - 4g(x,y+1) + 6g(x,y) - 4g(x,y-1) + g(x,y-2) \tag{3.7}$$

Now, substituting (3.6) and (3.7) into (3.1), we have

$$\begin{aligned}\nabla^4 g(x,y) = g(x+2,y) - 4g(x+1,y) + 12g(x,y) - 4g(x-1,y) + g(x-2,y) \\ + g(x,y+2) - 4g(x,y+1) - 4g(x,y-1) + g(x,y-2)\end{aligned} \tag{3.8}$$

The above equation can be implemented using a 5×5 filter mask by assigning the coefficients of

x, y into the corresponding location of the Laplacian of Laplacian (LOL) filter kernel. The remaining locations of the filter kernel are assigned zero. Thus, the 5×5 LOL filter kernel, $h_{LOL}(x, y)$ is given by,

$$h_{LOL}(x,y) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -4 & 0 & 0 \\ 1 & -4 & 12 & -4 & 1 \\ 0 & 0 & -4 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.9}$$

The above equation represents the 5×5 filter kernel based on 4th order derivative for image sharpening. Since the derivatives of any order are linear operation, the designed filter mask is a linear operator. The proposed filter mask is rotation invariant and hence gives isotropic result for rotations in increments of 90°. In addition, the sum of all the weights of the filter kernel is zero like Laplacian and hence is intended for sharpening operation.

To evaluate the HF restoration performance of the proposed algorithm, different images or video sequences are purposefully down-sampled at 4:1 compression ratio (CR) to generate the sub-sampled image or a video sequence. Later on, the down-sampled images are sharpened and re-scaled to their original size using the proposed algorithm for comparison with the original images.

The compression ratio mentioned in this algorithm refers to spatial sub-sampling of the original image, a degradation process through which the original signal undergoes typically just before transmission/storage applications prior to the restoration process presented by our algorithm. For example, if the signal has been down-sampled by a compression ratio of 4:1, then it is up-sampled in the second phase of our algorithm, by a factor of 1:4 employing Lanczos-3 interpolation scheme, preceded by sharpening using Laplacian of Laplacian operator to counteract the blurring effect of the interpolation. In general, up-sampling by a factor of 1:N is required here for an image that has spatially been down-sampled by a factor N:1.

The proposed algorithm: LLGP is presented in the next section.

3.1.2 LLGP algorithm

Let $g(x, y)$ denote a sub-sampled image or video frame. Let $\nabla^4 g(x, y)$ be the filtered version of the sub-sampled image using LOL filter kernel. The sharpened version of the sub-sampled image or video frame is denoted by $g_s(x, y)$. Let them be of size $(P \times Q)$. The original image is of size $(2P \times 2Q)$. The LLGP algorithm is given below.

Step-1. Select a 5×5 window, w in the sub-sampled image $g(x, y)$.

$w_{s,t}(x, y), \quad -2 \le s, t \le 2$

Step-2. Obtain $\nabla^4 g(x, y)$ by linearly convolving $h_{LOL}(x, y)$ with $g(x, y)$.

$$\nabla^4 g(x, y) = \sum_{s=-2}^{2}\sum_{t=-2}^{2} h_{LOL}(s,t)\ g(x+s, y+t) \qquad (3.10)$$

Step-3. Repeat Step-1 and Step-2 for all the (x, y) locations of $g(x, y)$ to obtain the filtered image, $\nabla^4 g(x, y)$.

Step-4. The weighted version of the filtered output $\nabla^4 g(x, y)$ is added to the original sub-sampled image to generate the sharpened image, $g_s(x, y)$.

$$g_s(x, y) = g(x, y) + K[\nabla^4 g(x, y)] \qquad (3.11)$$

where, $K = 0.0312$ for 4:1 compression ratio

Step-5 The sharpened image, $g_s(x, y)$ is finally up-sampled by Lanczos-3 interpolation using (2.3) to obtain the up-scaled image, $\hat{f}(x, y)$.

Step-6. For a video sequence, repeat Step-1 to Step-5 for all the frames to obtain the up-sampled, restored sequence, $\hat{f}(x, y, n)$. The term n is the frame number that represents discrete time.

The expression for Lanczos-3 interpolation is given in (2.3). The estimation of the weight factor, K for LLGP algorithm is illustrated in Section 3.4.1.

3.2 Hybrid Global Pre-processing (HGP-I) Scheme [P2]

The proposed HGP-I algorithm is a novel hybrid, global pre-processing scheme which sharpens the sub-sampled image more effectively by performing two global pre-processing operations in succession. The former is the high frequency (HF) enhancement using 5×5 Laplacian of Laplacian (LOL) kernel. The later is HF enhancement using 3×3 Laplacian kernel and so the algorithm is based on 6th order derivative. Furthermore, it is by using the kernels of different size in succession, the high frequency information corresponding to 5×5 and 3×3 neighborhood are enhanced simultaneously. The hybridization of both the schemes results in more accurate prediction of high frequency contents that result in much improved performance at the cost of a little additional computational complexity.

Since the degradation of high frequency information is much more in comparison to the medium and low frequency, superimposing the weighted version of HF extracts with the sub-sampled image alleviates the blurring considerably in the up-sampled image or in a video frame. The proposed scheme is computationally less complex since it makes use of two global processing schemes that operates on sub-sampled images. The HGP algorithm using LOL and Laplacian operator is illustrated in Fig. 3.2.

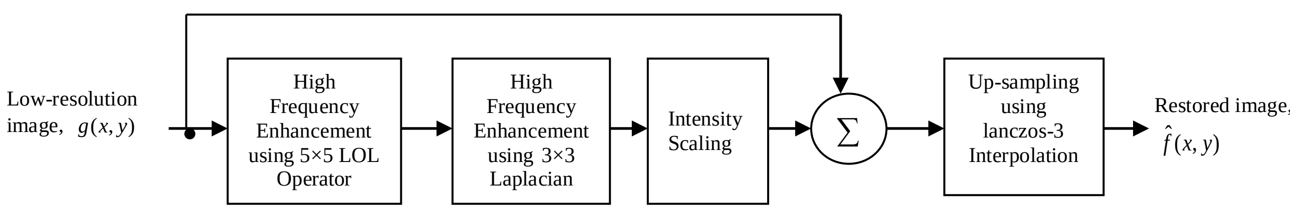


Fig. 3.2 Block Diagram of Hybrid Global Pre-processing (HGP-I) Scheme

Furthermore, since very fine details of an image that are much prone to degradation during an up-scaling process, can be better enhanced using higher order derivatives than the 1st and 2nd order derivative operators. As per the experimental evidences given in Fig. 3.4, higher order derivative operators are better in HF restoration than the lower order counterpart. Therefore, by using 4th order and 2nd order derivative operators in succession, 6th order derivative operator can be realized which is meant for better HF enhancement. In addition, the objective of this HGP-I scheme to show its performance improvement over standalone schemes (LOL and Laplacian) through their hybridization.

3.2.1 HGP-I algorithm

The HGP-I algorithm is almost same as LLGP algorithm with the following modifications. Let $g(x, y)$ be the original sub-sampled intra-frame. Let $\nabla^4 g(x, y)$ be the output using Laplacian of

Laplacian (LOL) as per LLGP algorithm. $\nabla^6 g(x,y)$ denotes the output which is obtained by linearly convolving $\nabla^4 g(x,y)$ with Laplacian kernel, $h_{La}(x,y)$ and is given by,

$$\nabla^6 g(x,y) = \sum_{s=-1}^{1} \sum_{t=-1}^{1} h_{La}(s,t)\, \nabla^4 g(x+s, y+t) \tag{3.12}$$

where,

$$h_{La}(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{3.13}$$

and s and t are dummy variables. $g_s(x,y)$ denotes the sharpened intra-frame using HGP algorithm and is obtained by adding the weighted version of $\nabla^6 g(x,y)$ with the original sub-sampled intra-frame and is given by,

$$g_s(x,y) = g(x,y) + K[\nabla^6 g(x,y)] \tag{3.14}$$

and the weight factor, $K = 0.0072$ for 4:1 compression ratio.

The remaining part of the algorithm is same as that of LLGP algorithm. The estimation of the weight factor, K for HGP-I algorithm is illustrated in Section 3.4.1.

Furthermore, the Laplacian mask and Laplacian of Laplacian mask can be combined into a single 7×7 mask which is based on 6th order derivative operator. The design of such a mask is given in Section 3.2.2.

3.2.2 Development of High Pass Filter Kernel based on 6th Order Derivative (HGP-II)

In case of HGP-I scheme, there is a requirement of two end-to-end filtering operations to obtain 6th order derivative using 5×5 Laplacian of Laplacian and 3×3 Laplacian operator. However, these two successive operations can be replaced by one single filtering operation by developing a single 7×7 filter kernel which is based on 6th order derivative. The development of 7×7 filter kernel is based on the 6th order derivative in $x-$, and $y-$directions. The 6th order derivative operator for a 2-D signal $g(x,y)$ is given by:

$$\nabla^6 g(x,y) = \frac{\partial^6 g}{\partial x^6} + \frac{\partial^6 g}{\partial y^6} \tag{3.15}$$

As per the basic definition, the 2nd order derivative of a 1-D function $g(x)$ is given by,

$$\frac{\partial^2 g}{\partial x^2} = g(x+1) - 2g(x) + g(x-1)$$

Similarly, the 4nd order derivative of the 1-D function $g(x)$ according to (3.4) is given by,

$$\frac{\partial^4 g}{\partial x^4} = g(x+2) - 4g(x+1) + 6g(x) - 4g(x-1) + g(x-2)$$

Now the 6th order derivative of the 1-D function is given by,

$$\begin{aligned}
\frac{\partial^6 g}{\partial x^6} &= \frac{\partial}{\partial x^2}\left[\frac{\partial^4 g}{\partial x^4}\right] \\
&= \frac{\partial}{\partial x^2}\left[g(x+2) - 4g(x+1) + 6g(x) - 4g(x-1) + g(x-2)\right] \\
&= \frac{\partial}{\partial x^2}[g(x+2)] - 4\frac{\partial}{\partial x^2}[g(x+1)] + 6\frac{\partial}{\partial x^2}[g(x)] - 4\frac{\partial}{\partial x^2}[g(x-1)] + \frac{\partial}{\partial x^2}[g(x-2)] \\
&= g(x+3) - 2g(x+2) + g(x+1) - 4[g(x+2) - 2g(x+1) + g(x)] + 6[g(x+1) - 2g(x) + g(x-1)] \\
&\quad - 4[g(x) - 2g(x-1) + g(x-2)] + g(x-1) - 2g(x-2) + g(x-3)
\end{aligned}$$

$$\therefore \quad \frac{\partial^6 g}{\partial x^6} = g(x+3) - 6g(x+2) + 15g(x+1) - 20g(x) + 15g(x-1) - 6g(x-2) + g(x-3)$$

Similarly, we have the 6th order derivative in $y-$direction:

$$\frac{\partial^6 g}{\partial y^6} = g(y+3) - 6g(y+2) + 15g(y+1) - 20g(y) + 15g(y-1) - 6g(y-2) + g(y-3)$$

In order to represent (3.15) in 2-D discrete form, the above two equations can be extended to 2-D as follows,

$$\frac{\partial^6 g}{\partial x^6} = g(x+3,y) - 6g(x+2,y) + 15g(x+1,y) - 20g(x,y) + 15g(x-1,y) - 6g(x-2,y) + g(x-3,y)$$

… (3.16)

$$\frac{\partial g}{\partial y^6} = g(x,y+3) - 6g(x,y+2) + 15g(x,y+1) - 20g(x,y) + 15g(x,y-1) - 6g(x,y-2) + g(x,y-3)$$

…. (3.17)

Substituting (3.16) and (3.17) into (3.15), we have

$$\begin{aligned}
\nabla^6 g(x,y) = {} & g(x+3,y) - 6g(x+2,y) + 15g(x+1,y) - 40g(x,y) + 15g(x-1,y) - 6g(x-2,y) + \\
& g(x-3,y) + g(x,y+3) - 6g(x,y+2) + 15g(x,y+1) + 15g(x,y-1) - 6g(x,y-2) + g(x,y-3)
\end{aligned}$$

….. (3.18)

The above equation can be implemented using a 7×7 filter mask by assigning the coefficients of x, y into the corresponding location of the filter kernel. The remaining locations of the filter kernel are assigned zero. Thus, the 7×7 filter kernel, $h_{LLL}(x,y)$, which is based on 6th order derivative, is given by:

$$h_{LLL}(x,y)=\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 15 & 0 & 0 & 0 \\ 1 & -6 & 15 & -40 & 15 & -6 & 1 \\ 0 & 0 & 0 & 15 & 0 & 0 & 0 \\ 0 & 0 & 0 & -6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.19}$$

The above equation represents the 7×7 filter kernel based on 6th order derivative meant for image sharpening. Since the derivatives of any order are linear operation, the designed filter mask is a linear operator. The sum of all the weights of the filter kernel is zero like Laplacian and hence is a necessity for a derivative operation.

The HF extract $\nabla^6 g(x,y)$ employing 6th order derivative operator, $h_{LLL}(x,y)$ is given by,

$$\nabla^6 g(x,y)=\sum_{s=-3}^{3}\sum_{t=-3}^{3} h_{LLL}(s,t)g(x+s,y+t) \tag{3.20}$$

where, the terms have their usual meaning. The sharpened low resolution image, $g_s(x,y)$ is obtained by superimposing the HF extract to original low resolution image, $g(x,y)$ and is given by:

$$g_s(x,y)=g(x,y)-K\nabla^6 g(x,y) \tag{3.21}$$

The sharpened image is finally up-sampled using Lanczos-3 interpolation to generate HR image $\hat{f}(x,y)$.

An iterative LLGP algorithm is presented in the next section for further improvement.

3.3 Iterative Laplacian of Laplacian based Global Pre-processing (ILLGP) Scheme [P3]

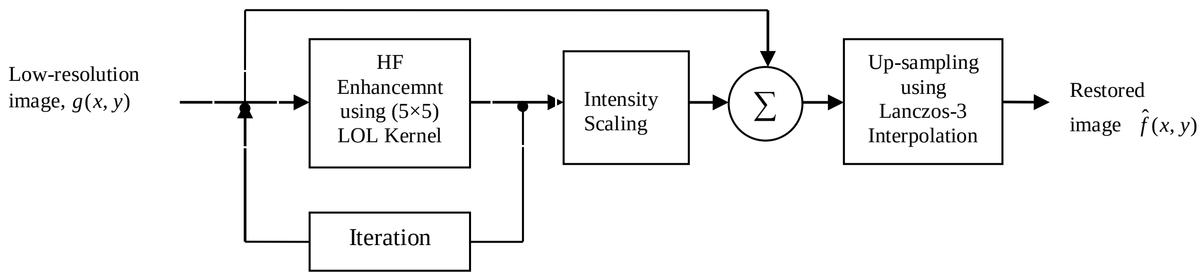


Fig. 3.3 Block Diagram of Iterative Laplacian of Laplacian based Global Pre-processing (ILLGP) Scheme

To incorporate much high frequency contents into the up-sampled and thus blurred image, the LLGP algorithm may be operated iteratively. Such a new scheme is suggested in this section. The proposed ILLGP scheme, illustrated in Fig. 3.3, employs Laplacian of Laplacian (LOL) kernel iteratively on the sub-sampled image to effectively enhance its very high frequency components. The ILLGP scheme enhances the very high frequency information by iteratively convolving the sub-sampled image with LOL kernel. However, the number of iteration plays a major role in determining the restoration performance of the algorithm.

To determine the number of iterations, performance characteristics between PSNR (dB) and weight factor are plotted for different iterations corresponding to various images as depicted in Fig. 3.5. The high frequency restoration performance of the algorithm is found to be maximum during the second iteration and then gradually reduces towards the higher order iterations. Although the high frequency extracts are more enhanced using higher order derivative, they undergo more deformation toward higher order derivatives. Consequently, the restoration performance declines towards higher order iterations. Hence, the number of iteration is kept two to have a better restored image quality. Therefore, ILLGP algorithm which is based on 8^{th} order derivative, employs LOL operator twice as a 4^{th} order derivative operator. Consequently, the weighted version of very high frequency extract is used to sharpen the sub-sampled image so as to reduce the degree of blurring in the subsequent Lanczos-3 based up-sampling process.

3.3.1 ILLGP Algorithm

The ILLGP algorithm is almost same as LLGP algorithm with the following modifications. Let $\nabla^4 g(x,y)$ be the output using Laplacian of Laplacian (LOL) operator as per LLGP algorithm. $\nabla^8 g(x,y)$ denotes the high frequency extracts which is obtained after two iterations by further convolving $\nabla^4 g(x,y)$ with Laplacian of Laplacian kernel, $h_{LOL}(x,y)$ and is given by,

$$\nabla^8 g(x,y) = \sum_{s=-2}^{2}\sum_{t=-2}^{2} h_{LOL}(s,t)\ \nabla^4 g(x+s, y+t) \tag{3.15}$$

The weighted version of $\nabla^8 g(x,y)$ is superimposed on the sub-sampled intra-frame to generate the sharpened image $g_s(x,y)$ and is given by,

$$g_s(x,y) = g(x,y) + K[\nabla^8 g(x,y)] \tag{3.16}$$

where, K is the weight factor.

3.4 Experiment and Simulation

To evaluate the HF restoration performance of the proposed algorithms, different images are down-sampled at 4:1 compression ratio (CR). Afterward, the down-sampled images are re-scaled to their original size using the proposed algorithms for comparison with the original images. The computational complexities, in terms of CPU execution time of the proposed algorithms, are computed and are compared with the existing algorithms to determine their feasibility for real-time applications. In addition, peak-signal-to-noise-ratio (PSNR) in dB and universal quality index (UQI) are measured to determine the objective performance of the proposed algorithms. The figures and tables showing the performance of the existing and proposed algorithms are explained below.

Fig. 3.4 shows the PSNR variations of the proposed schemes with respect to weight factor for different types of images, meant for overall weight factor estimation. Fig. 3.5 shows the variations of PSNR with respect to weight factor for different iterations which are meant for determining the number of iterations in ILLGP algorithm for maximum objective performance irrespective of image types. Fig. 3.6 reveals the PSNR comparisons of various existing and proposed schemes for various video sequences at 4:1 compression ratio. Fig. 3.7, Fig. 3.12 and Fig. 3.13 show the subjective performance of Lena (512×512), Boat (512×512) and Goldhill

(512×512) images, respectively using various up-sampling schemes at 4:1 compression ratio. In case of Fig. 3.7, four distinct regions with different features and thus different signal characteristics such as low, medium, high and their combinations are marked. Performance at these regions are analyzed. For this purpose, the output images at these regions are enlarged and shown in Fig. 3.8, Fig. 3.9, Fig. 3.10 and Fig. 3.11. The error images of Lena (512×512) corresponding to various schemes are given in Fig. 3.14. Table 3.1 shows the average weight factor estimation of the proposed algorithms. Table 3.2 and Table 3.3 show the PSNR and UQI comparison of different existing and proposed schemes at 4:1 and 16:1 compression ratio respectively. Table 3.5 shows the execution time of various existing and proposed schemes at 4:1 compression ratio.

3.4.1 Estimation of Weight Factor, K

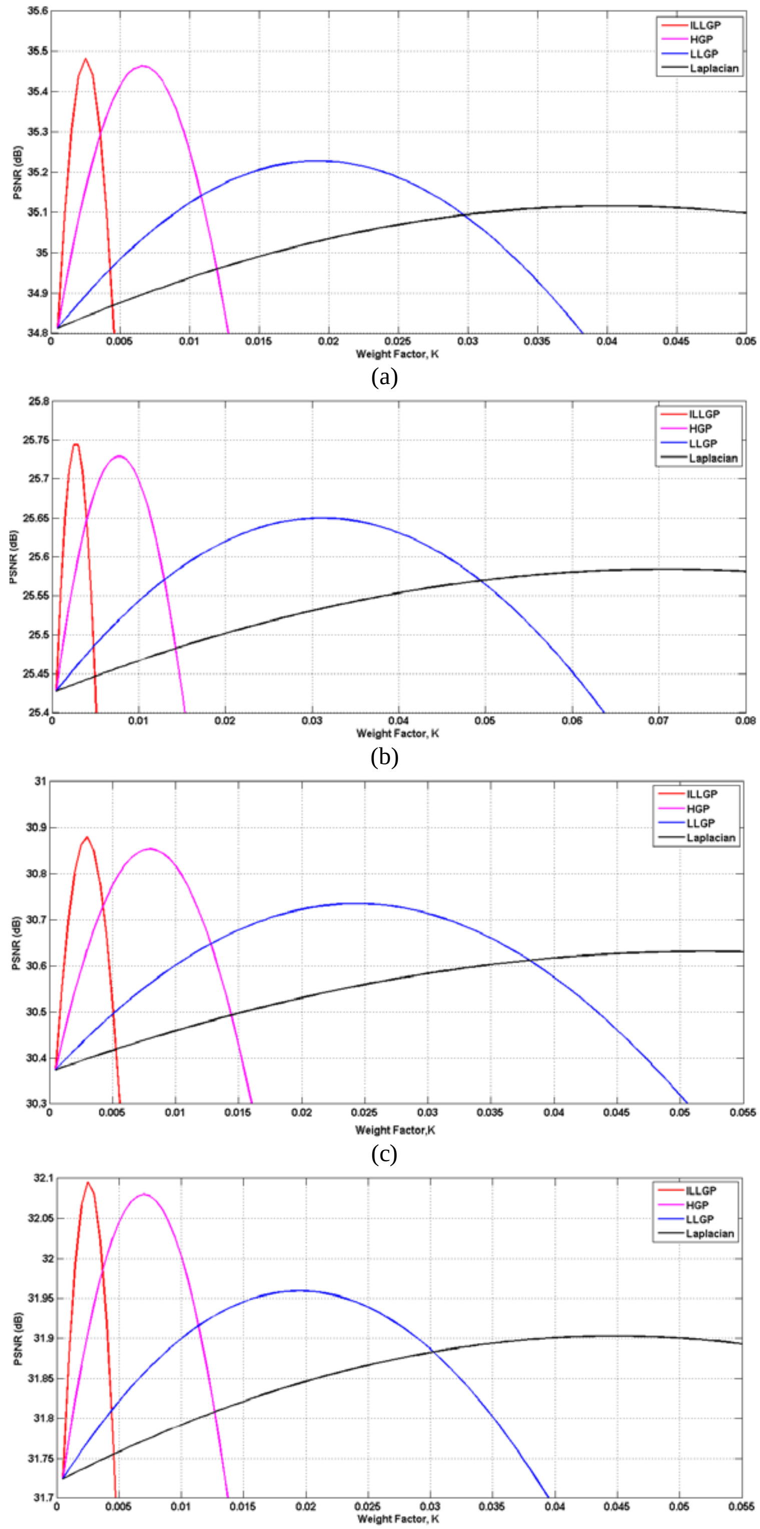
The HF restoration performance in terms of PSNR (dB) of the proposed algorithms is a function of weight factor and hence, the performance depends on the precise estimation of weight factor. Simulation studies are carried out to observe the variation of PSNR (dB) with respect to the weight factor, K for different images to determine a general, optimized weight factor as shown in Fig. 3.4. The gradients of PSNR vs. K plots vary directly according the order of the derivative operator used in the algorithms as depicted in Fig. 3.4. The ILLGP algorithm which makes use of 8th order derivative has the highest gradient of the plot whereas in case of Laplacian it is found to be the least because the Laplacian uses 2nd order derivative. Higher the gradient, the more localized are the peak PSNR variations for different images and hence is helpful in precise estimation of a generalized weight factor. The ILLGP, HGP and LLGP algorithms, which are capable of extracting HF extracts, have more localized peak PSNR variations as compared to Laplacian. In addition, their weight factor deviations are much less with respect to different images as compared to the Laplacian as shown in Table 3.1. In general, the overall weight factors are estimated by averaging the weight factors corresponding to different image types for better performance as illustrated in Table 3.1.

In case of ILLGP algorithm, the determination of number of iterations is the determining factor for the performance of the algorithm. Hence, simulation studies are carried out to observe the variations of PSNR with respect to weight factor for three iterations as shown in Fig. 3.5. It is well observed from the plot that the global maximum is achieved at 2nd iteration irrespective of the image types. Hence, the optimum HF restoration performance is achieved using 8th order derivative operator. However, the performance declines towards higher order derivatives. The final weight factors for the various proposed algorithms are given below.

$$K = \begin{cases} 0.0487, \; Laplacian \\ 0.0235, \; LLGP \\ 0.0072, \; HGP-I \\ 0.0072, \; HGP-II \\ 0.0025, \; ILLGP \end{cases} \tag{3.17}$$

Table 3.1 Weight factor, K estimation of the proposed pre-processing schemes

Algorithm	Weight factors corresponding to the maximum PSNR for different images				Average Weight Factor
	Lena	**Boat**	**Barbara**	**Goldhill**	
Laplacian	0.04	0.04	0.07	0.045	**0.0487**
LLGP	0.019	0.024	0.0315	0.0195	**0.0235**
HGP	0.0065	0.008	0.0075	0.007	**0.0072**
ILLGP	0.0024	0.0026	0.0025	0.0025	**0.0025**



(a) (b) (c) (d)

Fig. 3.4 PSNR vs. weight factor plots of different images using the proposed algorithms at 4:1 compression ratio: (a) Lena; (b) Barbara; (c) Boat; (d) Goldhill

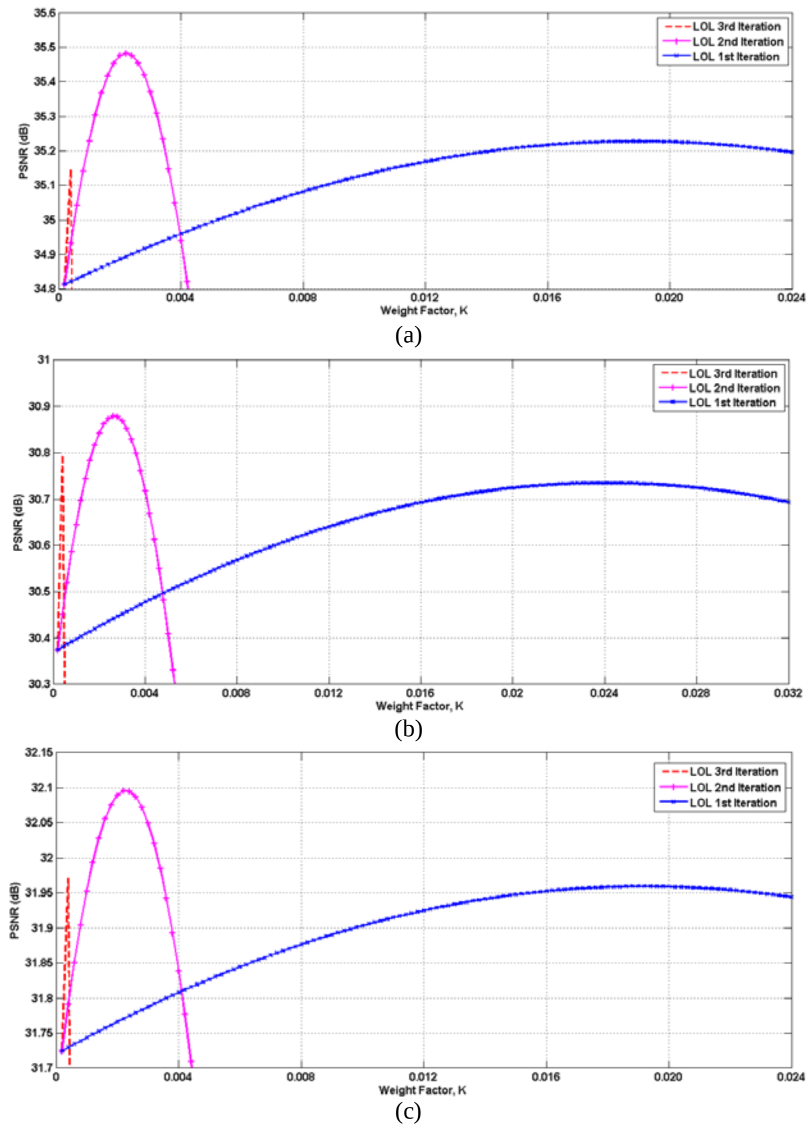


Fig. 3.5 PSNR (dB) vs. weight factor characteristic plot of different images for different iterations using ILLGP: (a) Lena; (b) Boat; (c) Goldhill

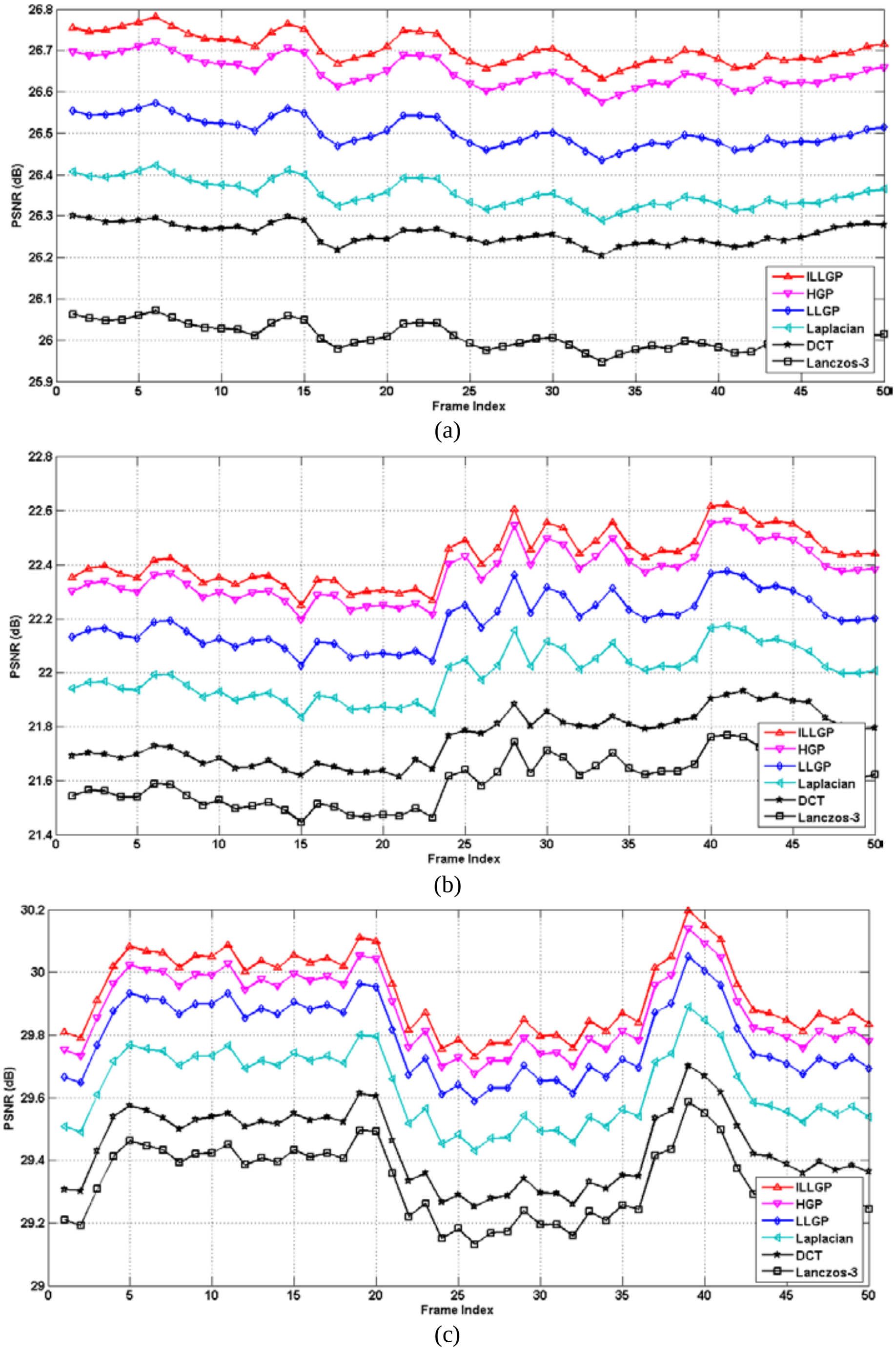


Fig. 3.6 PSNR (dB) comparisons of various up-sampling schemes at 4:1 CR meant for different sequences: (a) Container; (b) Mobile; (c) Salesman

Table 3.2 PSNR (dB) and UQI comparison of different schemes at 4:1 compression ratio for various (512×512) images

Image	**Image Metric**	**Bilinear [5]**	**Bicubic [4]**	**Lanczos3 [12]**	**DCT [74]**	**Laplacian**	**LLGP [P1]**	**HGP-I [P2]**	**HGP-II**	**ILLGP [P3]**
Mandril	PSNR	23.045	23.630	23.859	23.925	24.078	24.174	24.263	24.205	**24.288**
	UQI	0.8957	0.9114	0.9170	0.9187	0.9229	0.9247	0.9267	0.9250	**0.9271**
Lena	PSNR	32.704	34.148	34.813	35.023	35.101	35.200	**35.440**	35.204	35.439
	UQI	0.9922	0.9945	0.9953	0.9955	0.9956	0.9957	**0.9959**	0.9957	0.9959
Barbara	PSNR	24.925	25.352	25.428	25.183	25.569	25.637	25.729	25.661	**25.744**
	UQI	0.9631	0.9669	0.9675	0.9657	0.9688	0.9693	0.9700	0.9694	**0.9701**
Boat	PSNR	28.940	29.952	30.375	30.466	30.631	30.735	30.854	30.762	**30.879**
	UQI	0.9801	0.9845	0.9860	0.9863	0.9870	0.9872	0.9876	0.9873	**0.9876**
Goldhill	PSNR	30.574	31.405	31.725	31.716	31.901	31.946	32.076	31.951	**32.080**
	UQI	0.9880	0.9901	0.9909	0.9909	0.9913	0.9914	0.9916	0.9914	**0.9916**
Pirate	PSNR	30.027	31.058	31.490	31.606	31.761	31.874	32.027	31.900	**32.047**
	UQI	0.9853	0.9885	0.9897	0.9899	0.9904	0.9906	0.9909	0.9906	**0.9910**
Livingroom	PSNR	28.617	29.557	29.977	30.128	30.250	30.366	30.488	30.398	**30.524**
	UQI	0.9761	0.9811	0.9829	0.9835	0.9842	0.9846	0.9850	0.9847	**0.9851**
Fingerprint	PSNR	28.045	30.632	31.722	32.133	31.910	32.423	32.758	32.518	**32.798**
	UQI	0.9785	0.9889	0.9915	0.9922	0.9922	0.9929	0.9934	0.9930	**0.9934**
Baboon	PSNR	33.588	35.014	35.662	35.890	35.884	35.929	**36.161**	35.916	36.142
	UQI	0.9924	0.9946	0.9954	0.9956	0.9956	0.9957	**0.9959**	0.9956	0.9959
Bridge	PSNR	25.728	26.504	26.826	26.919	27.078	27.197	27.301	27.233	**27.333**
	UQI	0.9693	0.9748	0.9767	0.9772	0.9783	0.9789	0.9794	0.9790	**0.9795**
Cameraman	PSNR	33.214	35.757	37.216	37.832	37.723	37.909	**38.362**	37.917	38.349
	UQI	0.9959	0.9977	0.9984	0.9986	0.9986	0.9986	**0.9988**	0.9986	0.9988
Cat	PSNR	30.949	31.982	32.427	32.563	32.730	32.881	33.007	32.925	**33.046**
	UQI	0.9920	0.9937	0.9943	0.9945	0.9948	0.9949	0.9951	0.9950	**0.9951**
Crowd	PSNR	30.984	32.666	33.451	33.768	33.886	34.148	34.357	34.218	**34.415**
	UQI	0.9894	0.9930	0.9942	0.9946	0.9948	0.9951	0.9953	0.9951	**0.9953**
Cycle	PSNR	21.208	21.895	22.154	22.129	22.387	22.491	22.610	22.521	**22.633**
	UQI	0.9323	0.9437	0.9475	0.9474	0.9514	0.9524	0.9539	0.9526	**0.9541**
F16	PSNR	30.379	31.543	32.104	**32.722**	32.260	32.299	32.484	32.287	32.484
	UQI	0.9848	0.9885	0.9900	**0.9913**	0.9905	0.9905	0.9909	0.9905	0.9909
House	PSNR	29.248	30.314	30.807	30.862	31.202	31.405	31.530	31.469	**31.596**
	UQI	0.9719	0.9785	0.9809	0.9812	0.9829	0.9836	0.9841	0.9838	**0.9843**
Lake	PSNR	28.945	30.022	30.495	30.793	30.735	30.867	30.988	30.901	**31.021**
	UQI	0.9895	0.9919	0.9928	0.9933	0.9932	0.9934	0.9936	0.9934	**0.9936**
Peppers	PSNR	31.180	31.991	32.328	**32.747**	32.459	32.525	32.637	32.532	32.652
	UQI	0.9923	0.9937	0.9942	**0.9947**	0.9944	0.9945	0.9946	0.9945	0.9946
Elaine	PSNR	32.534	33.117	33.309	33.284	33.322	33.280	**33.406**	33.249	33.371
	UQI	0.9913	0.9924	0.9928	0.9927	0.9928	0.9928	**0.9930**	0.9927	0.9929
Ruler	PSNR	12.335	12.613	12.673	12.600	12.778	12.837	12.861	12.859	**12.890**
	UQI	0.5188	0.5735	0.5898	0.5920	0.6176	0.6273	0.6293	0.6288	**0.6351**

Table 3.3 PSNR (dB) and UQI comparison of different schemes at 16:1 compression ratio for various (512×512) images

Image	**Image Metric**	**Bilinear [5]**	**Bicubic [4]**	**Lanczos3 [12]**	**DCT [74]**	**Laplacian**	**LLGP [P1]**	**HGP [P2]**	**ILLGP [P3]**
Mandril	PSNR(dB)	20.883	21.085	21.156	21.167	21.226	21.252	21.287	**21.292**
	UQI	0.8191	0.8309	0.8351	0.8362	0.8408	0.8417	0.8438	**0.8440**
Lena	PSNR(dB)	28.053	28.848	29.183	29.296	29.361	29.419	**29.557**	29.556
	UQI	0.9767	0.9810	0.9825	0.9829	0.9835	0.9836	**0.9841**	0.9841
Barbara	PSNR(dB)	23.351	23.607	23.708	23.747	23.764	23.788	**23.831**	23.829
	UQI	0.9457	0.9496	0.9510	0.9515	0.9523	0.9524	**0.9528**	0.9528
Boat	PSNR(dB)	25.041	25.538	25.739	25.773	25.888	25.939	26.015	**26.024**
	UQI	0.9493	0.9557	0.9580	0.9584	0.9601	0.9605	0.9612	**0.9612**
Goldhill	PSNR(dB)	27.166	27.628	27.798	27.776	27.882	27.893	**27.979**	27.973
	UQI	0.9731	0.9761	0.9771	0.9770	0.9778	0.9778	**0.9782**	0.9782
Pirate	PSNR(dB)	26.286	26.861	27.083	27.127	27.228	27.283	27.372	**27.379**
	UQI	0.9641	0.9691	0.9708	0.9712	0.9723	0.9725	0.9731	**0.9731**
Livingroom	PSNR(dB)	24.932	25.385	25.560	25.563	25.691	25.736	25.806	**25.815**
	UQI	0.9420	0.9488	0.9511	0.9513	0.9535	0.9539	0.9546	**0.9547**
Fingerprint	PSNR(dB)	20.920	22.633	23.779	24.202	24.537	24.745	24.834	**24.948**
	UQI	0.8791	0.9192	0.9407	0.9467	0.9540	0.9566	0.9583	**0.9589**
Baboon	PSNR(dB)	28.812	29.604	29.931	30.007	30.123	30.173	**30.321**	30.318
	UQI	0.9766	0.9808	0.9823	0.9826	0.9834	0.9835	**0.9840**	0.9840
Bridge	PSNR(dB)	22.676	23.066	23.230	23.263	23.353	23.407	23.460	**23.473**
	UQI	0.9357	0.9423	0.9449	0.9454	0.9473	0.9479	0.9485	**0.9487**
Cameraman	PSNR(dB)	26.633	27.546	27.946	28.094	28.233	28.344	28.509	**28.532**
	UQI	0.9809	0.9847	0.9861	0.9866	0.9872	0.9875	0.9879	**0.9880**
Cat	PSNR(dB)	27.295	27.838	28.059	28.105	28.237	28.318	28.399	**28.417**
	UQI	0.9812	0.9835	0.9844	0.9845	0.9851	0.9854	0.9856	**0.9857**
Crowd	PSNR(dB)	25.759	26.601	26.974	27.066	27.227	27.358	27.470	**27.497**
	UQI	0.9634	0.9706	0.9733	0.9739	0.9754	0.9760	0.9766	**0.9767**
Cycle	PSNR(dB)	18.693	18.996	19.109	19.135	19.205	19.241	19.295	**19.302**
	UQI	0.8727	0.8842	0.8883	0.8893	0.8935	0.8943	0.8957	**0.8959**
F16	PSNR(dB)	25.767	26.457	26.773	26.939	26.909	26.932	**27.058**	27.055
	UQI	0.9540	0.9617	0.9647	0.9661	0.9666	0.9666	**0.9675**	0.9675
House	PSNR(dB)	25.401	25.863	26.050	26.063	26.193	26.258	26.305	**26.322**
	UQI	0.9285	0.9373	0.9406	0.9409	0.9437	0.9444	0.9450	**0.9452**
Lake	PSNR(dB)	24.783	25.466	25.746	25.834	25.937	26.026	26.119	**26.139**
	UQI	0.9720	0.9764	0.9780	0.9785	0.9793	0.9796	0.9801	**0.9801**
Peppers	PSNR(dB)	27.359	28.045	28.330	28.524	28.432	28.480	28.574	**28.578**
	UQI	0.9811	0.9841	0.9852	0.9859	0.9859	0.9858	0.9861	**0.9861**
Elaine	PSNR(dB)	29.771	30.496	30.785	30.884	30.818	30.792	**30.935**	30.902
	UQI	0.9832	0.9860	0.9870	0.9878	0.9873	0.9871	**0.9875**	0.9874
Ruler	PSNR(dB)	10.707	10.804	10.873	10.876	10.889	10.895	10.899	**10.902**
	UQI	0.2036	0.2394	0.2608	0.2636	0.2788	0.2804	0.2785	**0.2788**

Table 3.4 Average PSNR (dB) and UQI comparison of different interpolation techniques at 4:1 compression ratio for various sequences over 50 frames

Sequence	Image Metric	Bilinear [5]	Bicubic [4]	Lanczos3 [12]	DCT [74]	Laplacian	LLGP (P1)	HGP (P2)	ILLGP (P3)
Salesman	PSNR(dB)	28.107	28.979	29.327	29.440	29.628	29.788	29.877	**29.934**
	UQI	0.9649	0.9719	0.9742	0.9750	0.9763	0.9772	0.9776	**0.9779**
Bus	PSNR(dB)	24.243	25.262	25.716	25.794	26.130	26.342	26.480	**26.557**
	UQI	0.9491	0.9610	0.9652	0.9660	0.9691	0.9706	0.9715	**0.9720**
Akiyo	PSNR(dB)	31.686	32.911	33.450	33.647	33.820	33.963	34.179	**34.206**
	UQI	0.9927	0.9945	0.9952	0.9954	0.9956	0.9957	0.9959	**0.9959**
City	PSNR(dB)	26.827	27.592	27.879	27.852	28.144	28.257	28.258	**28.419**
	UQI	0.9104	0.9277	0.9333	0.9333	0.9392	0.9408	0.9425	**0.9431**
Container	PSNR(dB)	24.663	25.567	26.008	26.256	26.355	26.503	26.647	**26.705**
	UQI	0.9592	0.9676	0.9709	0.9727	0.9737	0.9746	0.9754	**0.9757**
Football	PSNR(dB)	27.106	28.568	29.366	29.687	29.978	30.273	30.521	**30.604**
	UQI	0.9668	0.9769	0.9810	0.9824	0.9839	0.9849	0.9858	**0.9860**
Mobile	PSNR(dB)	20.310	21.197	21.596	21.758	21.995	22.192	22.370	**22.427**
	UQI	0.9373	0.9502	0.9550	0.9569	0.9599	0.9617	0.9634	**0.9639**
Soccer	PSNR(dB)	29.271	30.254	30.664	30.748	30.927	31.009	31.201	**31.217**
	UQI	0.9828	0.9863	0.9876	0.9878	0.9884	0.9887	0.9891	**0.9892**
Coast	PSNR(dB)	25.585	26.500	26.939	27.080	27.328	27.521	27.655	**27.733**
	UQI	0.9692	0.9754	0.9779	0.9787	0.9801	0.9810	0.9815	**0.9819**

Table 3.5 Execution time of the proposed and existing algorithms at 4:1 CR

Images of different size (M×N)	Execution time in Seconds for different interpolation schemes							
	Bilinear [5]	**Bicubic [4]**	**Lanczos3 [12]**	**DCT [74]**	**Laplacian**	**LLGP [P1]**	**HGP [P2]**	**ILLGP [P3]**
Clock (200×200)	0.0141	0.0150	0.0151	0.0518	0.0933	0.0946	0.1291	0.1300
Lena (256×256)	0.0147	0.0152	0.0157	0.0633	0.0942	0.0987	0.1315	0.1328
Fruit (377×321)	0.0153	0.0163	0.0168	0.1629	0.0997	0.1027	0.1332	0.1395
Lena (512×512)	0.0201	0.0215	0.0222	0.1773	0.1023	0.1137	0.1388	0.1420
Pentagon (1024×1024)	0.0354	0.0409	0.0442	0.6372	0.1175	0.1281	0.1680	0.1767

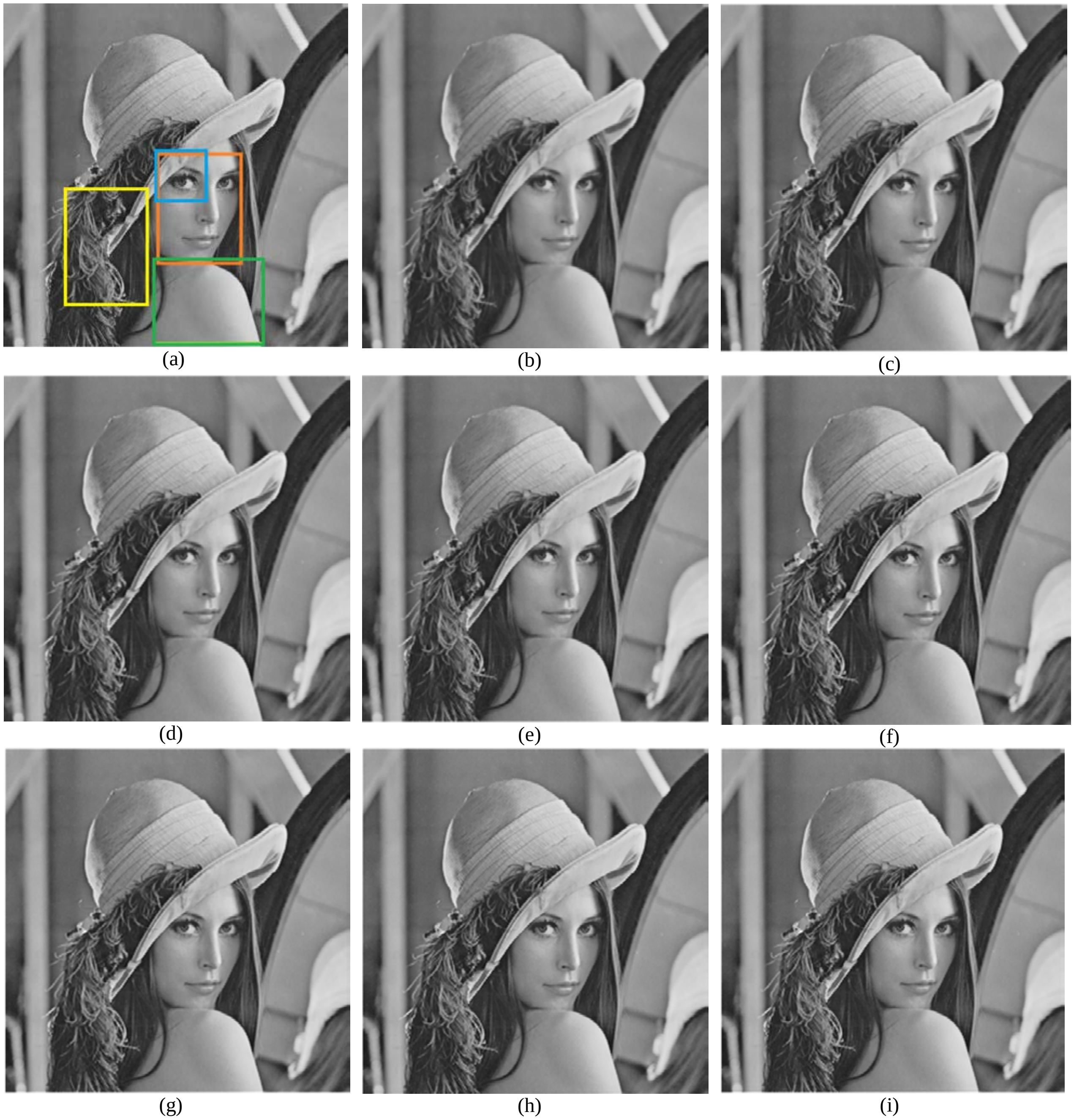


Fig. 3.7 Subjective evaluation of Lena (512×512) image using various up-sampling schemes at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) Laplacian; (g) LLGP (P1) ; (h) HGP (P2); (i) ILLGP (P3)

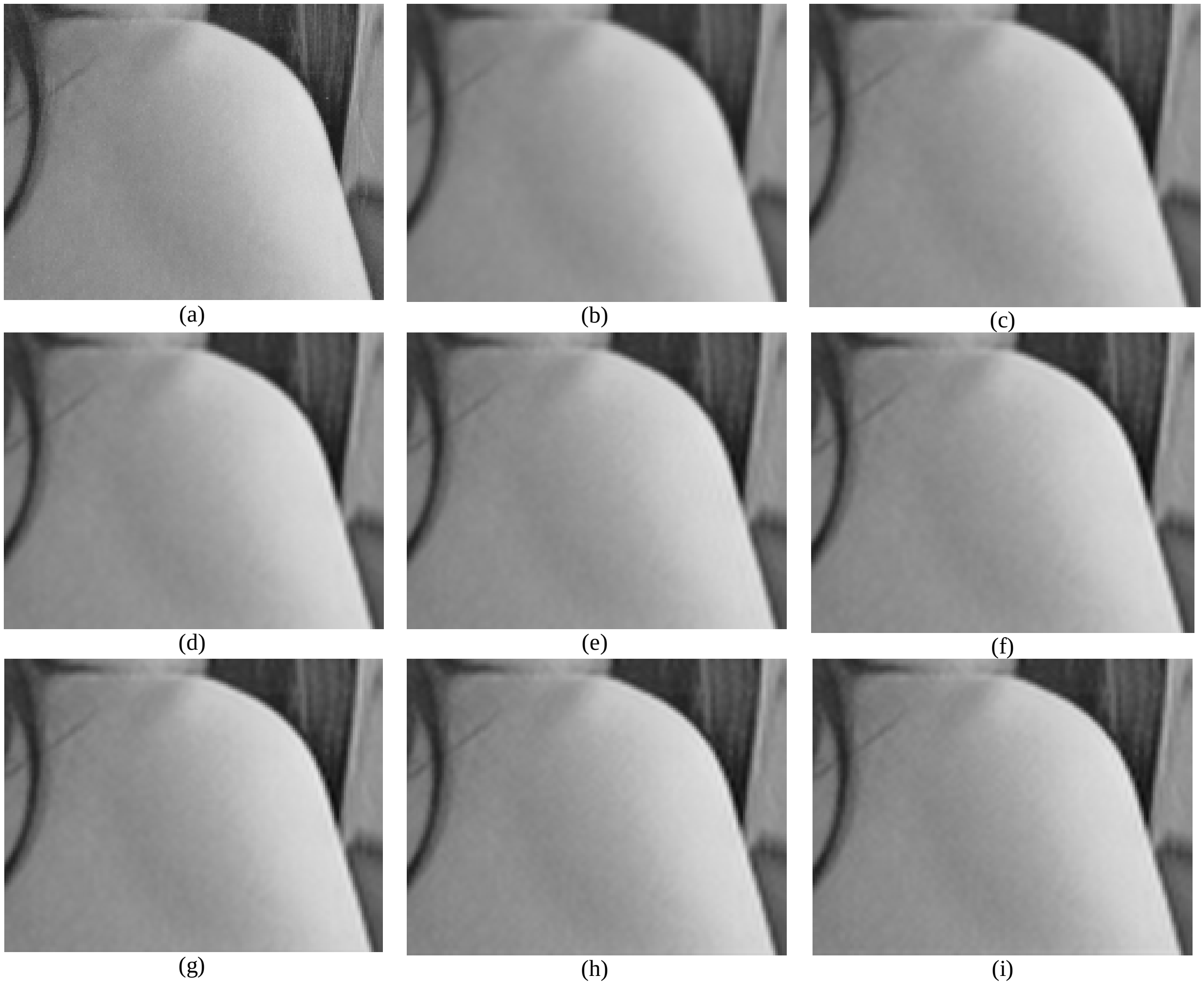
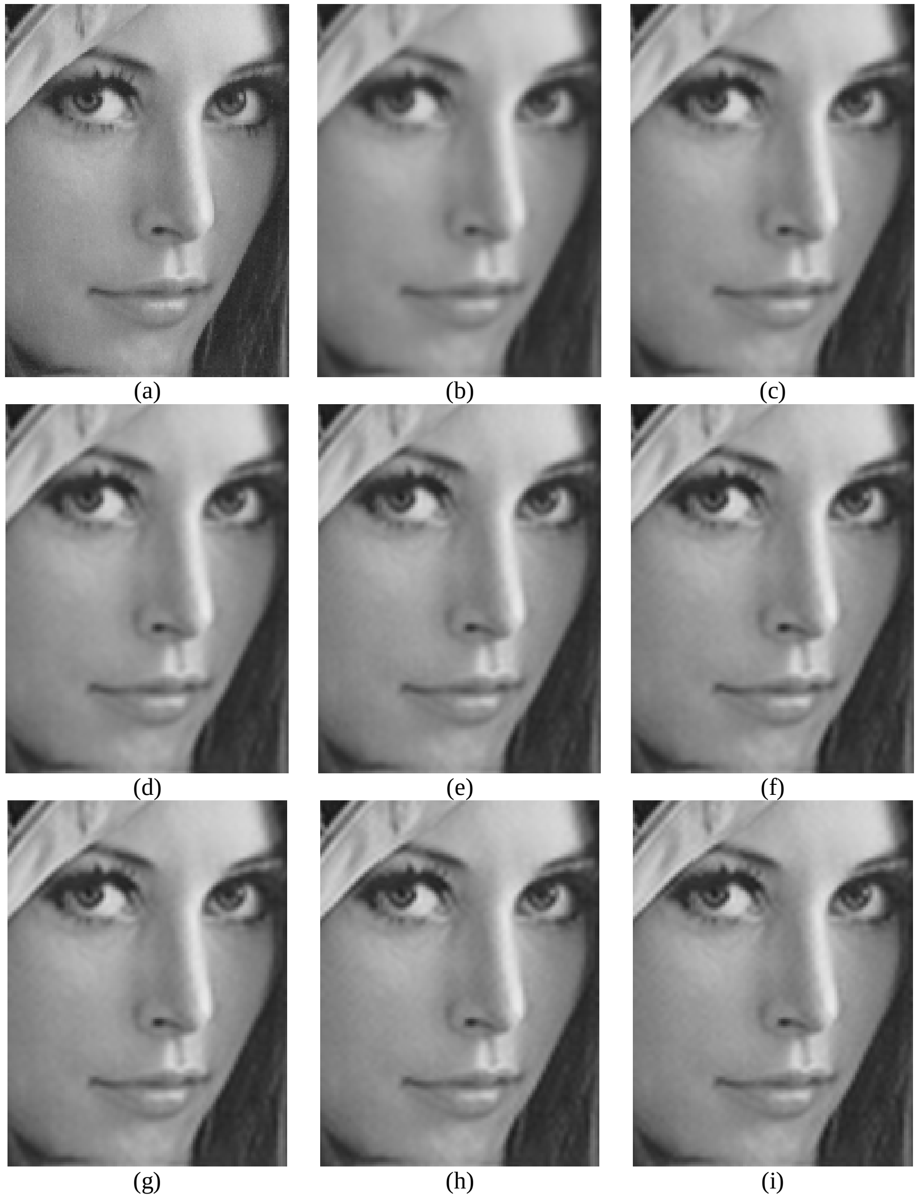
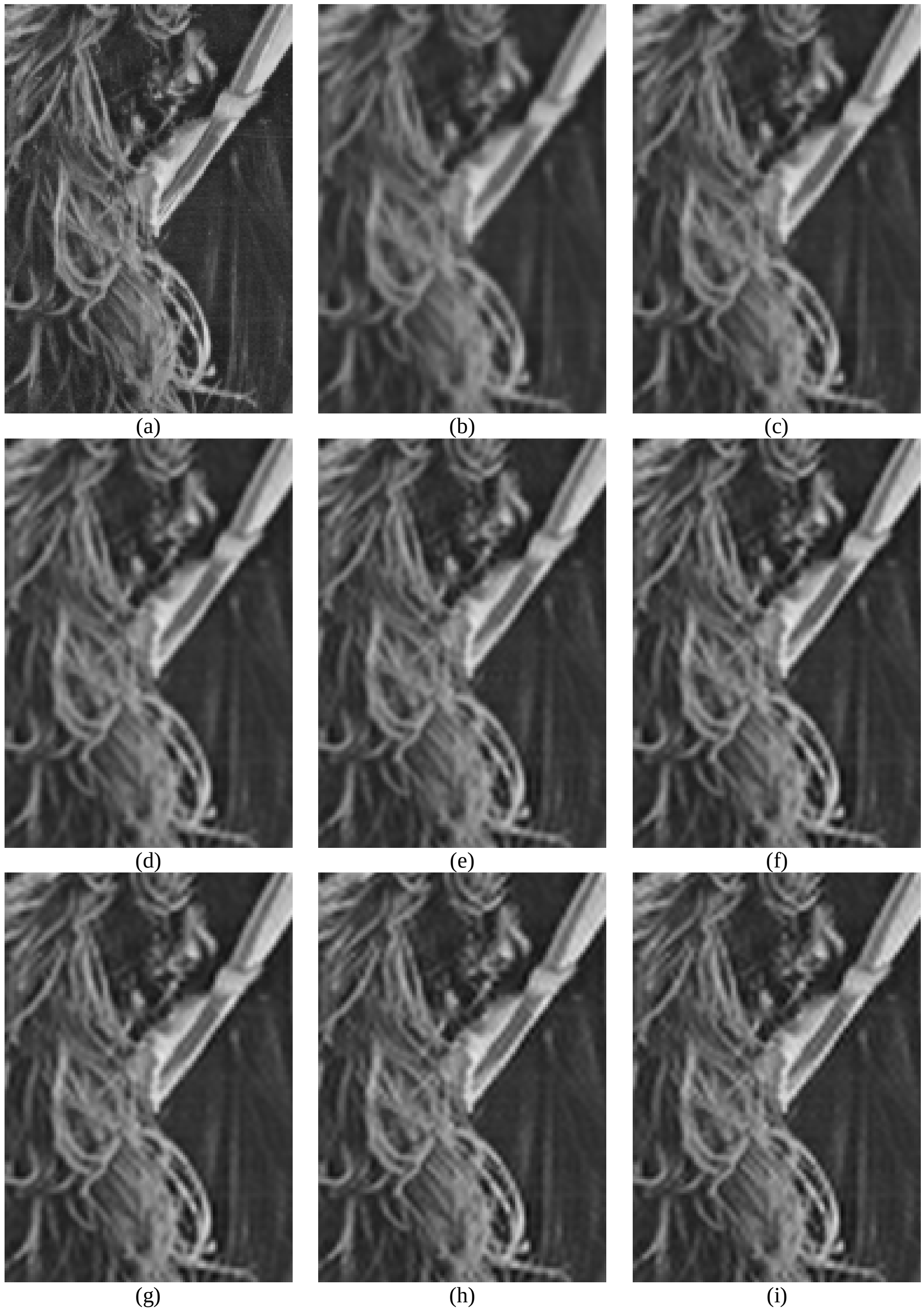


Fig. 3.8 Subjective evaluation of the selected low frequency green rectangular region (127×164) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) Laplacian; (g) LLGP(P1); (h) HGP (P2); (i) ILLGP (P3)



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 3.9 Subjective evaluation of the selected medium frequency orange rectangular region (164×125) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) Laplacian; (g) LLGP(P1); (h) HGP (P2); (i) ILLGP (P3)



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 3.10 Subjective evaluation of the selected high frequency yellow rectangular region (123×174) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) Laplacian; (g) LLGP(P1); (h) HGP (P2); (i) ILLGP (P3)

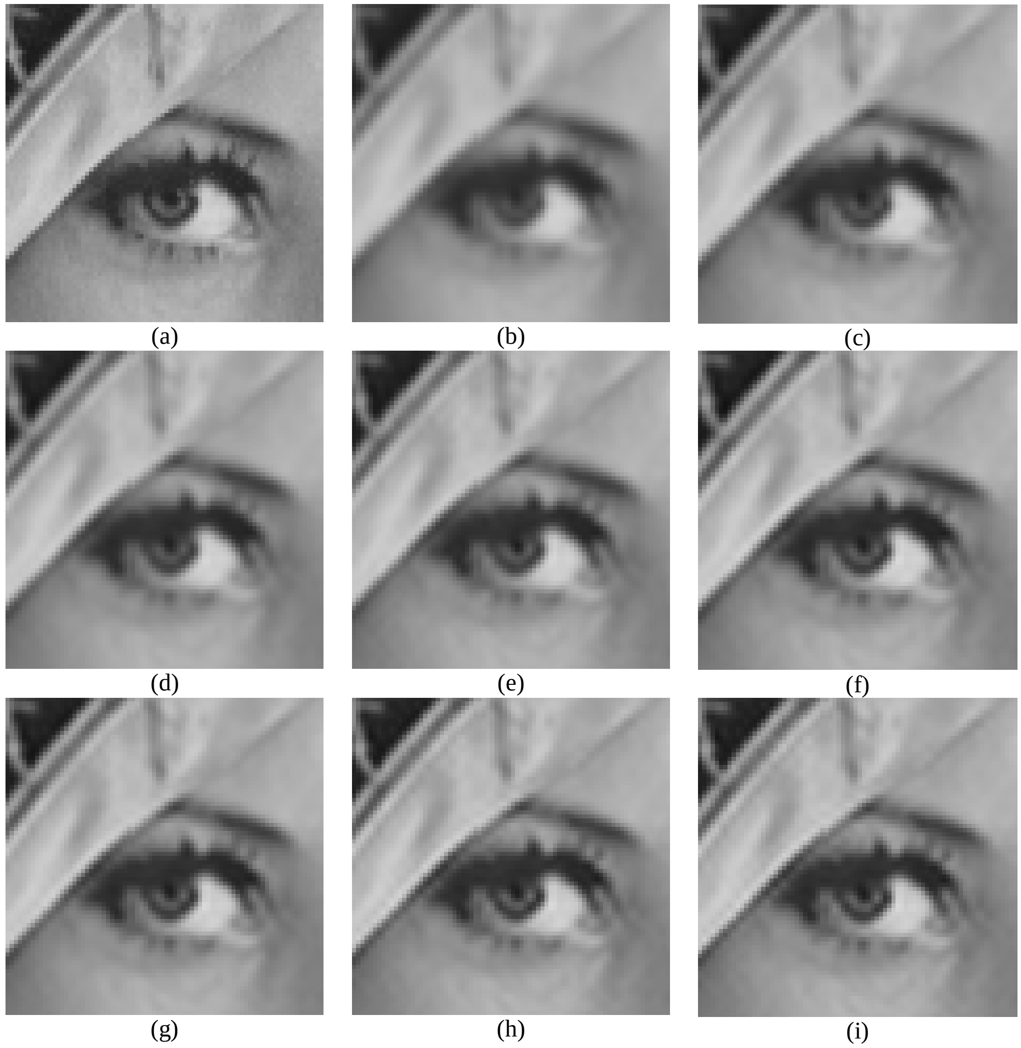


Fig. 3.11 Subjective evaluation of the selected blue rectangular (76×76) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) Laplacian; (g) LLGP(P1); (h) HGP (P2); (i) ILLGP (P3)

Fig. 3.12 Subjective evaluation of Boat (512×512) image using various up-sampling schemes at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) Laplacian; (g) LLGP (P1) ; (h) HGP (P2); (i) ILLGP (P3)

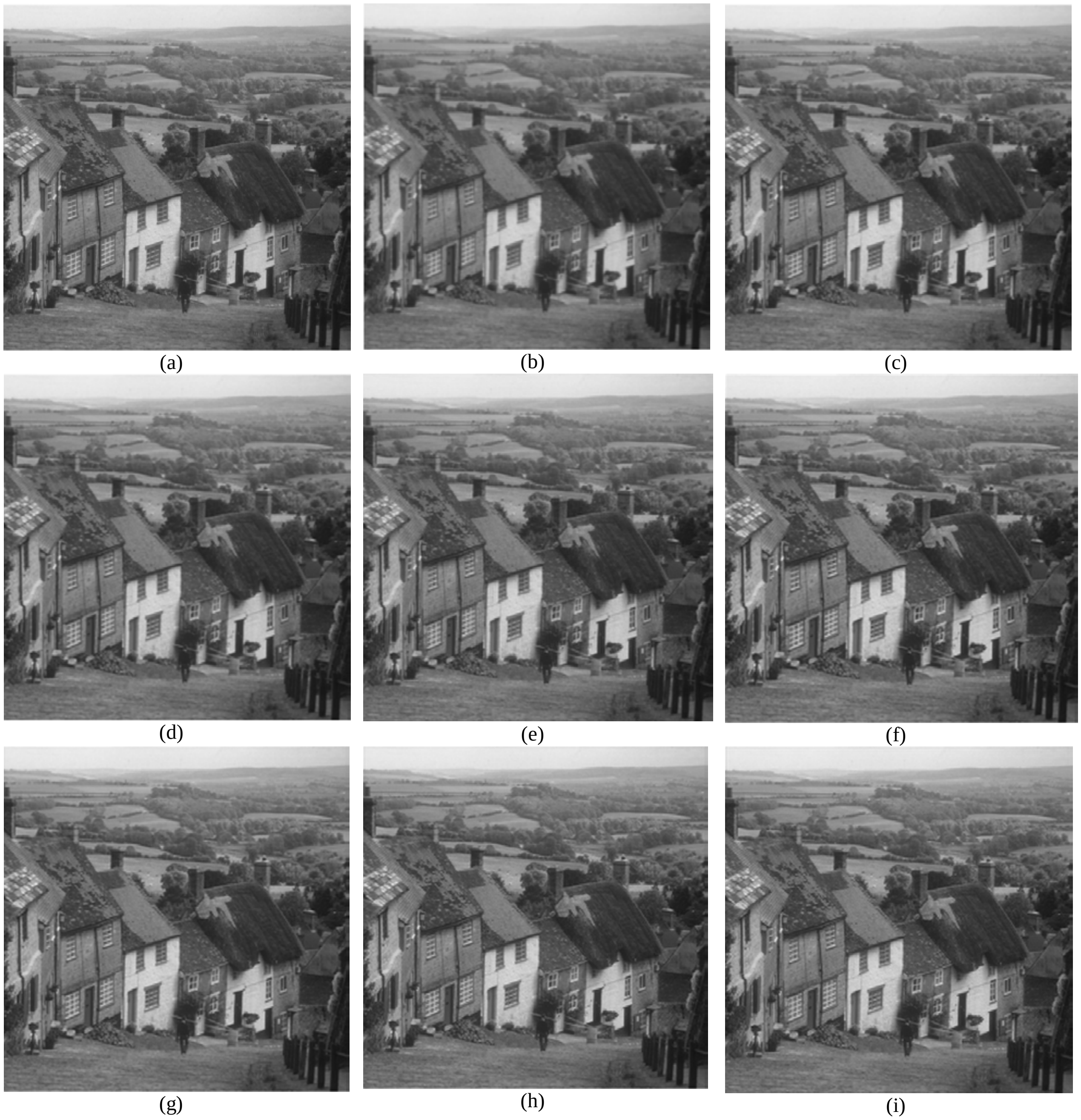


Fig. 3.13 Subjective evaluation of Goldhill (512×512) image using various up-sampling schemes at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) Laplacian; (g) LLGP (P1) ; (h) HGP (P2); (i) ILLGP (P3)

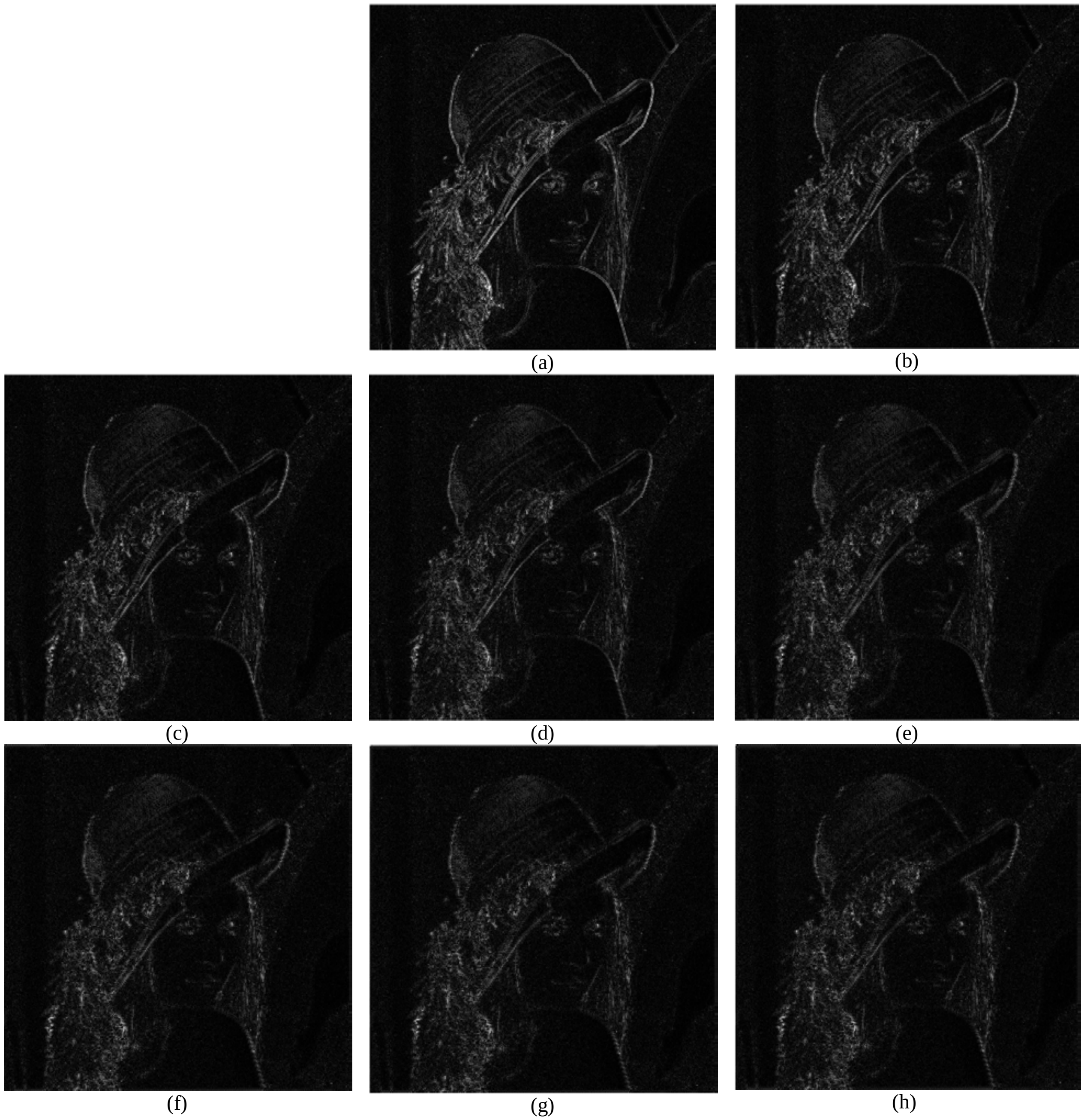


Fig. 3.14 Error image of Lena (512×512) using various up-sampling schemes at 4:1 compression ratio: (a) Bilinear; (b) Bicubic; (c) Lanczos-3; (d) DCT; (e) Laplacian; (f) LLGP (P1) ; (g) HGP (P2); (h) ILLGP (P3)

3.5 Results and Discussion

The proposed algorithms show better results than many of the existing algorithms in terms of subjective and objective measures.

For most of the images, the proposed algorithms achieve better objective performances than many of the existing algorithms available in the literature as illustrated in Table 3.2, Table 3.3 and Table 3.4. In case of House (512×512) image, the ILLGP algorithm achieves a maximum PSNR improvement of 0.734 dB than DCT at 4:1 compression ratio. In addition, it attains a better PSNR gain of 0.561 dB, 0.665 dB and 0.647 dB in case of Barbara, Fingerprint and Crowd respectively. Since LLGP, HGP and ILLGP algorithms are based on the HF enhancement of an image, they exhibit better performance in case of images rich in HF patterns like Barbara, Fingerprint and crowd. However, the performance of these algorithms is moderate for all other images.

Besides this, the proposed algorithms show better performance for various common intermediate format (CIF) video sequences as depicted in Table 3.4. In case of Football (352×288) sequence, the ILLGP algorithm achieves an improved average PSNR gain of 0.917 dB over DCT at 4:1 compression ratio. So, these algorithms achieve better objective performance for video signals of different resolutions.

Furthermore, these algorithms provide considerable performance improvement irrespective of the variation in compression ratio. Table 3.2 and Table 3.3 show the PSNR and UQI improvements at 4:1 and 16:1 compression ratio respectively.

The proposed algorithms are computationally less complex than DCT because they are the pre-processing schemes and operate on the low resolution images as illustrated in Table 3.5. The DCT, being a frequency domain interpolation technique, has more computational time requirement than the proposed algorithms because of the need for conversion from spatial to frequency domain and back. It may be observed from the table that the execution time of LLGP algorithm is less than HGP and ILLGP algorithm. The HGP algorithm makes use of LOL and Laplacian operator in succession and hence it is computationally more complex than LLGP. Likewise, the ILLGP algorithm utilizes LOL operator twice and therefore consumes more time than HGP. However, the image quality in case of HGP and ILLGP is found to be better than LLGP at the cost of computational complexity. In addition, it is also found that the difference of execution time between LLGP and ILLGP the algorithms is very less i.e. of the order of one tenth of a second for a 512×512 image and hence both of the algorithms are suitable candidate for real-time applications.

Error image is a quality metric which is used to measure the restoration performance of various algorithms. Let $f(x, y)$ and $\hat{f}(x, y)$ represent the original and restored image respectively. Then, the expression for error image, in terms of absolute error, is given by,

$$e(x, y) = \left| \hat{f}(x, y) - f(x, y) \right| \tag{3.18}$$

The error images are shown in Fig. 3.14. For better visibility, a scale factor of 5 is employed. It may be observed from this figure that the absolute error is reduced in case of the proposed algorithms as compared to DCT and other existing algorithms which indicate a better HF restoration performance.

Lena (512×512) image, which has different regions of low, medium and high frequencies, is a suitable candidate for subjective evaluation of the proposed algorithms. The low, medium, high frequency patterns of Lena image are considered for performance evaluation of the algorithms. The region rich in combination of different patterns is also considered for performance evaluation. Four distinct regions with different features and thus different signal characteristics are marked as shown in Fig. 3.7. Performance at these regions are distinctly analyzed. For this purpose, the output images at these regions are enlarged and shown in Fig. 3.8, Fig.3.9, Fig. 3.10 and Fig. 3.11

The green rectangular region (127×164) containing the shoulder portion of Lena image is considered as a low frequency region. The enlarged version of it using various algorithms is given in Fig. 3.8. Similarly, the enlarged versions of medium and high frequency regions are given in Fig. 3.9 and 3.10, respectively. The face and hair regions represent the medium and high frequencies, respectively. Fig 3.11 shows the eye and its surrounding region comprising low, medium and high frequencies. In all these cases, it may be perceived that low frequency regions are well preserved, the mid frequency regions are moderately enhanced and the HF regions are highly emphasized so as to compensate the HF loss during the up-sampling process. In case of the proposed algorithms, the fine details and edge information, which represent the high frequency contents, are effectively enhanced resulting in a better visual quality. However, the degrees of HF enhancement of the proposed algorithms are different depending on the order of derivative operator employed. The degree of HF enhancement is maximum in case of ILLGP which uses 8^{th} order derivative as shown in Fig. 3.10. Likewise, it may be observed from the figures that the HF enhancement is high and moderate in case of HGP and LLGP algorithm, which makes use of 6^{th} order and 4^{th} order derivatives, respectively. Furthermore, the blue rectangular region of Lena which embodies the eye region contains various frequencies. Under this condition, the proposed algorithms work much better than the existing algorithms in restoring the HF contents of an image as illustrated in Fig. 3.11. So, the overall subjective performance of the proposed schemes is more satisfactory than the existing schemes.

From Table 3.2 and Table 3.4, it may be observed that for most of the images and videos, LLGP, HGP and ILLGP algorithms achieve better PSNR gain than DCT at 4:1 compression ratio. In case of Barbara and Fingerprint (512×512) images, ILLGP attains the maximum PSNR gain of 0.561dB and 0.665 dB, respectively and HGP shows similar PSNR hike of 0.546 dB and 0.625dB over DCT, respectively. Likewise, in case of Football sequence, ILLGP gives an average PSNR gain of 0.917 dB whereas HGP attains the average PSNR gain of 0.834. Hence, the performance of ILLGP is slightly better than HGP in most of the images because it employs higher order derivative operator than HGP. The LLGP algorithm achieves the PSNR improvement of 0.454 dB and 0.29 dB than DCT in case of Barbara and Fingerprint images, respectively. It also attains the average PSNR gain of 0.586 dB than DCT in case of Football sequence at 4:1 compression ratio. So, the performance of LLGP is less than HGP and ILLGP algorithm but much better than that of DCT. In such type of images, which are rich in HF pattern, the proposed algorithms work much efficiently for better objective performance.

From Table 3.3, it may be perceived that, ILLGP shows a PSNR gain of 0.26 dB and 0.251 dB than DCT for Lena and Boat images at 4:1 compression ratio. Similar improvement may be seen in case of HGP algorithm. In case of images like Lena and Boat which are rich in low, medium, high frequency regions, the proposed algorithms show moderate performance. Therefore, we feel that the proposed algorithms show much better performances for images rich with HF contents such as Barbara and Fingerprint. However, their performance become moderate in the images such as Lena and Boat that comprise regions with low, medium and high frequencies.

Furthermore, the proposed schemes achieve better performance than the existing schemes at 16:1 compression ratio as depicted in Table 3.3. Hence, these schemes demonstrate better objective performance in terms of PSNR and UQI irrespective of variation in compression ratio, image resolution and image types and so are more versatile.

3.6 Conclusion

The proposed algorithms: LLGP, HGP and ILLGP, which are based on higher order derivatives, perform better in terms of PSNR and UQI than various existing algorithms for various images and video types with a more pronounced edge and fine details preservation. Since the ILLGP algorithm is based on 8^{th} order derivative, it is capable of extracting much finer and subtler details of an image and hence produces the maximum improved performance amongst all the proposed algorithms. The HGP and LLGP algorithms are based on 6^{th} order and 4^{th} order derivatives, respectively and hence HGP performance is better than LLGP and less as compared to that of ILLGP. Down the order, the Laplacian which is based on 2^{nd} order derivative shows the least performance compared to the above mentioned algorithms.

The HGP and ILLGP algorithm attain the noticeable PSNR gains in case of Barbara, Fingerprint and Crowd respectively which are rich in various HF patterns at 4:1 compression ratio. In contrast, these algorithms work moderately with the images equally rich in low, medium and high frequency contents such as Lena and Boat. This is because the proposed algorithms are based on emphasizing the high frequency details of an image prior to up-sampling. Furthermore, in case of LLGP algorithm, the objective performance gain is similar to that of HGP and ILLGP for these images. However, its performance is less in comparison to those algorithms because of the employment of 4^{th} order derivative operator.

The proposed pre-processing schemes are basically based on inverse operation performed on the down-scaled images through the enhancement of HF contents so as to effectively reduce the blurring in its up-sampled counterpart. The degradation of high frequency details is much higher compared to that of the medium and low frequency details during the up-sampling process. Hence, the algorithms such as LLGP, HGP and ILLGP which are more capable of restoring high frequency information shows better objective and subjective performance than other widely used up-sampling schemes.

Since these algorithms are global pre-processing techniques and operate on low-resolution images, have much reduced computational complexities than DCT. So the proposed pre-processing techniques impart very less computational burden on the various displaying devices and hence are suitable for real-time applications. In addition, the proposed algorithms work efficiently under variation in compression ratio and image resolution and hence can be considered to be more versatile.

Chapter 4

Pre-processing Algorithms using Unsharp Masking

Preview

Unsharp masking is one of the schemes, used for the HF enhancement of an image and is employed extensively in printing and publishing industries. In this chapter, the unsharp masking and its variants are exploited for enhancing the HF contents of a low resolution image so as to lessen the blurring effect in its up-scaled counterpart. In case of unsharp masking based pre-processing (UMP) scheme, the blurred version of a low resolution image is used for HF extraction from the original version through image subtraction. The weighted version of the HF extracts are superimposed with the original image to produce a sharpened image prior to image up-scaling to counter blurring effectively. Some variants of unsharp masking are also proposed in this chapter for HF restoration in the up-scaled images. In case of the second proposed scheme, namely iterative unsharp masking (IUM), an unsharp mask is generated using many iterations which contains very high frequency (VHF) components. The very high frequency extracts is the result of signal decomposition in terms of sub-bands using the concept of analysis filter bank. Since the degradation of VHF components is maximum, restoration of such components would produce much better restoration performance.

In another approach, error based sharpening (ES) scheme, the HF degradation due to image up-scaling is extracted and is called prediction error. The prediction error contains the lost high frequency components. When this error is superimposed on the low resolution image prior to image up-sampling, blurring is considerably reduced in the up-scaled images. The HF degradation due to up-sampling is determined by purposely down-scaling an LR image and restoring it back to its original dimension using Lanczos-3 interpolation. Therefore, the HF degradation is determined by the error between the restored image and the original. Since the lanczos-3 interpolation scheme is employed for the determining the error component, superimposition of the error with the LR image reduces the HF degradation in the subsequent up-sampling process which makes use of the same lanczos-3 interpolation.

This results in the preservation of HF details and improvement of the quality of the up-scaled image.

This chapter critically compares the capabilities and limitations of these proposed pre-processing algorithms and their relevance in the perspective of adaptability to the varying conditions, computational complexities and visual quality. The simulation results, presented at the end of the chapter, are quite encouraging.

The organisation of this chapter is given below.

- *Unsharp Masking based Pre-processing (UMP)*
- *Iterative Unsharp Masking (IUM)*
- *Error based Up-sampling (EU) Scheme*
- *Experiment and Simulation*
- *Results and Discussion*
- *Conclusion*

4.1 Unsharp Masking based Pre-processing (UMP) Scheme [P4]

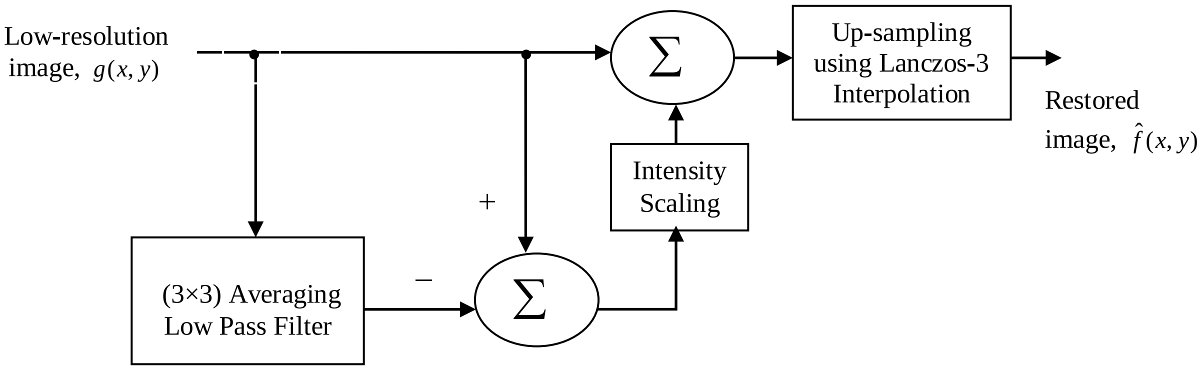
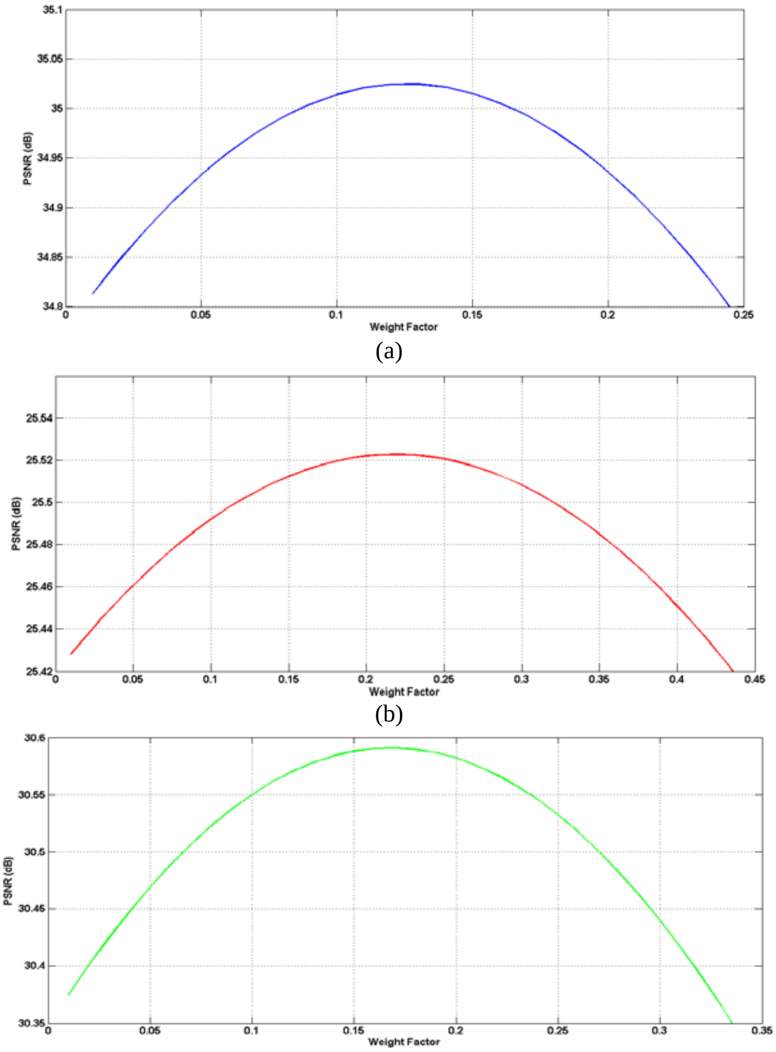


Fig. 4.1 Block Diagram of Unsharp Masking based Pre-processing Scheme

Unsharp masking is a global scheme which is used for image sharpening and is employed here for HF enhancement so that blurring can be reduced in an up-scaled image. In this proposed scheme, the smoothed version of an image is used for extracting the image HF details. The HF details are used for enhancing the fine details and edge regions of an image so as to minimize the HF degradation during the up-sampling process.

The smooth or the blurred version of an image is obtained by low pass filtering using an averaging filter. The blurring occurs due to the loss of HF details of an image. Hence, the lost HF details can be extracted by subtracting the blurred version of an image from its original version which is also called an unsharp mask. The degree of HF enhancement must match with the level of HF degradation for efficient HF restoration. Therefore, the weighted version of the unsharp mask is superimposed upon the original image prior to image up-sampling for effectual HF restoration.

The weight factor, K determines the degree of HF enhancement. In this case, the weight factor is taken as fraction because it deemphasizes the contribution by the unsharp mask so as to provide adequate amount of sharpening to counter the level of blurring effectively. The weight factor is not same for all images but slightly varies from image to image depending on its characteristics and type. Hence, a generalized weight factor is estimated by averaging the weight factors corresponding to four different image types as illustrated in Table 4.1.



(a)
(b)
(c)

Fig. 4.2 PSNR (dB) vs. weight factor plots of different images using the proposed UMP algorithm at 4:1 compression ratio: (a) Lena; (b) Barbara; (c) Boat;

Table 4.1 Weight factor, K estimation of the proposed UMP scheme

Algorithm	Weight factors corresponding to the maximum PSNR for different images				Average Weight Factor
	Lena	Boat	Barbara	Goldhill	
UMP	0.125	0.170	0.225	0.140	$0.165 \approx 0.16$

To determine a generalized, optimum weight factor, simulation studies are carried out to observe the variation of PSNR (dB) with respect to the weight factor, K for different images as shown in Fig. 4.2. The weight factor corresponding to maximum PSNR is taken for each image for overall estimation. The generalized weight factor is computed by averaging the weight factors corresponding to four different images and finally, approximated to 0.16 as shown in Table 4.1.

The proposed algorithm: UMP is presented in the next section.

4.1.1 UMP Algorithm

Let $g(x,y)$ and $g_{Lav}(x,y)$ denote a sub-sampled image and the blurred version of it, respectively. The unsharp mask and sharpened image are denoted by $g_{UM}(x,y)$ and $g_S(x,y)$, respectively. Let them of size $(P \times Q)$. The original image is of size $(2P \times 2Q)$. The UMP algorithm is given below.

Step-1. Select the (3×3) averaging LPF as the blurring kernel, $h_{Lav}(x,y)$.

$$h_{Lav}(x,y) = \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{4.1}$$

Step-2. Select a (3×3) window, w in the sub-sampled image $g(x,y)$.

$w_{s,t}(x,y), \quad -1 \le s,t \le 1$

Step-3. Obtain $g_{Lav}(x,y)$ by linearly convolving $h_{Lav}(x,y)$ with $g(x,y)$.

$$g_{Lav}(x,y) = \sum_{s=-1}^{1} \sum_{t=-1}^{1} h_{Lav}(s,t)\, g(x+s, y+t) \tag{4.2}$$

Step-4. Repeat Step-2 and Step-3 for all (x,y) locations to generate the low pass filtered image, $g_{Lav}(x,y)$.

Step-5. Obtain the unsharp mask by subtracting $g_{Lav}(x,y)$ from $g(x,y)$.

$$g_{UM}(x,y) = g(x,y) - g_{Lav}(x,y) \tag{4.3}$$

Step-6. Generate the sharpened image by superimposing the weighted version of the unsharp mask with the original sub-sampled image.

$$g_S(x,y) = g(x,y) + K\, g_{UM}(x,y) \tag{4.4}$$

where, $K = 0.16$

Step-7. The sharpened image, $g_S(x,y)$ is finally up-scaled by Lanczos-3 interpolation using (2.3) to obtain the up-scaled, restored image, $\hat{f}(x,y)$.

The expression for Lanczos-3 interpolation is given in (2.3) of Chapter 2.

Simulation experiments on this algorithm and results obtained are presented in Section 4.4 and Section 4.5, respectively. An iterative unsharp masking (IUM) algorithm is presented in the next section for further improvement.

4.2 Iterative Unsharp Masking (IUM) Scheme [P5]

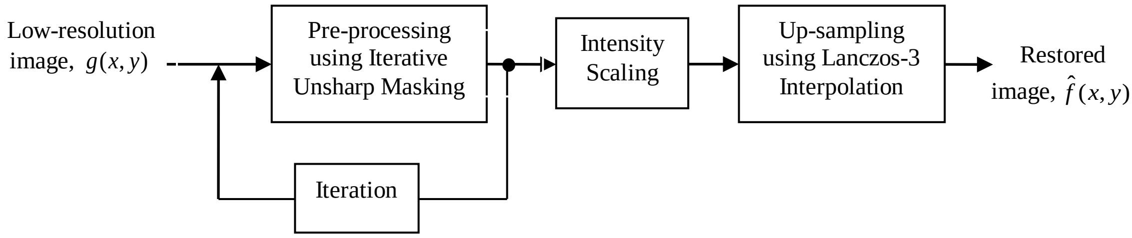


Fig. 4.3 HF Restoration using Iterative Unsharp Masking

The process of image up-scaling results in degradation of fine and edge details in an up-scaled image. The degradation is more significant in the fast changing HF regions as compared to the slowly varying and flat regions resulting in blurring artifacts. This means, the very high frequency (VHF) and high frequency (HF) components of an image are more degraded as compared to low and medium frequency components which are relatively well preserved. Hence, the loss of VHF and HF components contribute more towards the total degradation than the medium and low frequencies. Therefore, the restoration would be much effective, if it would be possible to restore the most degraded frequency components. So, the objective of iterative unsharp masking (IUM) is to extract the most degraded VHF sub-band of an image. As a result it can be used for efficient VHF restoration of the degraded image. The basic block diagram of HF restoration using IUM is illustrated in Fig. 4.3. Iterative unsharp masking (IUM) exploits the concept of signal decomposition through analysis filter bank so as to obtain the VHF sub-band of an image. The VHF component is superimposed on the sub-sampled image prior to up-sampling to reduce blurring in its up-scaled counterpart.

The Fig. 4.4 may be considered for illustrating how VHF can be extracted in a different manner to HF. Generally, HF corresponds to all the high frequency components. HF further consists of several high frequency sub-bands. VHF is actually a sub-band of HF which contains very high frequency contents. Initially, the input image is passed through an LPF to generate the blurred version of the input image which represents low frequency. The HF component is obtained by subtracting the blurred version of the image from its original. In the next iteration, the HF component is further passed through the LPF filter so that HL component or HL sub-band is obtained. Subsequently, it is by subtracting the HL sub-band from the high frequency, HH sub-band or very high frequency sub-band is obtained. This method can be further continued to the next level of signal decomposition to obtain still higher level of HF sub-band as explained in Fig. 4.5a.

In this algorithm, unsharp masking is used for signal decomposition which is similar to analysis filter bank. In wavelet domain signal decomposition, the analysis filter bank is used which employs low pass and high pass filters along with the down-samplers to obtain the image sub-bands at lower resolutions. However, in our proposed scheme, 2-D low pass filters along with 2-D subtractors are used to generate the sub-bands of same resolution as that of the input image. This criterion remains the same for all the levels of signal decomposition. Here, we aim to obtain the VHF extract of the original image without changing its resolution so that it can be superimposed on the original image prior to image up-sampling. Therefore, the down-samplers are not included in our signal decomposition scheme. Signal decomposition in the first and second stage using unsharp masking is depicted in the Fig. 4.4.

Since our intension is to extract only the VHF components, the HF components are only decomposed in the subsequent stages and at the same time LF components are discarded. This process continues for all the remaining stages of signal decomposition as illustrated in Fig. 4.5a.

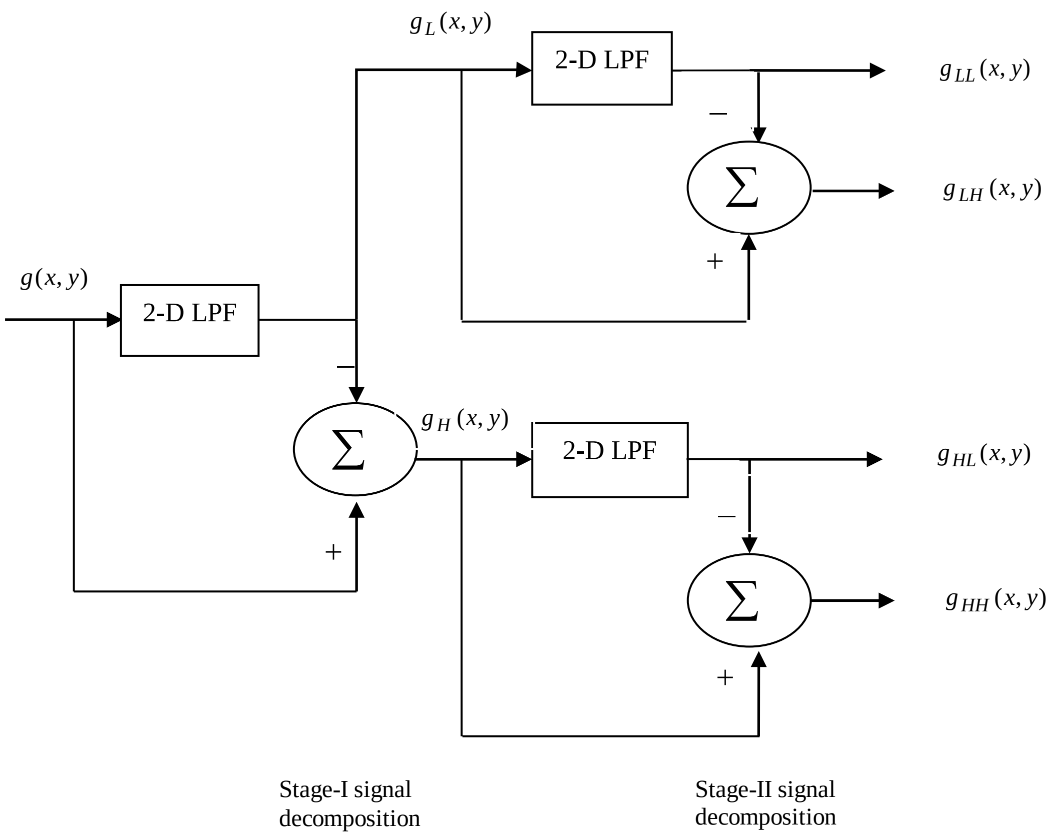


Fig. 4.4 Signal Decomposition using Unsharp Masking based Analysis Filter Bank

In the first stage of signal decomposition or at the first iteration, the low pass filtered output, $g_L(x, y)$ is subtracted from the original sub-sampled image, $g(x, y)$ to obtain the high pass filtered image output, $g_H(x, y)$. In our algorithm, we have used the weighted average LPF having point spread function, $h_{Lwa}(x, y)$ instead of averaging filter for low pass filtering operation. It is given by:

$$h_{Lwa}(x, y) = \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{4.5}$$

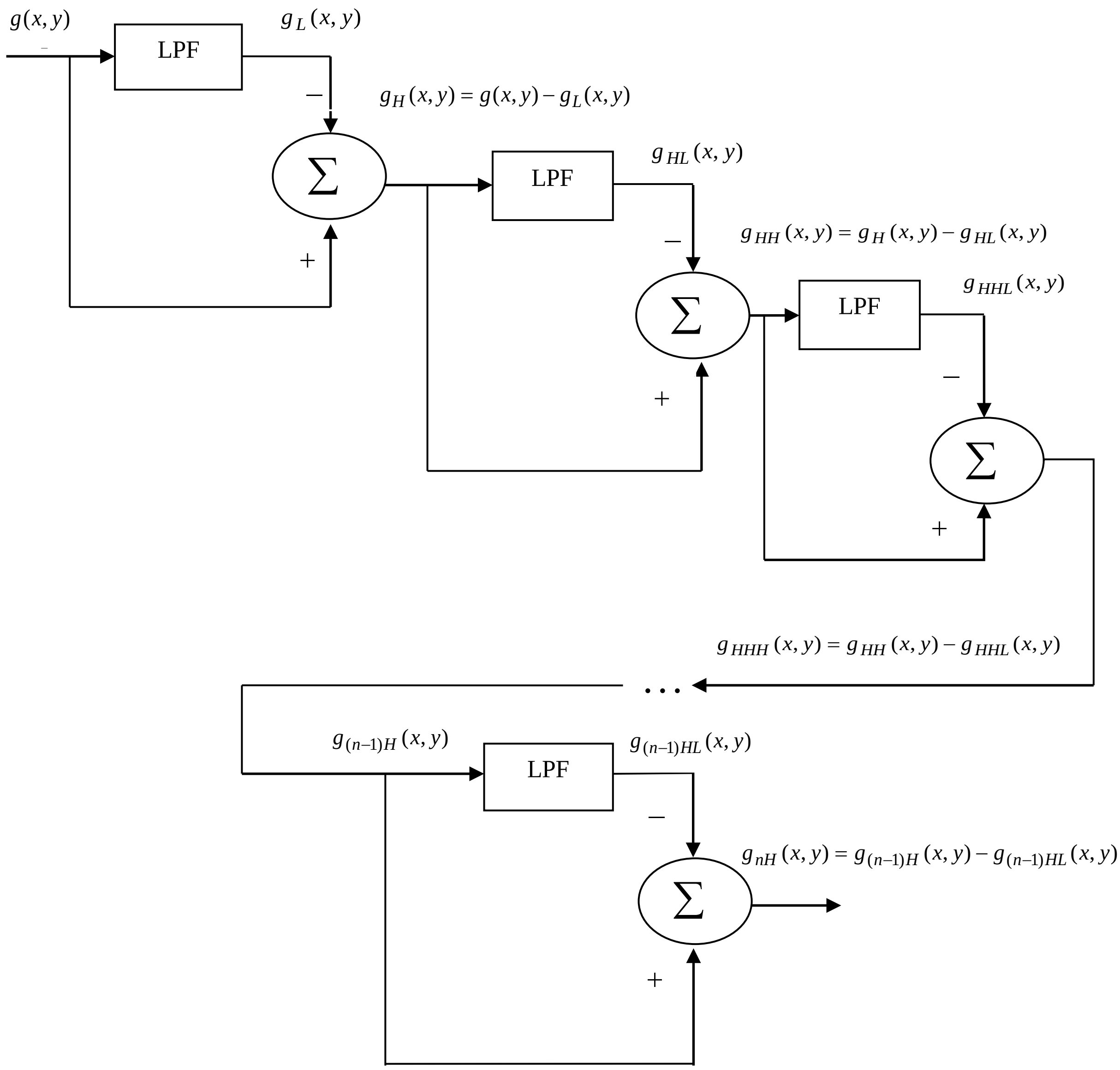


Fig. 4.5a Generation of Iterative Unsharp Mask after nth Iterations

We obtain $g_L(x, y)$ and $g_H(x, y)$ by the following operations.

$$g_L(x, y) = g(x, y) * h_{Lwa}(x, y) \tag{4.6}$$

$$g_H(x, y) = g(x, y) - g_L(x, y) \tag{4.7}$$

where $*$ represents 2-D convolution.

After the first stage, the low frequency sub-band, $g_L(x, y)$ is discarded and only the high frequency sub-band, $g_H(x, y)$ is forwarded to the next level of signal decomposition. $g_H(x, y)$ is also called as the unshap mask after first iteration. In the second level of signal decomposition, the corresponding sub-bands of $g_H(x, y)$ are generated and are given by:

$$g_{HL}(x, y) = g_H(x, y) * h_{Lwa}(x, y) \tag{4.8}$$

$$g_{HH}(x, y) = g_H(x, y) - g_{HL}(x, y) \tag{4.9}$$

The low frequency sub-band, $g_{HL}(x, y)$ is discarded and the HF sub-band, $g_{HH}(x, y)$ which is the unsharp mask after second iteration is forwarded to the next level of signal decomposition. This process continues for several stages of signal decomposition to arrive at the final unsharp mask, $g_{nH}(x, y)$ of desired VHF sub-band.

The number of iterations represent the number of stages of signal decomposition is kept seven for arriving at the desired VHF sub-band for optimum performance. The number of iteration of IUM algorithm and is determined experimentally and is discussed in the next section. The final unsharp mask is then superimposed on the original image to obtain the sharpened image prior to up-scaling and is given by,

$$g_S(x, y) = g(x, y) + K\ g_{nH}(x, y) \tag{4.10}$$

where, $n = 7$ and $K = 1$ are taken for optimum performance of the algorithm. The sharpen image, $g_S(x, y)$ is finally up-scaled using Lanczos-3 interpolation to obtain the HF restored image, $\hat{f}(x, y)$ using (2.3).

The spectrum of the HF extract for different images after 7 iterations is shown in Fig. 4.5b. The figure reveals that there is a considerable attenuation of low frequency spectral components and preservation of high frequency spectral components. This shows that the HF extract is a result of high-pass filtering operation through the employment of IUM. Fig 4.5c shows the spectral comparison of original, low-resolution, Lanczos-3 and IUM up-scaled image of Lena. In addition, Fig 4.5d shows the comparison of amplitude spectra of various Lanczos-3 and IUM up-scaled images. A scaling factor of 5 and logarithm transformation are employed for proper visibility of the spectrums. The left column of Fig 4.5d represents the spectra of Lanczos-3 up-scaled images and the middle column represents the IUM output spectra of the corresponding images. The right column represents their spectral difference. The IUM spectra of different images show an enhancement in HF spectral components over Lanczos-3. The more pronounced HF spectrum is due to the superimposition of very high frequency extract obtained through the iterative scheme.

The number of iterations, n and the weight factor, K play a major role in the performance of the algorithm. Precise estimation of the number of iterations and weight factor is very essential for overall better performance of the algorithm and is done on experimental basis. The estimation of the number of iterations and weight factor are explained in the subsequent section.

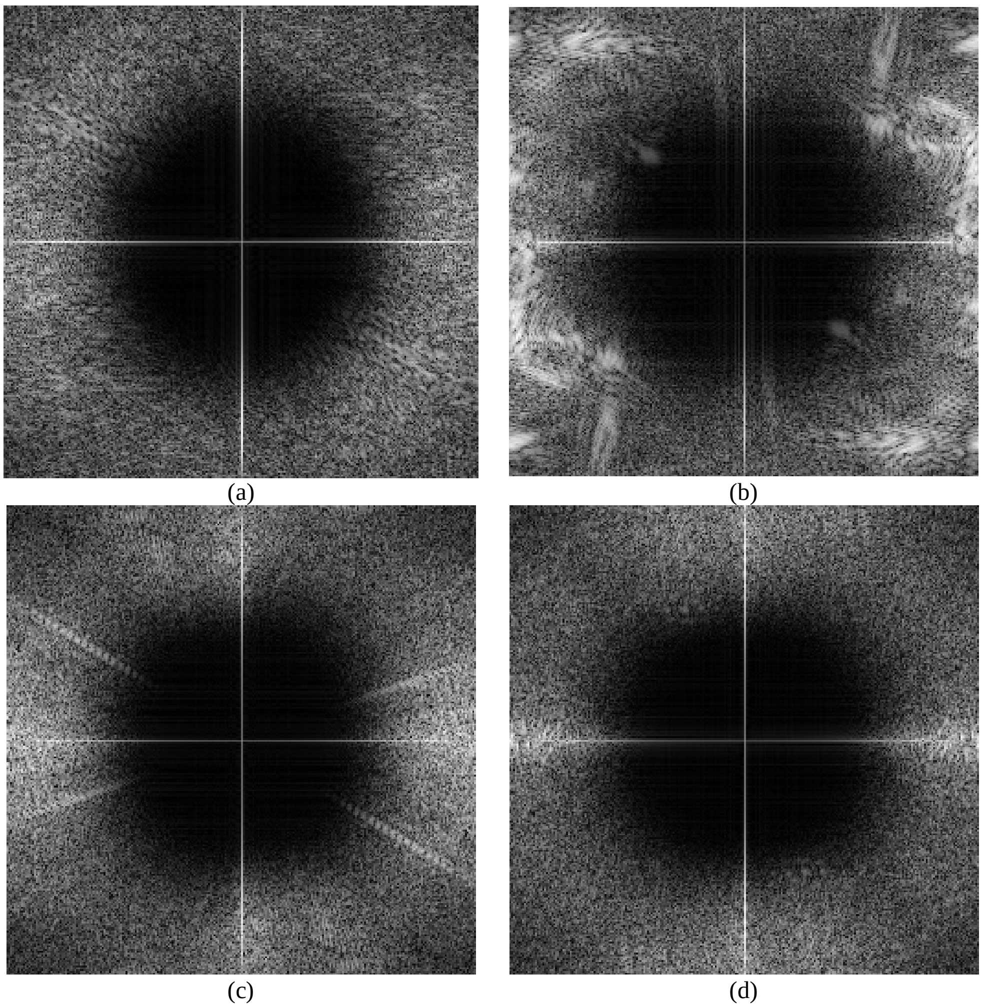


Fig. 4.5b Spectrum of HF extract obtained after 7^{th} iteration corresponding to different (512×512) images: (a) Lena; (b) Barbara; (c) Boat; (d) Goldhill

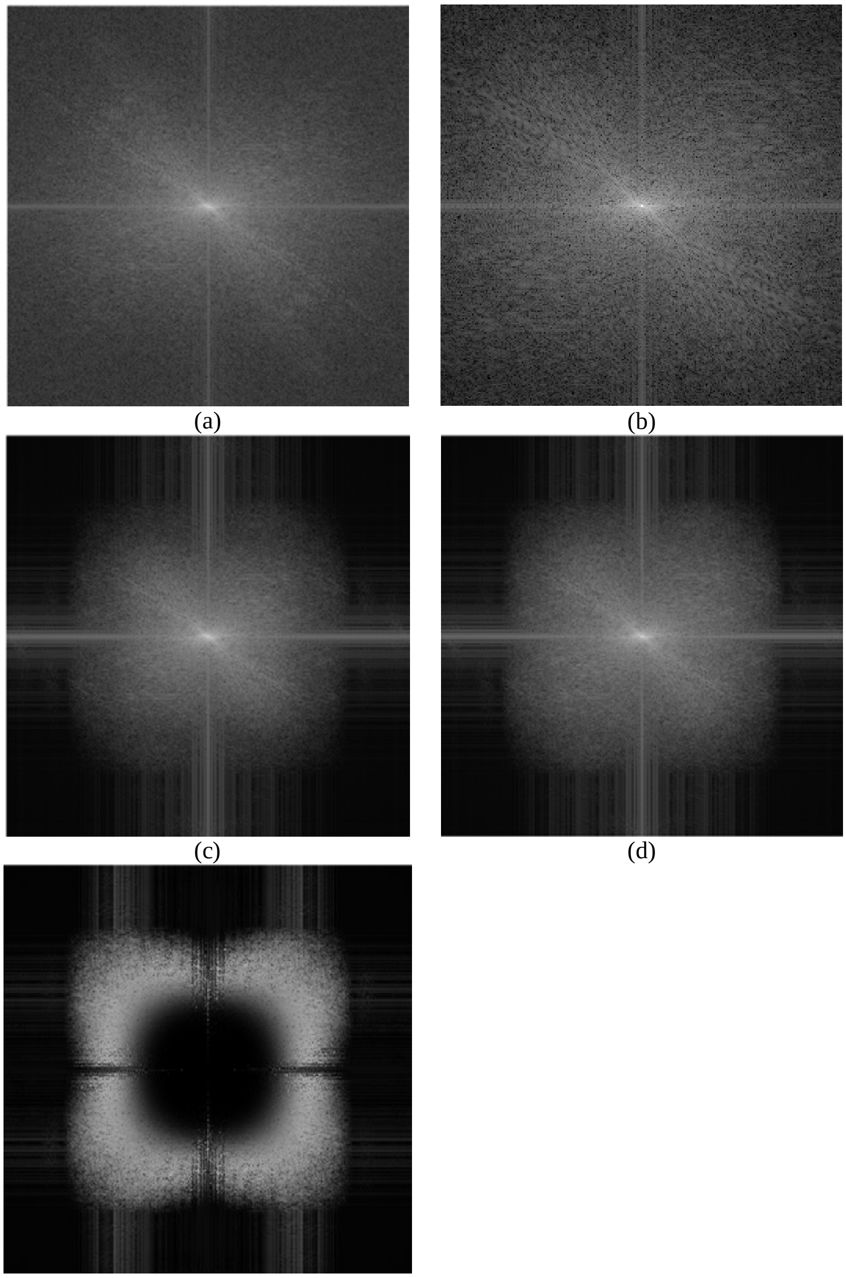


Fig. 4.5c Spectrum of Lena (512×512) image: (a) Original; (b) Low-resolution; (c) Lanczos-3 upscaled; (d) IUM up-scaled; (e) Spectral difference between Lanczos-3 and IUM up-scaled images

4.2.1 Estimation of Number of Iterations and Weight Factor

The number of iterations determines the level of signal decomposition. The bandwidth of the sub-bands reduces towards higher order iterations and consequently more sub-bands are generated. To determine the appropriate HF sub-band that achieves maximum restoration performance, simulation studies are carried out for different images to observe the variation of PSNR (dB) gain with respect to the weight factor, K for different iterations as shown in Fig. 4.6 and Fig 4.7 respectively. It may be observed from the figures that the PSNR gain gradually increases towards higher order iterations and beyond certain iterations it declines. In case of most of the images, the performance of the IUM algorithm declines beyond 7th iteration. The number of iterations and the weight factor corresponding to global maximum of the characteristic plot are determined for six different images as illustrated in Table 4.2. The generalized weight factor and the number of iterations are determined by taking the mean of the weight factors and iterations corresponding to the global maximum PSNR for six different images as shown in Table 4.2. The weight factor, K and number of iterations, n are estimated as one and seven, respectively for improved performance.

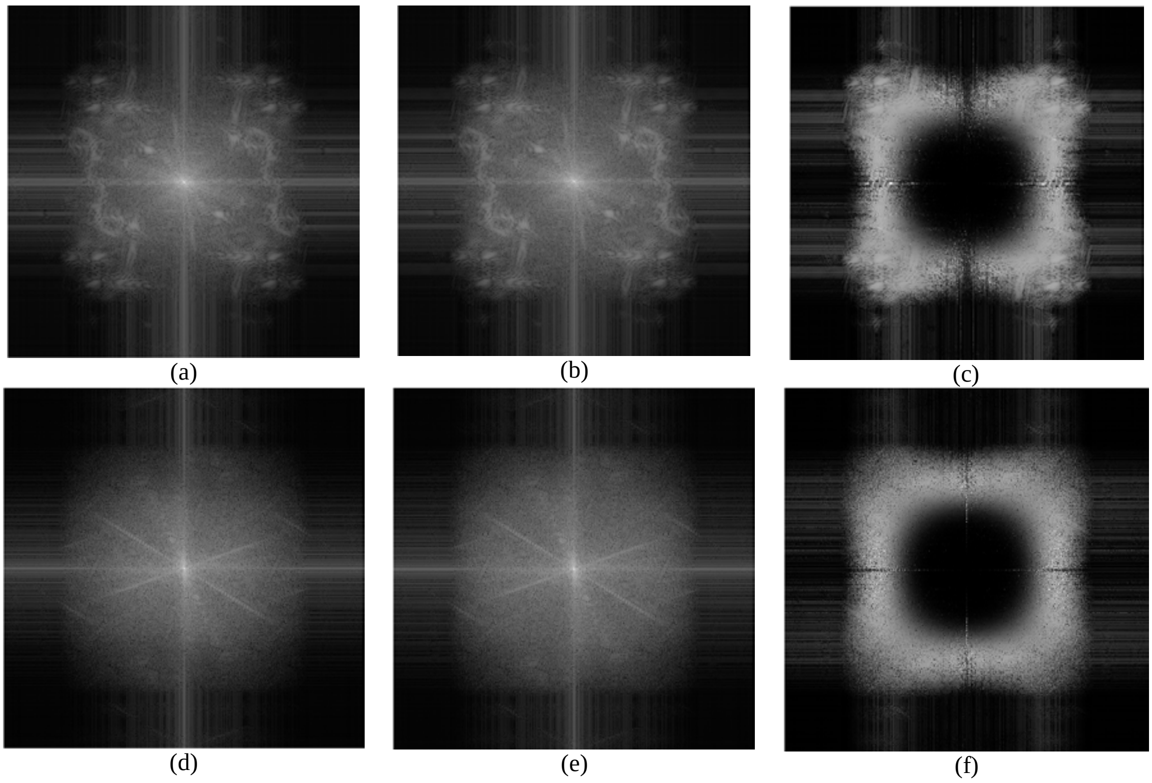


Fig. 4.5d Spectra of different Lanczos-3, IUM up-scaled images and their spectral difference: (a) Barbara_Lanczos-3; (b) Barbara_IUM; (c) Spectral difference for Barbara image; (d) Boat_Lanczos-3; (e) Boat_IUM; (f) Spectral difference for Boat image

Table 4.2 Estimation of weight factor, K and the number of iterations for the IUM scheme

Parameter	Weight factor and iteration corresponding to the maximum PSNR for different images						Average
	Lena	Boat	Barbara	Goldhill	Living room	Cameraman	
Weight Factor	1.1	0.98	1.1	0.98	1	0.9	$1.01 \approx 1$
Iterations	8	6	8	7	6	6	$6.83 \approx 7$

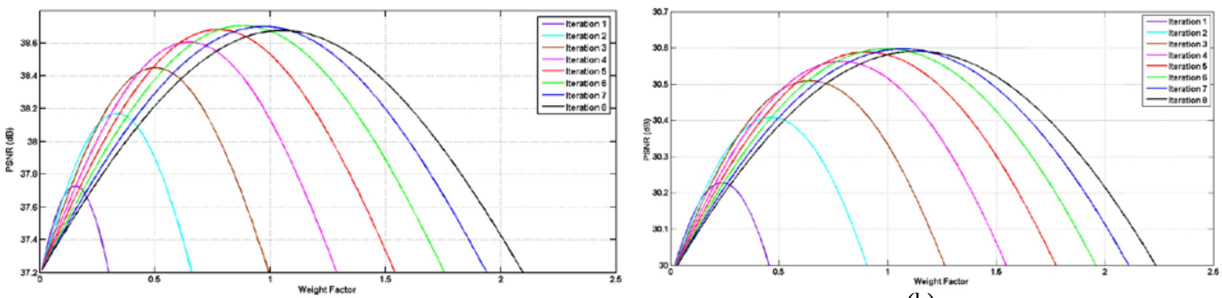
Table 4.3 Weight factor deviation, ΔK among various images corresponding to different iterations of the proposed IUM scheme

Image	**Weight factors corresponding to maximum PSNR for different iteration**							
	Iteration 1	**Iteration 2**	**Iteration 3**	**Iteration 4**	**Iteration 5**	**Iteration 6**	**Iteration 7**	**Iteration 8**
Lena	0.16	0.37	0.55	0.62	0.76	0.9	1	1.1
Barbara	0.3	0.6	0.75	0.8	0.9	1	1.1	1.1
Boat	0.23	0.42	0.61	0.75	0.88	0.98	1.05	1.1
Goldhill	0.19	0.35	0.6	0.7	0.8	0.9	0.98	1.04
Room	0.24	0.45	0.65	0.78	0.9	1	1.06	1.15
Cameraman	0.15	0.3	0.5	0.65	0.76	0.87	0.98	1.05
ΔK	0.15	0.3	0.25	0.18	0.14	0.13	0.12	0.11

The weight factor deviation, ΔK in a particular iteration is the difference between its maximum and minimum value of weight factor amongst different images. Mathematically, it is given by,

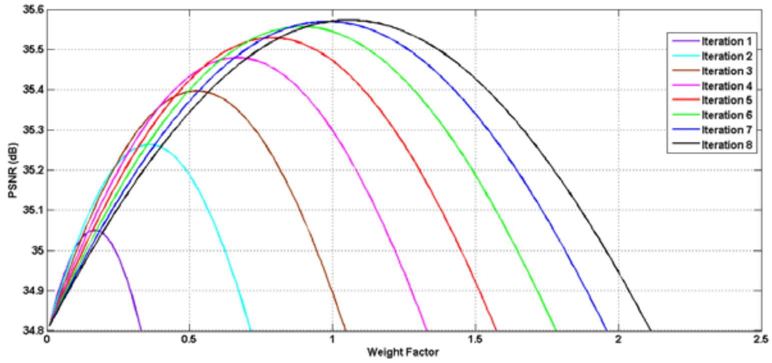
$$\Delta K = K_{\max} - K_{\min} \tag{4.11}$$

It may be well observed from Table 4.3 that the weight factor deviation, ΔK is more toward the lower order iterations. Since there is much deviation in weight factors at the lower order iterations, the performance of the algorithm deteriorates by using a single average weight factor for various images. However, the weight factor deviation, ΔK converges more towards the higher order iterations. Hence, the estimated weight factor becomes more precise and versatile to provide more improved performance.

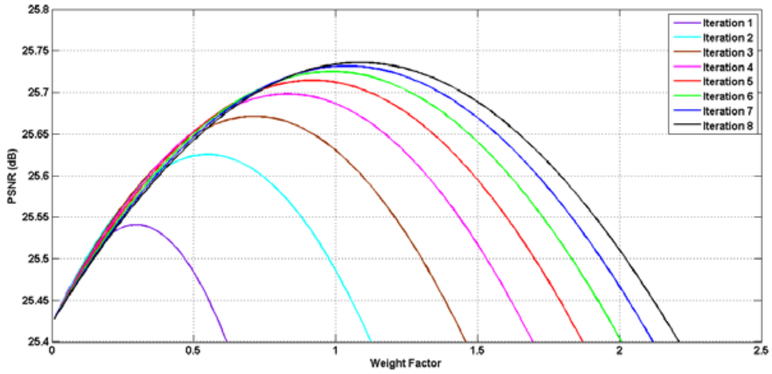


(a) (b)

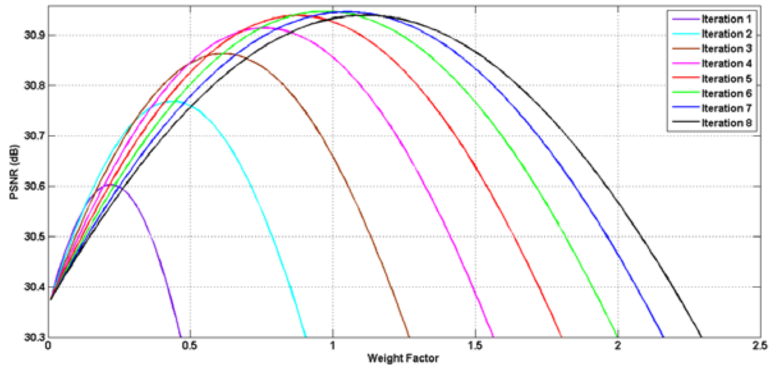
Fig. 4.6 PSNR vs. weight factor plots of various images at different iterations using IUM algorithms at 4:1 compression ratio: (a) Cameraman; (b) Living room



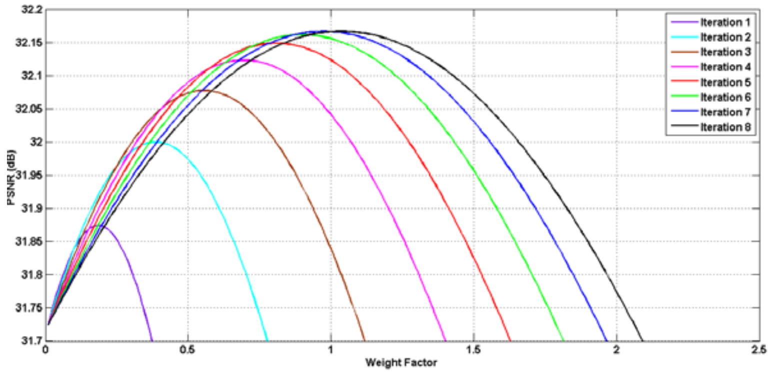
(a)



(b)

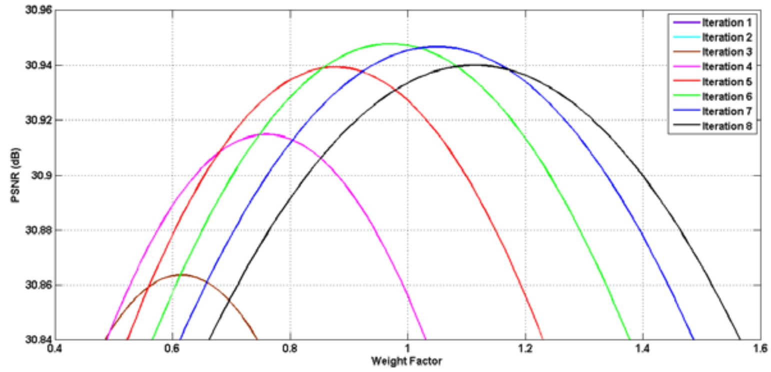


(c)

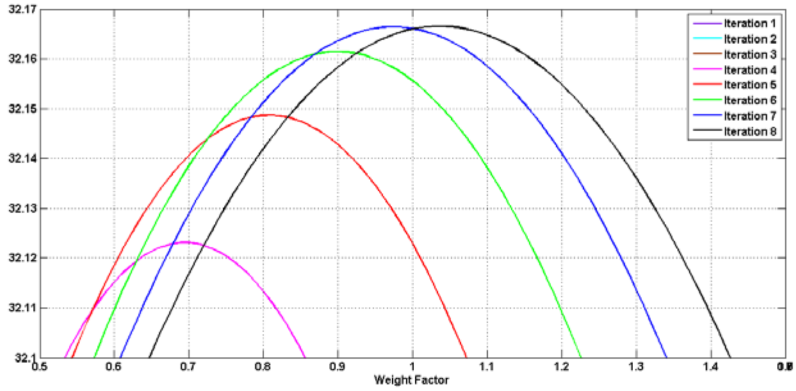


(d)

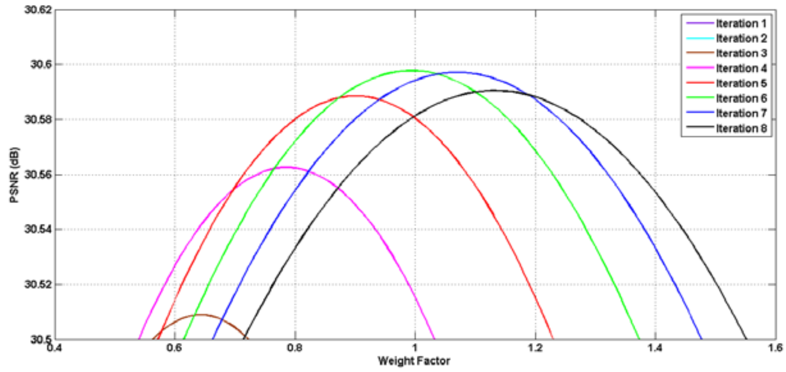
Fig. 4.7 PSNR vs. weight factor plots of various images at different iterations using IUM algorithms at 4:1 compression ratio: (a) Lena; (b) Barbara; (c) Boat; (d) Goldhill



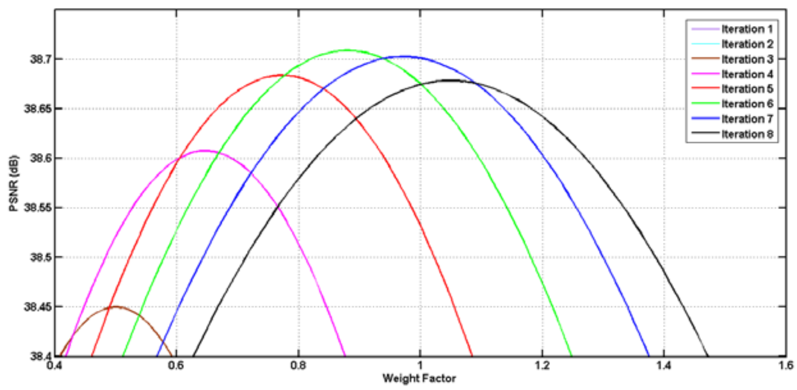
(a)



(b)



(c)



(d)

Fig. 4.8 Enlarged version of PSNR vs. weight factor plots at different iterations using IUM algorithm for different images at 4:1 compression ratio: (a) Boat; (b) Goldhill; (c) Living room; (d) Cameraman

4.2.2 IUM Algorithm

Let $g(x,y)$ be the sub-sampled, low resolution input image. Let $g_{7H}(x,y)$ be the iterative unsharp mask after 7 iterations and $g_S(x,y)$ be the corresponding sharpened image. Let them of size $(P \times Q)$. The original image is of size $(2P \times 2Q)$. $h_{Lwa}(x,y)$ denotes the weighted average blurring kernel, used in iterative unsharp masking. The iterative unsharp masking (IUM) algorithm is given below.

Step-1. Select a (3×3) window, w in a sub-sampled image, $g(x,y)$.

$$w_{s,t}(x,y), \quad -1 \le s,t \le 1$$

Step-2. Obtain the blurred, low pass filtered image $g_L(x,y)$ by linearly convolving $h_{Lwa}(x,y)$ with $g(x,y)$.

$$g_L(x,y) = \sum_{s=-1}^{1}\sum_{i=-1}^{1} h_{Lwa}(s,t)\, g(x+s, y+t) \tag{4.12}$$

Step-3. Repeat step-1 and step-2 for all (x,y) locations of $g(x,y)$ to obtain the filtered image, $g_L(x,y)$.

Step-4. Subtract the low pass filtered image $g_L(x,y)$ from the original image, $g(x,y)$ to generate the unsharp mask, $g_H(x,y)$ for the first iteration.

$$g_H(x,y) = g(x,y) - g_L(x,y) \tag{4.13}$$

Step-5. Assign $g(x,y)$ as $g_H(x,y)$.

$$g(x,y) = g_H(x,y) \tag{4.14}$$

Step-6. Repeat step-1 to step-5 for six times to obtain the iterative unsharp mask, $g_{7H}(x,y)$.

Step-7. Add the iterative unsharp mask $g_{7H}(x,y)$ with the original sub-sampled image, $g(x,y)$ to obtain the sharpened image, $g_S(x,y)$.

$$g_S(x,y) = g(x,y) + K\, g_{7H}(x,y) \tag{4.15}$$

where, $K = 1$

Step-8. The sharpened image, $g_S(x,y)$ is finally up-sampled by Lanczos-3 interpolation using (2.3) to obtain the up-scaled image, $\hat{f}(x,y)$.

An error based up-sampling (EU) scheme is presented in the next section which is based on inverse modeling approach of image degradation.

4.3 Error based Up-sampling (EU) Scheme [P6]

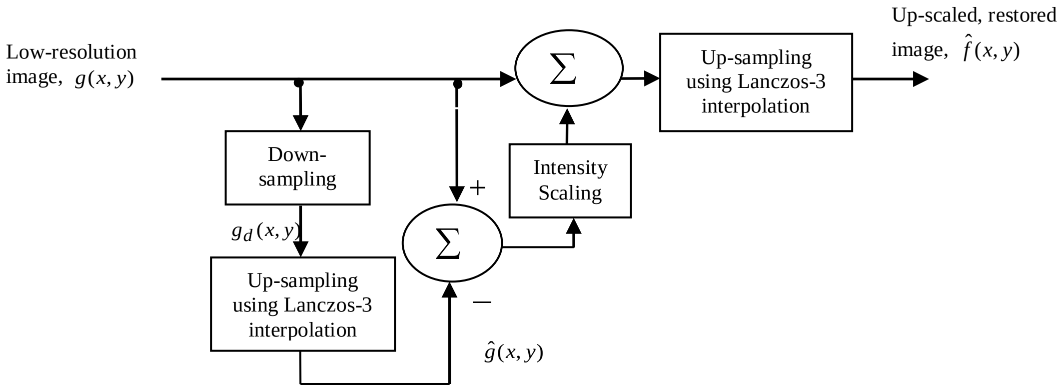


Fig. 4.9 Block diagram of the proposed error based up-sampling scheme

The proposed error based up-sampling scheme estimates the high frequency degradation due to Lanczos-3 up-sampling scheme and performs the inverse operation on the sub-sampled image so that the high frequency contents will be restored in its up-sampled counterpart. The proposed scheme extracts the degraded high frequency information and superimposes it on the original sub-sampled image to lessen the blurring in its up-scaled image. The degraded high frequency information is obtained by purposefully down-sampling the given low resolution image at 4:1 compression ratio and then rescaling back to its original size using Lanczos-3 interpolation. The lost high frequency information or the prediction error is obtained by subtracting the restored image from the original. The prediction error contains all the high frequency details that are lost during the up-sampling process. Hence, by the weighted superimposition of those lost high frequency details with the original image, the degree of blurring can be considerably reduced. The error based up-sampling scheme is illustrated in Fig. 4.9.

The precise estimation of weight factor is very essential for overall performance of the algorithm. To determine the overall weight factor, the performance characteristics of PSNR (dB) vs. weight factor for different images are obtained as illustrated in Fig. 4.10. The weight factor corresponding to maximum PSNR is determined for various images. The overall weight factor is obtained by averaging those weight factors as illustrated in Table 4.4.

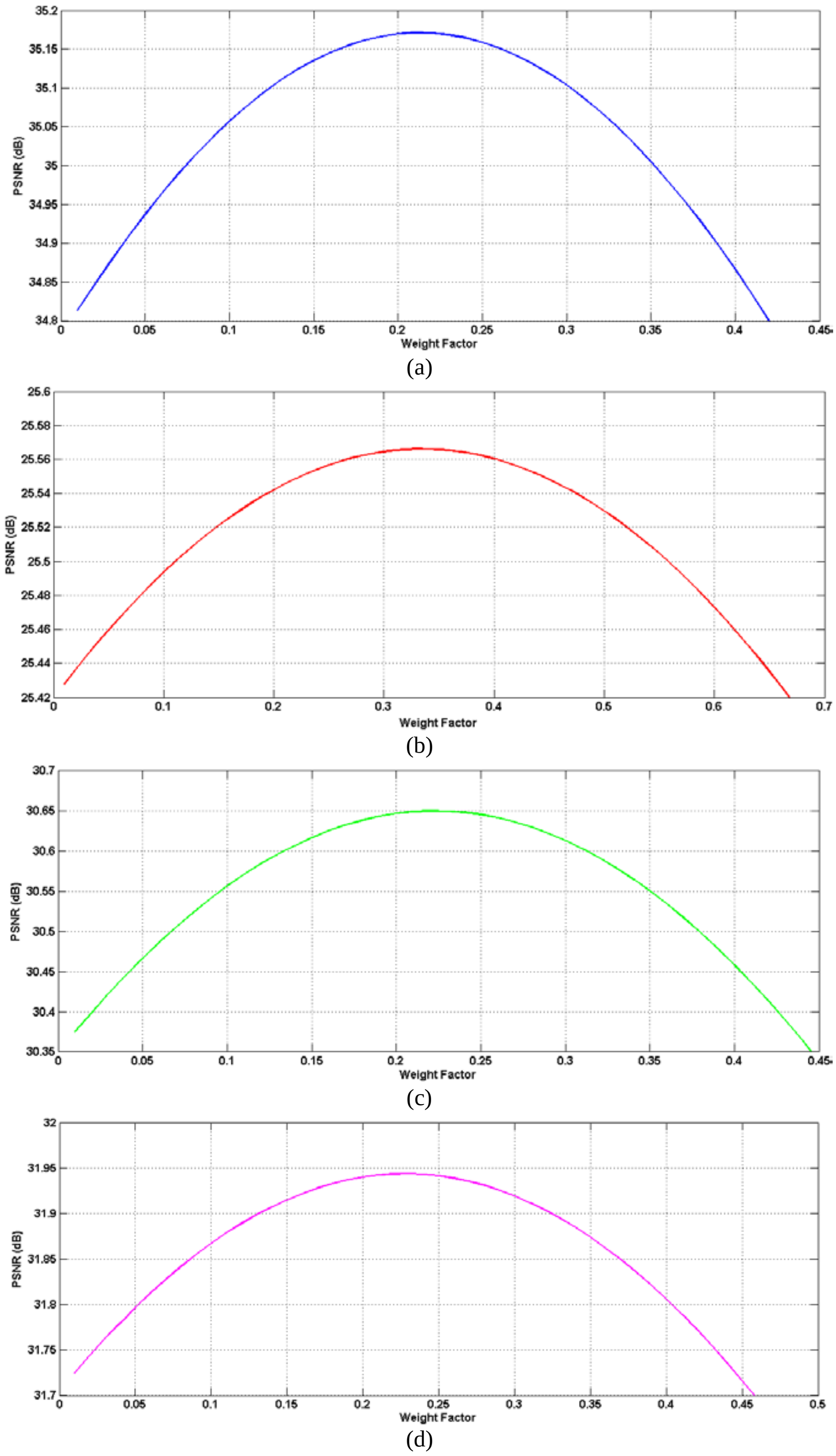


Fig. 4.10 PSNR (dB) vs. weight factor plots of different images using the proposed EU algorithm at 4:1 compression ratio: (a) Lena; (b) Barbara; (c) Boat; (d) Goldhill

Table 4.4 Weight factor, K estimation of the proposed pre-processing schemes

Algorithm	Weight factors corresponding to the maximum PSNR for different images				Average Weight Factor
	Lena	Boat	Barbara	Goldhill	
EU	0.220	0.225	0.330	0.225	0.25

4.3.1 EU Algorithm

Let $g(x,y)$ be the sub-sampled image of size $(P\times Q)$. $g_d(x,y)$ denotes the down-sampled version of $g(x,y)$ at 4:1 compression ratio. The down-sampled image, $g_d(x,y)$ is rescaled back to the original size $(P\times Q)$ using Lanczos-3 interpolation and is denoted by $\hat{g}(x,y)$. The EU algorithm is given below.

Step-1. Obtain the down-sampled image, $g_d(x,y)$ by sub-sampling $g(x,y)$ at 4:1 compression ratio.

Step-2. Generate the up-scaled image, $\hat{g}(x,y)$ from $g_d(x,y)$ by Lanczos-3 interpolation using (2.3).

Step-3. Generate the HF extract, $g_e(x,y)$ by subtracting $\hat{g}(x,y)$ from $g(x,y)$.

$$g_e(x,y) = g(x,y) - \hat{g}(x,y) \tag{4.16}$$

Step-4. Generate the sharpened image, $g_S(x,y)$ by weighted superimposition of $g_H(x,y)$ with $g(x,y)$.

$$g_S(x,y) = g(x,y) + K\, g_e(x,y) \tag{4.17}$$

where $K = 0.25$

Step-5. The sharpened image, $g_S(x,y)$ is finally up-sampled by Lanczos-3 interpolation to generate the up-scaled, restored image, $\hat{f}(x,y)$ of size using (2.3).

4.4 Experiment and Simulation

To evaluate the HF restoration performance of the proposed algorithms, different images are down-sampled at 4:1 compression ratio (CR). Afterward, the down-sampled images are re-scaled to their original size using the proposed algorithms for comparison with the original images. Peak-signal-to-noise-ratio (PSNR) in dB and universal quality index (UQI) are measured to determine the objective performance of the proposed algorithms. PSNR and UQI are also measured at 16:1 compression ratio to examine their performance under varying conditions. The algorithms are also tested for various video sequences.

Lena (512×512) image, which has different regions of low, medium and high frequencies, is a suitable candidate for subjective evaluation of the proposed algorithms. The low, medium, high frequency regions of Lena image are examined separately for performance evaluation of the algorithms. The region rich in combination of different patterns is also considered for performance evaluation. Four distinct regions with different features and thus different signal characteristics are marked and performance at these regions is distinctly analyzed. For this purpose, the output images at these regions are enlarged for human interpretation.

Error image is a quality metric which is used to measure the restoration performance of various algorithms. Reduced error indicates better restoration performance of the algorithms. In addition, the computational complexities in terms of CPU execution time of the proposed algorithms are computed and are compared with those for the existing algorithms to determine their feasibility for real-time applications.

The figures and tables showing the performance of the existing and proposed algorithms are explained below.

Fig. 4.11 shows the PSNR comparisons of various existing and proposed schemes for various video sequences over 50 frames at 4:1 compression ratio. Fig. 4.12, Fig. 4.17 and Fig. 4.18 show the subjective performance of Lena (512×512), Boat (512×512) and Goldhill (512×512) images, respectively using various up-sampling schemes at 4:1 compression ratio. In case of Fig. 4.12, four distinct regions with different features and thus different signal characteristics such as low, medium, high and their combinations are marked. Performance at these regions is analyzed. For this purpose, the output images at these regions are enlarged and shown in Fig. 4.13, Fig. 4.14, Fig. 4.15 and Fig. 4.16. The error images of Lena (512×512) corresponding to various schemes are given in Fig. 4.19.

Table 4.5 and Table 4.6 show the PSNR and UQI comparison of different existing and proposed schemes at 4:1 and 16:1 compression ratio, respectively. Likewise, Table 4.7 represents the average PSNR and UQI comparison of different schemes at 4:1 compression ratio, meant for various video sequences. PSNR and UQI values in bold letter represent the peak performance. Table 4.8 shows the execution time of various existing and proposed schemes at 4:1 compression ratio corresponding to images of different sizes.

Table 4.5 PSNR (dB) and UQI comparison of different schemes at 4:1 compression ratio for various (512×512) images

Image	Image Metric	Bilinear [5]	Bicubic [4]	Lanczos3 [12]	DCT [74]	DASR [116]	DSWD [82]	UMP [P4]	IUM [P5]	EU [P6]
Mandril	PSNR	23.045	23.630	23.859	23.925	22.726	23.025	24.030	**24.296**	24.073
	UQI	0.8957	0.9114	0.9170	0.9187	0.8968	0.9031	0.9219	**0.9271**	0.9232
Lena	PSNR	32.704	34.148	34.813	35.023	31.220	31.345	34.993	**35.569**	35.151
	UQI	0.9922	0.9945	0.9953	0.9955	0.9894	0.9897	0.9955	**0.9961**	0.9957
Barbara	PSNR	24.925	25.352	25.428	25.183	24.764	24.837	25.517	**25.731**	25.559
	UQI	0.9631	0.9669	0.9675	0.9657	0.9627	0.9633	0.9685	**0.9700**	0.9687
Boat	PSNR	28.940	29.952	30.375	30.466	27.865	28.446	30.591	**30.946**	30.642
	UQI	0.9801	0.9845	0.9860	0.9863	0.9756	0.9786	0.9868	**0.9878**	0.9870
Goldhill	PSNR	30.574	31.405	31.725	31.716	29.609	29.829	31.855	**32.166**	31.939
	UQI	0.9880	0.9901	0.9909	0.9909	0.9853	0.9860	0.9912	**0.9918**	0.9914
Pirate	PSNR	30.027	31.058	31.490	31.606	28.955	29.391	31.690	**32.103**	31.765
	UQI	0.9853	0.9885	0.9897	0.9899	0.9818	0.9835	0.9902	**0.9911**	0.9904
Livingroom	PSNR	28.617	29.557	29.977	30.128	27.459	28.174	30.216	**30.595**	30.277
	UQI	0.9761	0.9811	0.9829	0.9835	0.9702	0.9747	0.9841	**0.9853**	0.9843
Fingerprint	PSNR	28.045	30.632	31.722	32.133	26.378	26.411	31.787	**32.903**	31.932
	UQI	0.9785	0.9889	0.9915	0.9922	0.9724	0.9727	0.9919	**0.9936**	0.9922
Baboon	PSNR	33.588	35.014	35.662	35.890	32.145	32.373	35.794	**36.315**	35.961
	UQI	0.9924	0.9946	0.9954	0.9956	0.9897	0.9902	0.9955	**0.9960**	0.9957
Bridge	PSNR	25.728	26.504	26.826	26.919	24.910	25.348	27.029	**27.348**	27.065
	UQI	0.9693	0.9748	0.9767	0.9772	0.9646	0.9680	0.9781	**0.9795**	0.9783
Cameraman	PSNR	33.214	35.757	37.216	37.832	30.972	31.854	37.573	**38.700**	37.720
	UQI	0.9959	0.9977	0.9984	0.9986	0.9933	0.9945	0.9985	**0.9989**	0.9986
Cat	PSNR	30.949	31.982	32.427	32.563	29.992	30.464	32.668	**33.073**	32.709
	UQI	0.9920	0.9937	0.9943	0.9945	0.9902	0.9912	0.9947	**0.9951**	0.9947
Crowd	PSNR	30.984	32.666	33.451	33.768	29.277	29.582	33.792	**34.467**	33.833
	UQI	0.9894	0.9930	0.9942	0.9946	0.9850	0.9860	0.9947	**0.9954**	0.9947
Cycle	PSNR	21.208	21.895	22.154	22.129	20.693	21.173	22.325	**22.631**	22.372
	UQI	0.9323	0.9437	0.9475	0.9474	0.9290	0.9363	0.9506	**0.9539**	0.9512
F16	PSNR	30.379	31.543	32.104	**32.722**	29.689	29.830	32.199	32.653	32.425
	UQI	0.9848	0.9885	0.9900	**0.9913**	0.9829	0.9835	0.9903	0.9912	0.9908
House	PSNR	29.248	30.314	30.807	30.862	28.049	28.790	31.157	**31.628**	31.197
	UQI	0.9719	0.9785	0.9809	0.9812	0.9651	0.9705	0.9827	**0.9844**	0.9829
Lake	PSNR	28.945	30.022	30.495	30.793	27.997	28.549	30.685	**31.069**	30.751
	UQI	0.9895	0.9919	0.9928	0.9933	0.9873	0.9888	0.9931	**0.9937**	0.9932
Peppers	PSNR	31.180	31.991	32.328	**32.747**	30.451	30.795	32.419	32.727	32.545
	UQI	0.9923	0.9937	0.9942	**0.9947**	0.9911	0.9918	0.9943	0.9947	0.9945
Elaine	PSNR	32.534	33.117	33.309	33.284	31.617	31.519	33.268	**33.478**	33.406
	UQI	0.9913	0.9924	0.9928	0.9927	0.9894	0.9892	0.9928	**0.9931**	0.9930
Ruler	PSNR	12.335	12.613	12.673	12.600	14.458	**15.638**	12.781	12.911	12.789
	UQI	0.5188	0.5735	0.5898	0.5920	0.7650	**0.8156**	0.6189	0.6395	0.6307

Table 4.6 PSNR (dB) and UQI comparison of different schemes at 16:1 compression ratio for various (512×512) images

Image	**Image Metric**	**Bilinear [5]**	**Bicubic [4]**	**Lanczos3 [12]**	**DCT [74]**	**UMP [P4]**	**IUM [P5]**	**EU [P6]**
Mandril	PSNR(dB)	20.883	21.085	21.156	21.167	21.205	**21.301**	21.228
	UQI	0.8191	0.8309	0.8351	0.8362	0.8398	**0.8439**	0.8413
Lena	PSNR(dB)	28.053	28.848	29.183	29.296	29.291	**29.622**	29.383
	UQI	0.9767	0.9810	0.9825	0.9829	0.9832	**0.9843**	0.9835
Barbara	PSNR(dB)	23.351	23.607	23.708	23.747	23.741	**23.844**	23.765
	UQI	0.9457	0.9496	0.9510	0.9515	0.9520	**0.9528**	0.9522
Boat	PSNR(dB)	25.041	25.538	25.739	25.773	25.857	**26.069**	25.911
	UQI	0.9493	0.9557	0.9580	0.9584	0.9598	**0.9616**	0.9603
Goldhill	PSNR(dB)	27.166	27.628	27.798	27.776	27.851	**28.041**	27.920
	UQI	0.9731	0.9761	0.9771	0.9770	0.9777	**0.9785**	0.9779
Pirate	PSNR(dB)	26.286	26.861	27.083	27.127	27.186	**27.418**	27.236
	UQI	0.9641	0.9691	0.9708	0.9712	0.9720	**0.9733**	0.9723
Livingroom	PSNR(dB)	24.932	25.385	25.560	25.563	25.667	**25.857**	25.712
	UQI	0.9420	0.9488	0.9511	0.9513	0.9533	**0.9550**	0.9538
Fingerprint	PSNR(dB)	20.920	22.633	23.779	24.202	24.435	**25.018**	24.484
	UQI	0.8791	0.9192	0.9407	0.9467	0.9523	**0.9591**	0.9544
Baboon	PSNR(dB)	28.812	29.604	29.931	30.007	30.056	**30.412**	30.155
	UQI	0.9766	0.9808	0.9823	0.9826	0.9831	**0.9843**	0.9834
Bridge	PSNR(dB)	22.676	23.066	23.230	23.263	23.328	**23.486**	23.353
	UQI	0.9357	0.9423	0.9449	0.9454	0.9470	**0.9487**	0.9474
Cameraman	PSNR(dB)	26.633	27.546	27.946	28.094	28.168	**28.604**	28.264
	UQI	0.9809	0.9847	0.9861	0.9866	0.9870	**0.9882**	0.9873
Cat	PSNR(dB)	27.295	27.838	28.059	28.105	28.193	**28.429**	28.229
	UQI	0.9812	0.9835	0.9844	0.9845	0.9849	**0.9857**	0.9851
Crowd	PSNR(dB)	25.759	26.601	26.974	27.066	27.168	**27.514**	27.207
	UQI	0.9634	0.9706	0.9733	0.9739	0.9750	**0.9767**	0.9752
Cycle	PSNR(dB)	18.693	18.996	19.109	19.135	19.177	**19.317**	19.205
	UQI	0.8727	0.8842	0.8883	0.8893	0.8927	**0.8958**	0.8934
F16	PSNR(dB)	25.767	26.457	26.773	26.939	26.858	**27.146**	26.972
	UQI	0.9540	0.9617	0.9647	0.9661	0.9662	**0.9681**	0.9671
House	PSNR(dB)	25.401	25.863	26.050	26.063	26.172	**26.331**	26.173
	UQI	0.9285	0.9373	0.9406	0.9409	0.9435	**0.9452**	0.9438
Lake	PSNR(dB)	24.783	25.466	25.746	25.834	25.893	**26.166**	25.932
	UQI	0.9720	0.9764	0.9780	0.9785	0.9791	**0.9802**	0.9793
Peppers	PSNR(dB)	27.359	28.045	28.330	28.524	28.395	**28.641**	28.470
	UQI	0.9811	0.9841	0.9852	0.9859	0.9856	**0.9863**	0.9858
Elaine	PSNR(dB)	29.771	30.496	30.785	30.884	30.758	**31.024**	30.903
	UQI	0.9832	0.9860	0.9870	0.9878	0.9871	**0.9877**	0.9875
Ruler	PSNR(dB)	10.707	10.804	10.873	10.876	10.888	**10.906**	10.887
	UQI	0.2036	0.2394	0.2608	0.2636	**0.2796**	0.2789	0.2986

Table 4.7 Average PSNR (dB) and UQI comparison of different interpolation techniques at 4:1 compression ratio for various sequences over 50 frames

Sequence	**Image Metric**	**Bilinear [5]**	**Bicubic [4]**	**Lanczos3 [12]**	**DCT [74]**	**UMP [P4]**	**IUM [P5]**	**EU [P6]**
Salesman	PSNR(dB)	28.107	28.979	29.327	29.440	29.606	**29.972**	29.631
	UQI	0.9649	0.9719	0.9742	0.9750	0.9762	**0.9781**	0.9764
Bus	PSNR(dB)	24.243	25.262	25.716	25.794	26.085	**26.588**	26.124
	UQI	0.9491	0.9610	0.9652	0.9660	0.9688	**0.9722**	0.9693
Akiyo	PSNR(dB)	31.686	32.911	33.450	33.647	33.734	**34.298**	33.843
	UQI	0.9927	0.9945	0.9952	0.9954	0.9955	**0.9960**	0.9956
City	PSNR(dB)	26.827	27.592	27.879	27.852	28.101	**28.475**	28.173
	UQI	0.9104	0.9277	0.9333	0.9333	0.9386	**0.9437**	0.9399
Container	PSNR(dB)	24.663	25.567	26.008	26.256	26.316	**26.782**	26.423
	UQI	0.9592	0.9676	0.9709	0.9727	0.9735	**0.9761**	0.9742
Football	PSNR(dB)	27.106	28.568	29.366	29.687	29.855	**30.642**	29.932
	UQI	0.9668	0.9769	0.9810	0.9824	0.9834	**0.9861**	0.9837
Mobile	PSNR(dB)	20.310	21.197	21.596	21.758	21.907	**22.428**	21.972
	UQI	0.9373	0.9502	0.9550	0.9569	0.9590	**0.9638**	0.9598
Soccer	PSNR(dB)	29.271	30.254	30.664	30.748	30.857	**31.323**	30.992
	UQI	0.9828	0.9863	0.9876	0.9878	0.9883	**0.9894**	0.9886
Coast	PSNR(dB)	25.585	26.500	26.939	27.080	27.297	**27.800**	27.368
	UQI	0.9692	0.9754	0.9779	0.9787	0.9800	**0.9822**	0.9803

Table 4.8 Execution time of the proposed and existing algorithms at 4:1 CR

Images of different size M×N	**Execution time in seconds for different interpolation schemes**						
	Bilinear [5]	**Bicubic [4]**	**Lanczos-3 [12]**	**DCT [74]**	**UMP [P4]**	**IUM [P5]**	**EU [P6]**
Clock (200×200)	0.0141	0.0150	0.0151	0.0518	0.1261	0.1445	0.0732
Lena (256×256)	0.0147	0.0152	0.0157	0.0633	0.1282	0.1471	0.0761
Fruit (377×321)	0.0153	0.0163	0.0168	0.1629	0.1335	0.1506	0.0797
Lena (512×512)	0.0201	0.0215	0.0222	0.1773	0.1457	0.1564	0.0882
Pentagon (1024×1024)	0.0354	0.0409	0.0442	0.6372	0.1754	0.1905	0.1349

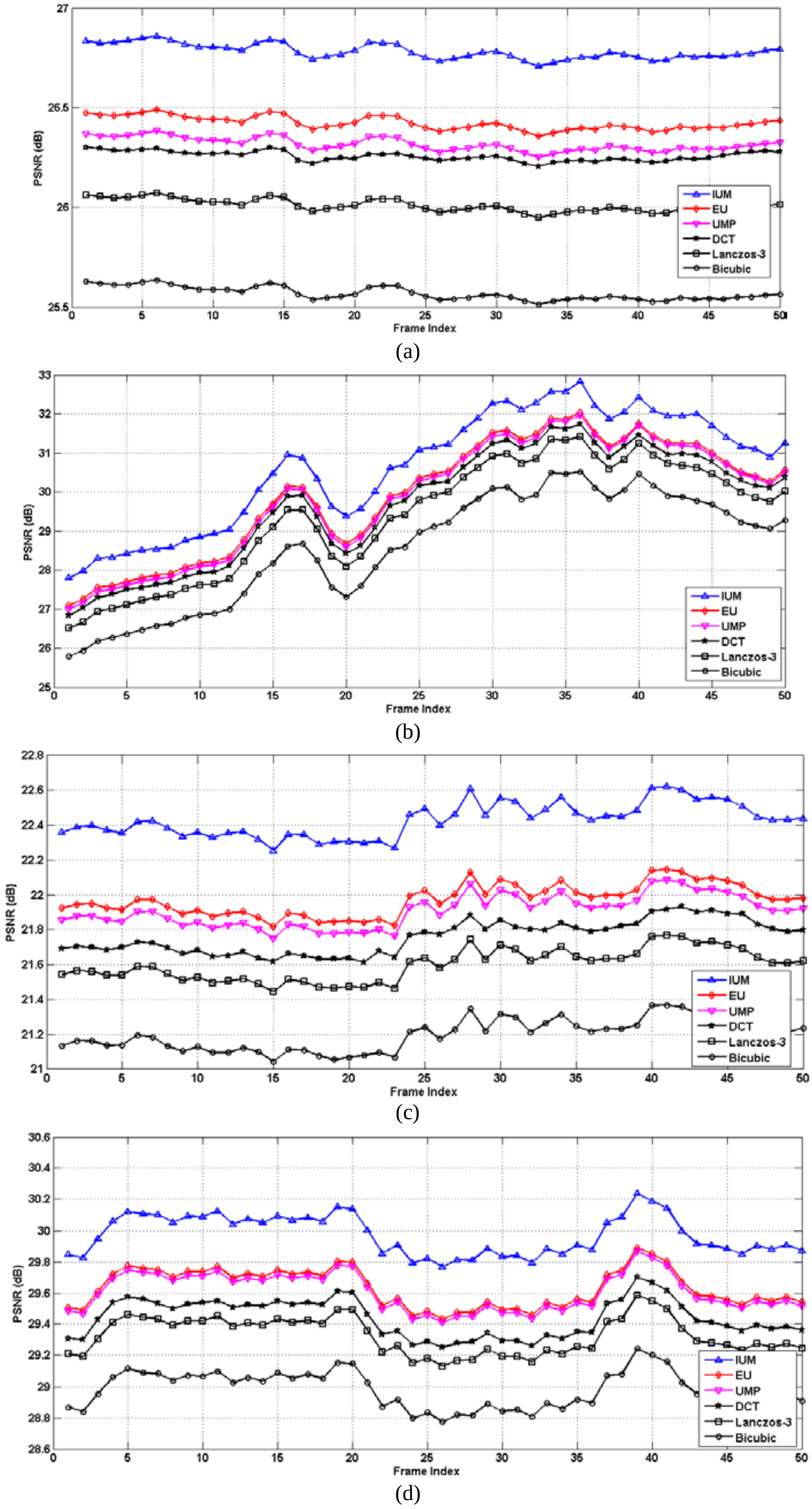


Fig. 4.11 PSNR (dB) comparisons of various up-sampling schemes at 4:1 CR, meant for different sequences: (a) Container; (b) Football; (c) Mobile; (d) Salesman

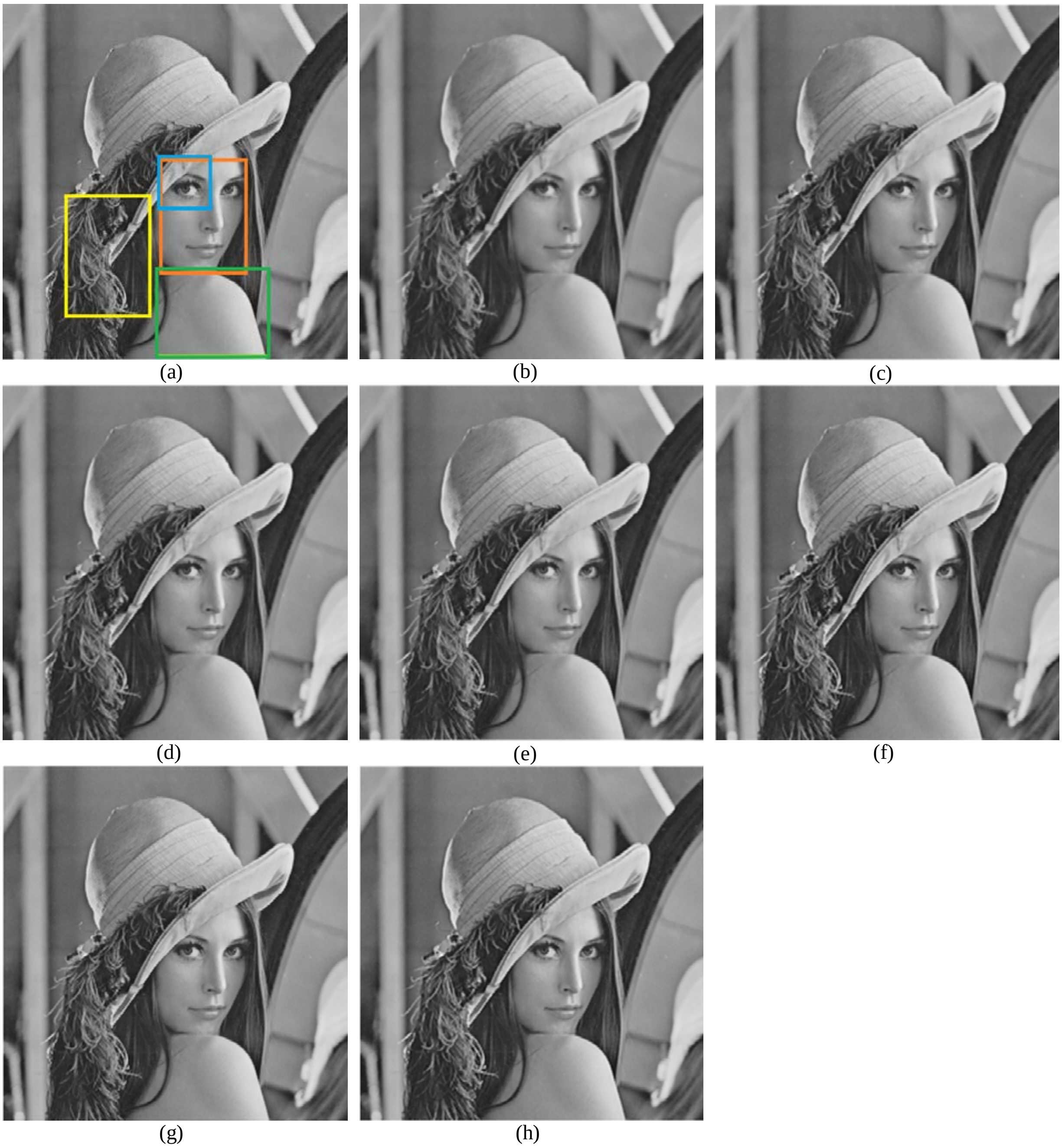


Fig. 4.12 Subjective evaluation of Lena (512×512) image using various up-sampling schemes at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) UMP (P4); (g) IUM (P5) ; (h) EU (P6)

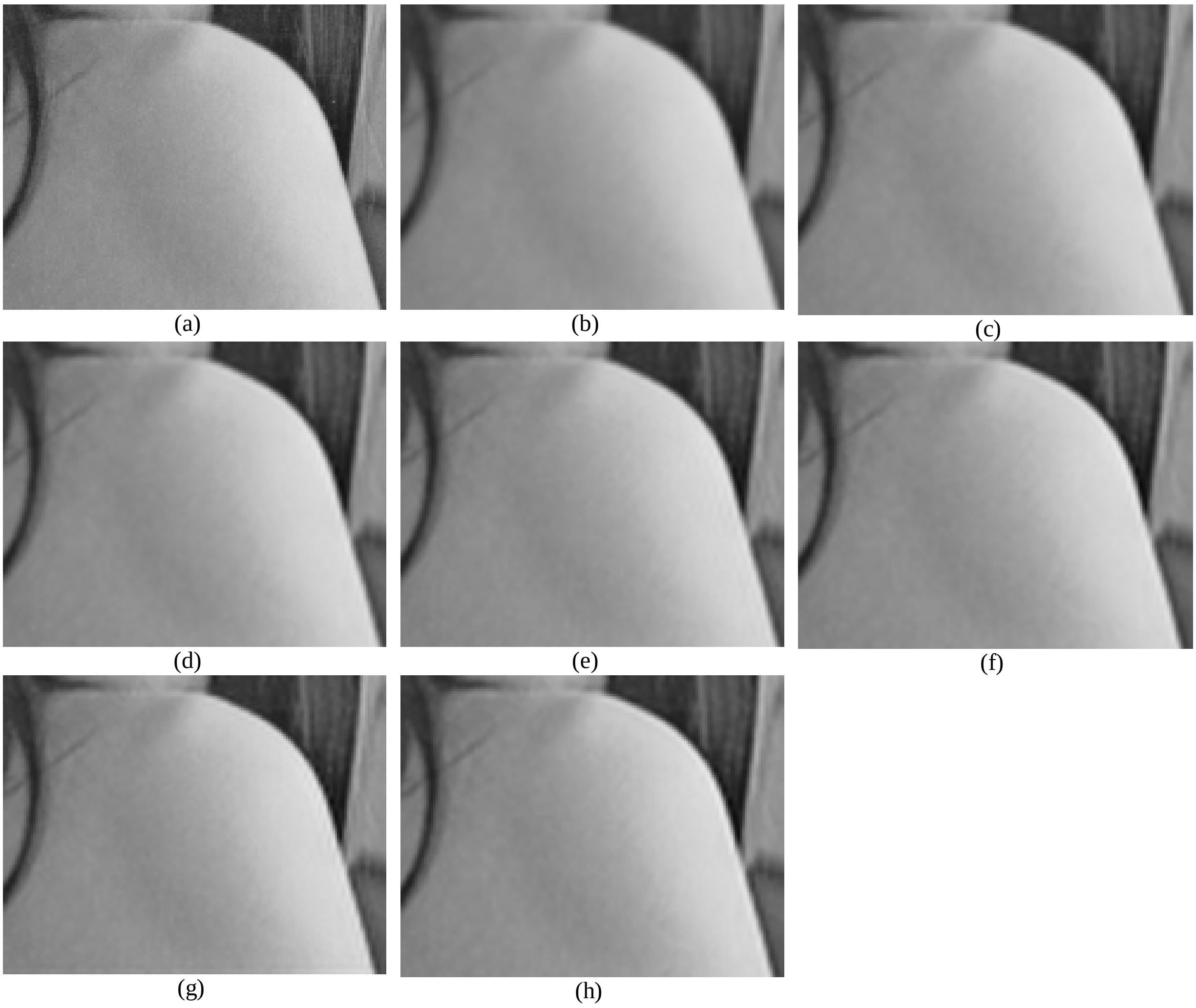


Fig. 4.13 Subjective evaluation of the selected low frequency green rectangular region (127×164) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) UMP (P4); (g) IUM (P5); (h) EU (P6)

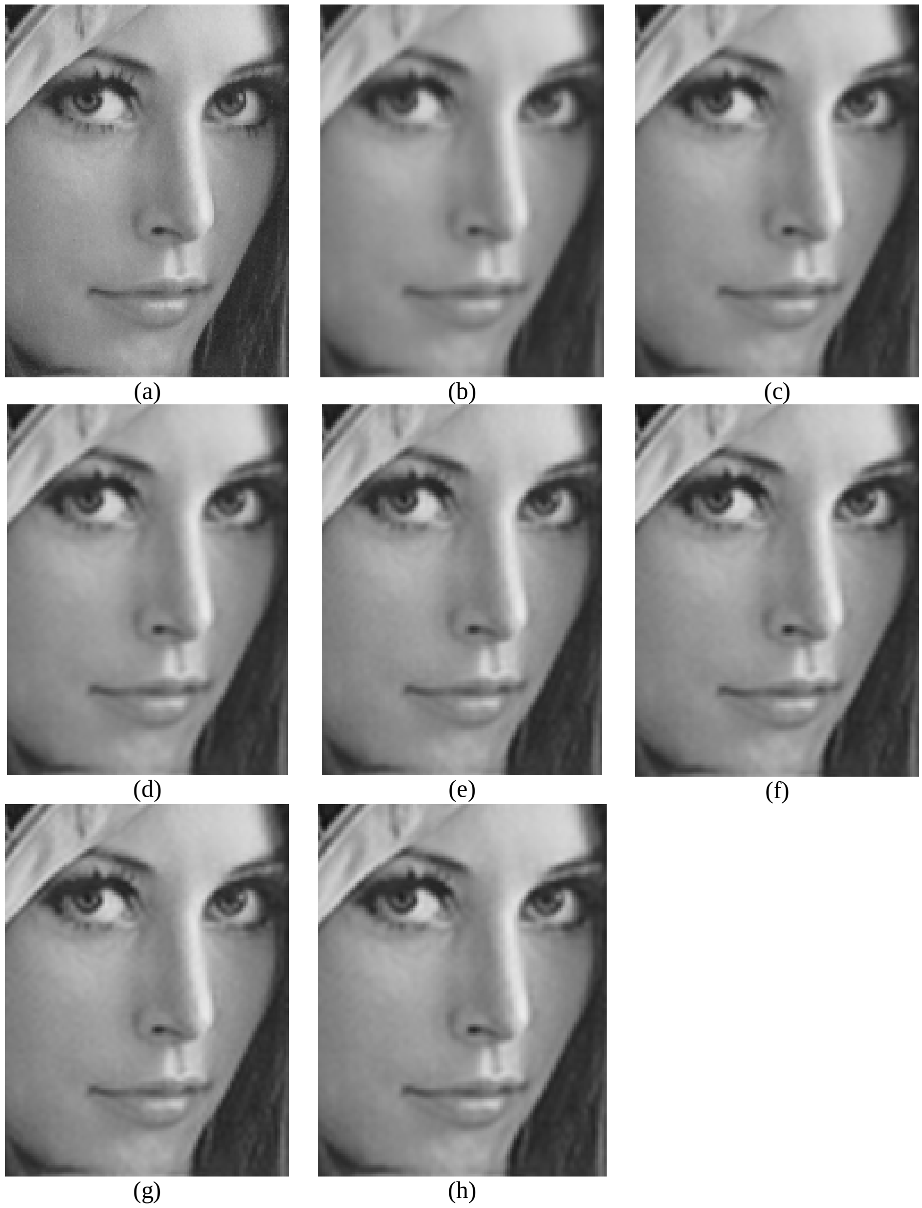
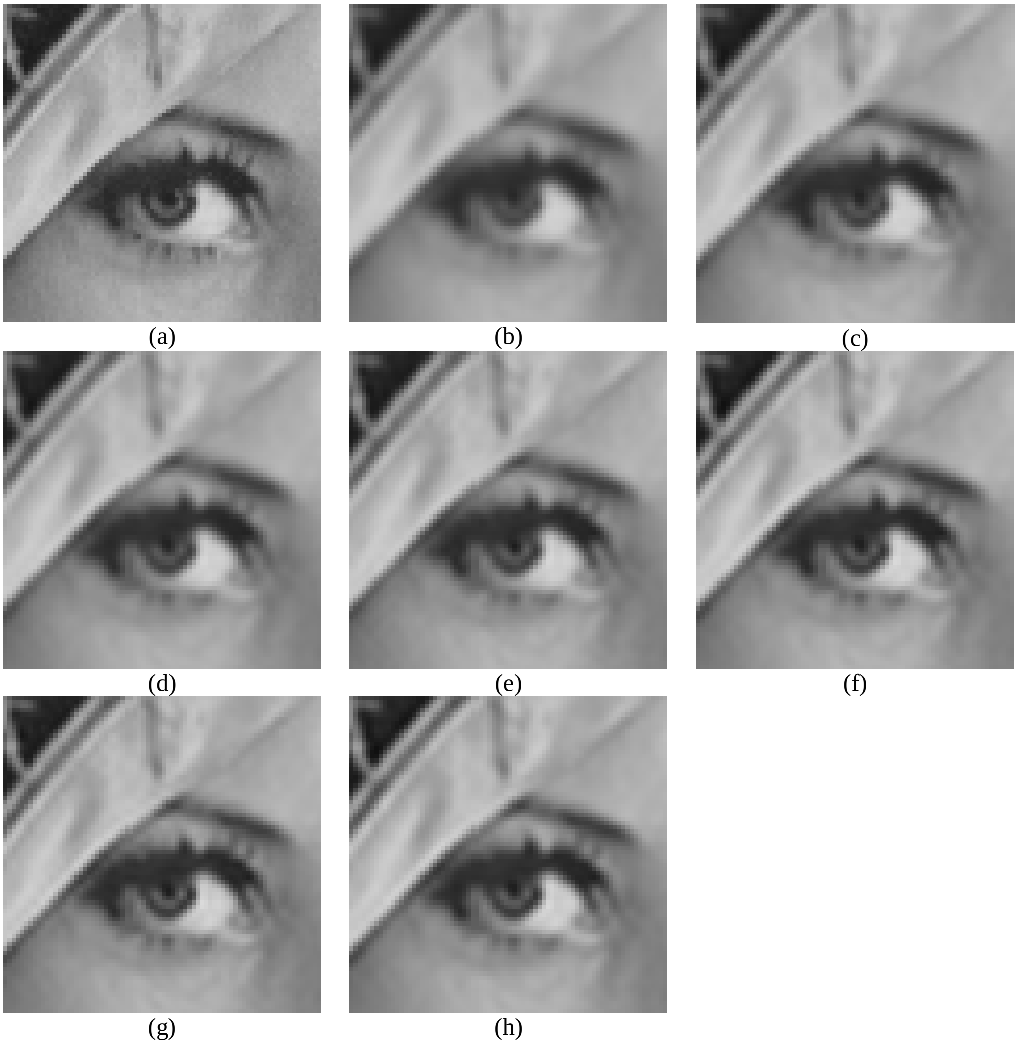


Fig. 4.14 Subjective evaluation of the selected medium frequency orange rectangular region (164×125) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) UMP (P4); (g) IUM (P5); (h) EU (P6)

Fig. 4.15 Subjective evaluation of the selected high frequency yellow rectangular region (123×174) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) UMP (P4); (g) IUM (P5); (h) EU (P6)



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 4.16 Subjective evaluation of the selected blue rectangular (76×76) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) UMP (P4); (g) IUM (P5); (h) EU (P6)

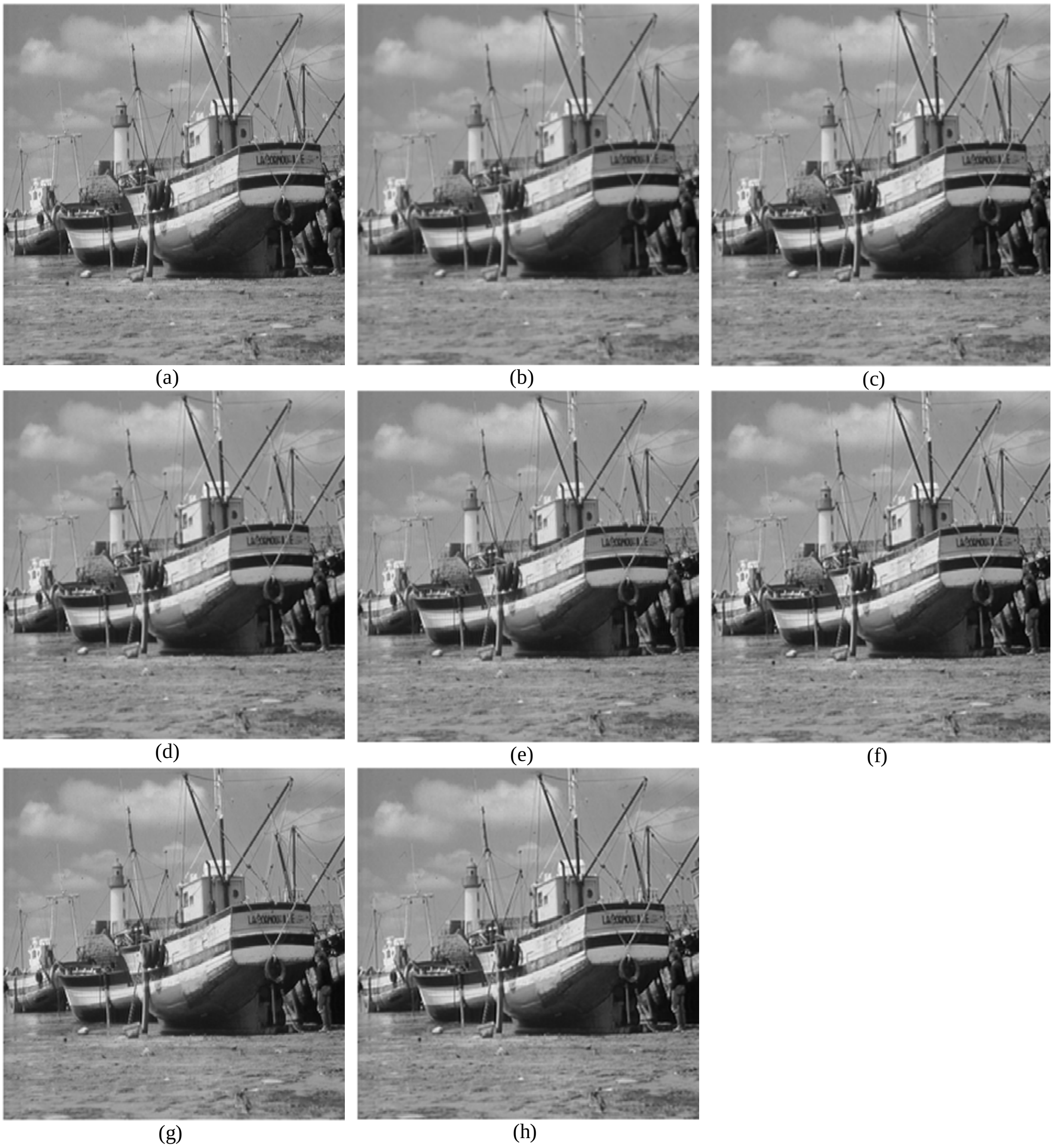


Fig. 4.17 Subjective evaluation of Boat (512×512) image using various up-sampling schemes at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) UMP (P4); (g) IUM (P5); (h) EU (P6)

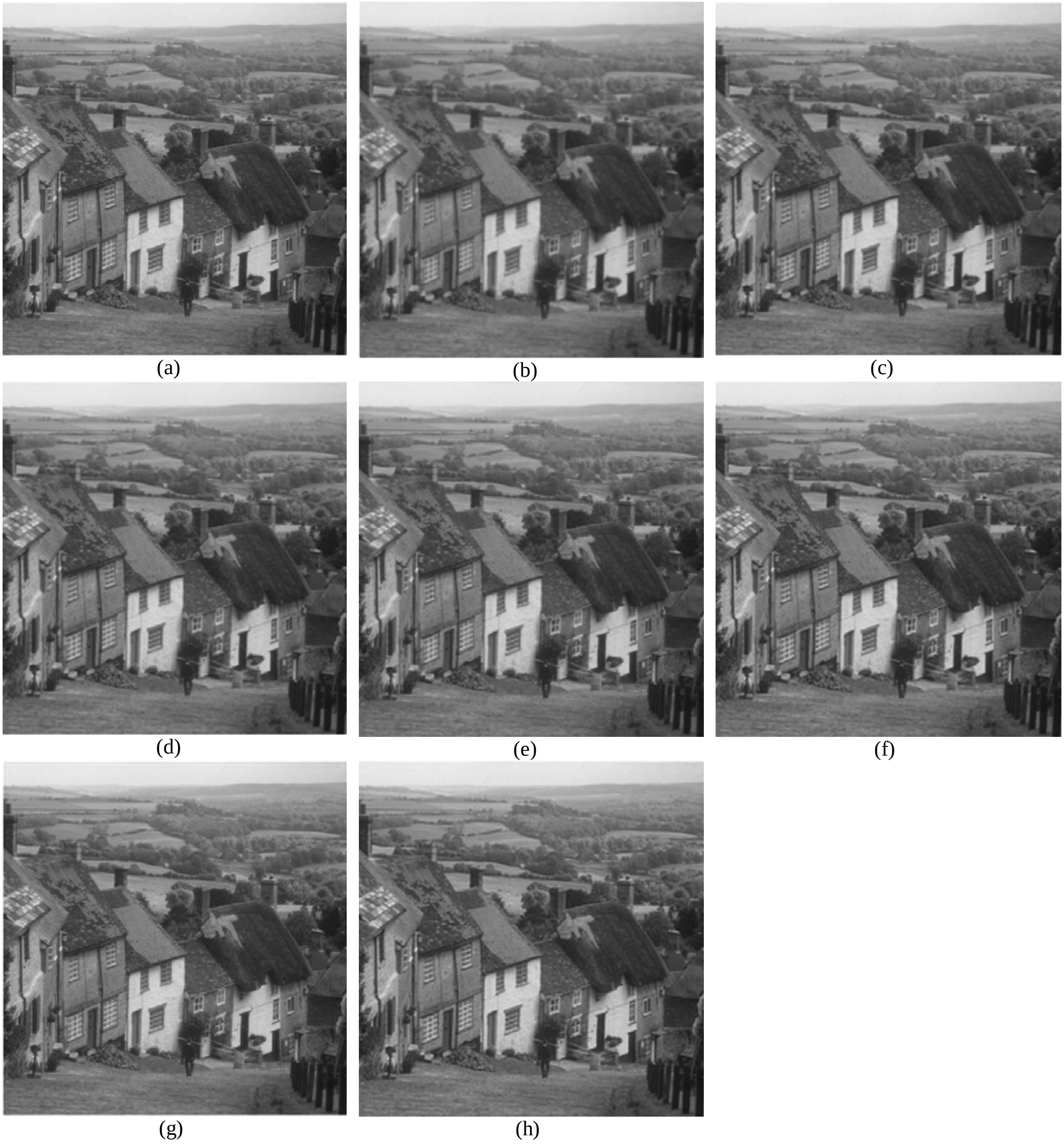


Fig. 4.18 Subjective evaluation of Goldhill (512×512) image using various up-sampling schemes at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) UMP (P4); (g) IUM (P5); (h) EU (P6)

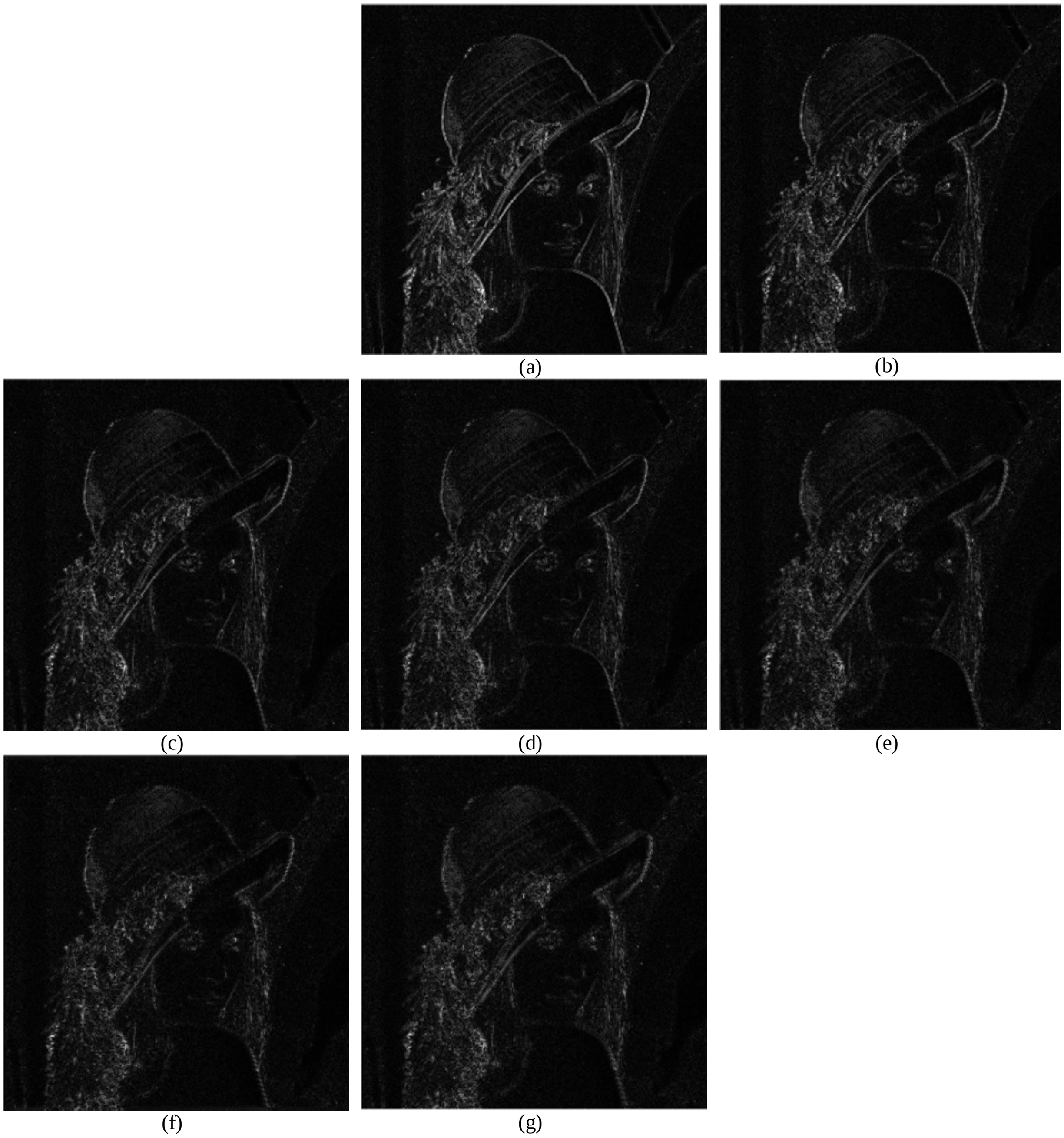


Fig. 4.19 Error image of Lena (512×512) using various up-sampling schemes at 4:1 compression ratio: (a) Bilinear; (b) Bicubic; (c) Lanczos-3; (d) DCT; (e) UMP (P4); (f) IUM (P5); (g) EU (P6)

4.5 Results and Discussion

The proposed schemes show overall better performance than the most of the existing schemes available in the literature in terms of objective and subjective measures. The IUM shows better performance amongst all the proposed and existing algorithms. Although UMP and EU algorithms exhibit better objective performance than the existing algorithms, perform considerably less than IUM as depicted in Table 4.5, Table 4.6 and Table 4.7. It may be well observed from the tables that, for almost all types of images, IUM algorithm performs much better than the other proposed and existing schemes.

In case of Cameraman (512×512) image, IUM shows a maximum PSNR improvement of 0.868 dB than DCT at 4:1 compression ratio. Similarly, in case of Fingerprint, Crowd and Lena images, IUM algorithm achieves a significant improvement of 0.77 dB, 0.7 dB and 0.546 dB over DCT respectively as shown in Table 4.5. Likewise, the algorithm shows significant PSNR gain than Lanczos-3 interpolation. In case of images like Lena, Fingerprint, Crowd and Cameraman, IUM shows significant PSNR gain of 0.756 dB, 1.181 dB, 1.016 dB and 1.484 dB. In the same way, the algorithm shows better performance in terms of PSNR and UQI than DCT and Lanczos-3 interpolation at 16:1 compression ratio as depicted in Table 4.6. Thus, the proposed schemes show better objective performance irrespective of change in compression ratio.

Besides this, the proposed algorithms show better performance than DCT for various CIF video sequences as depicted in Table 4.7 and Fig. 4.11, respectively. In case of Football (352×288) sequence, the IUM algorithm achieves an improved average PSNR gain of 0.955 dB over DCT at 4:1 compression ratio. Likewise, the IUM algorithm gives average PSNR improvement of 0.794 dB, 0.72 dB and 0.67 dB corresponding to Bus, Coastguard and Mobile sequence respectively as illustrated in Table 4.7.

In case of Barbara, IUM shows an improvement of 0.548 dB than DCT whereas UMP and EU algorithms show PSNR improvement of 0.334 dB and 0.376 dB, respectively over DCT at 4:1 compression ratio as depicted in Table 4.5. Likewise, in case of Goldhill image, IUM gives a PSNR gain of 0.45 dB than DCT whereas UMP and EU give PSNR gain of 0.139 dB and 0.223 dB respectively. Therefore, it is clear from the results that the performance of IUM is significantly better than UMP and EU algorithms whereas the performance of EU scheme is better than UMP.

This is so because the IUM algorithm extracts the VHF component through signal decomposition using unsharp masking operation iteratively and restores the VHF component of an image which is highly degraded during the up-sampling process. On the other hand, the unsharp masking pre-processing (UMP) scheme extracts the HF component which is moderately degraded during the up-sampling process. Hence, restoration of such HF component would give less degree of PSNR gain as compared to IUM. Furthermore, the performance of error based up-sampling scheme (EU) is better than UMP because of the employment of inverse model of HF degradation due to Lanczos-3 interpolation. Since the error due to Lanczos-3 interpolation is used to enhance the lost HF components, the HF components in the up-scaled image are well restored employing Lanczos-3 interpolation.

Computational time is one of the parameters to evaluate the suitability of the algorithms for real-time applications. It may be observed from Table 4.8 that the proposed algorithms consume more computational time than DCT for images of small dimensions such as Clock (200×200) and Lena (256×256). However, DCT takes more computational time than the proposed algorithms for the images of larger dimensions such as Lena (512×512) and Pentagon (1024×1024). Therefore, for high resolution images, the proposed algorithms are faster as compared to DCT. Moreover, the proposed algorithms are computationally less complex because they are global pre-processing schemes and operate on low resolution images prior to image up-scaling. In contrast, the DCT scheme, being a frequency domain interpolation technique, has more computational time requirement than the proposed algorithms because of the need for conversion from spatial to frequency domain and back.

It may be well observed from the Table 4.8 that the computation time of IUM algorithm is more than UMP because of the employment of unsharp masking for seven iterations. In contrast, the computational time of UMP is less than IUM because, it employs unsharp masking for only once. The EU algorithm has the least computation time amongst all the proposed algorithms. Furthermore, it is also found that the difference of execution time between UMP, IUM and EU algorithms is very less i.e. of the order of one hundredth of a second for images such as Lena (512×512) and Pentagon (1024×1024). Hence, all the proposed algorithms are suitable candidate for real-time applications. However, for a better objective and subjective performance, IUM must be preferred over UMP and EU algorithms.

Error image is a quality metric which is used to measure the restoration performance of various algorithms. The error images are shown in Fig. 4.19. It may be observed from this figure that the absolute error is reduced in case of the IUM and EU as compared to DCT and other existing algorithms which indicate a better HF restoration performance. In case of IUM algorithm, the absolute error is found to be least amongst all the algorithms and thus must be preferred for image up-scaling applications.

Lena (512×512) image, which has different regions of low, medium and high frequencies, is a suitable candidate for subjective evaluation of the proposed algorithms. The low, medium, high frequency regions of Lena image are considered for performance evaluation of the algorithms. The region rich in combination of different patterns is also considered for performance evaluation. Four distinct regions with different features and thus different signal characteristics are marked as shown in Fig. 4.12. Performance at these regions is distinctly analyzed. For this purpose, the output images at these regions are enlarged and shown in Fig. 4.13, Fig.4.14, Fig. 4.15 and Fig. 4.16.

The green rectangular region (127×164) containing the shoulder portion of Lena image is considered as a low frequency region. The enlarged version of it using various algorithms is given in Fig. 4.13. Similarly, the enlarged versions of medium and high frequency regions are given in Fig. 4.14 and 4.15, respectively. The face and hair regions represent the medium and high frequencies, respectively. Fig 4.16 shows the eye and its surrounding region comprising low, medium and high frequencies. In all these cases, it may be perceived that low frequency regions are well preserved, the mid frequency regions are moderately enhanced and the HF regions are highly emphasized so as to compensate the HF loss during the up-sampling process. In case of the proposed algorithms, the fine details and edge information, which represent the high frequency contents, are effectively enhanced resulting in a better visual quality. However, HF restoration performance of the proposed algorithms depends on the enhancement of frequency components that has been degraded the most. IUM algorithm achieves the best subjective performance amongst all the proposed algorithms because it restores the most degraded, VHF component.

Furthermore, the blue rectangular region of Lena which embodies the eye region contains various frequencies. Under this condition, the proposed algorithms work much better than the existing algorithms in restoring the HF contents of an image as illustrated in Fig. 4.16. So, the overall subjective performance of the proposed schemes is more satisfactory than the existing schemes.

Besides, the proposed schemes achieve better performance than the existing schemes at 16:1 compression ratio as depicted in Table 3.3. These schemes demonstrate better objective performance in terms of PSNR and UQI irrespective of variation in compression ratio, image resolution and image types and so are more versatile.

4.6 Conclusion

From the result analysis, it is apparent that the IUM shows much better subjective and objective performance over DCT, EU, UMP and most of the other existing algorithms available in the literature irrespective of image and video types. The EU performs less as compared to IUM but shows better performance than UMP algorithm for different image and video signals. UMP performs least amongst the proposed algorithms.

IUM shows its peak performance in case of images like Fingerprint, Crowd and Cameraman which are rich in HF contents. For rest of the images, it performs much better than other algorithms. It is so because the algorithm is based on enhancement of HF contents to counter blurring in the up-scaled images. It shows better performance because of its ability to restore the most degraded VHF component through signal decomposition using the filter bank which employs unsharp masking iteratively.

Although the EU algorithm shows inferior performance than IUM, it performs better than UMP, DCT and most of the other existing algorithms for majority of images and videos. It shows low-grade performance than DCT in case of images like Fingerprint, Cameraman and Lake. But for most of the images, it shows comparatively better performance than DCT and UMP because it employs the inverse model of HF degradation due to Lanczos-3 interpolation prior to image up-scaling. Since the same Lanczos-3 interpolation is used to determine the HF degradation through error calculation and also used in the up-scaling process, it ensures better HF restoration performance.

The UMP algorithm shows inferior performance than EU and IUM but for majority of images, it shows better performance than DCT and other existing algorithms. Since the UMP restores the HF component which is less degraded than VHF component, contribute lesser PSNR hike than IUM. The UMP shows better PSNR gain than DCT in some images but performs less than DCT in case of images such as Fingerprint, Baboon, Cameraman, Lake and Elaine. This is due to more weight factor deviation, ΔK with respect to the average weight factor. This problem is resolved in case of IUM because of the convergence of ΔK at higher order iterations.

Since these proposed algorithms are global pre-processing techniques and operate on low-resolution images, have much reduced computational complexities than DCT. Therefore, these proposed schemes impart very less computational burden on the various displaying devices and hence are suitable for real-time applications. In addition, the proposed algorithms work efficiently under variation in compression ratio and image resolution irrespective of images and video types and hence are more versatile and suitable for various multimedia applications.

Chapter 5

Post-processing Algorithms using Soft-computing Techniques

Preview

This chapter presents three spatial domain post-processing algorithms to tackle non-uniform blurring in an up-scaled image. The degree of blurring depends on the level of local high frequency (HF) information. The blurring is more significant in the local regions with more HF contents than the medium and low frequency regions. To tackle such non-uniform blurring, three post-processing schemes are proposed in this chapter. The term *post-processing* refers to processing an image which is previously up-scaled using a suitable interpolation technique.

Local adaptive Laplacian (LAL) scheme is a non-fuzzy, local based scheme. The local regions of an up-scaled image with high variance are sharpened more than the region with moderate or low variance by employing a local adaptive Laplacian kernel. The weights of the LAL kernel are varied as per the normalized local variance so as to provide more degree of HF enhancement to high variance regions than the low variance counterpart so as to effectively counter the non-uniform blurring.

Soft-computing techniques like fuzzy systems, adaptive neural based fuzzy inference system (ANFIS), artificial neural network (ANN) etc. have been tested for various non-stationary time series prediction over the last few decades. Hence, there is a scope of applying different soft computing techniques for HF prediction in the up-scaled images. In this chapter, fuzzy logic and Legendre functional link artificial neural network (LFLANN) are applied to alleviate blurring in up-scaled images and are compared with non-fuzzy approach such as LAL.

A fuzzy weighted Laplacian (FWL) post-processing scheme with a higher degree of non-linearity is proposed in this chapter to overcome the non-uniform blurring problem. FWL, being a fuzzy based mapping scheme, is highly nonlinear to resolve the blurring problem more effectively than a non-fuzzy approach such as LAL which employs a linear mapping. FWL is a local sharpening scheme that sharpens the up-sampled images as per the local statistics by employing fuzzy based mapping so as to alleviate the blurring caused by Lanczos-3 interpolation.

Furthermore, a Legendre functional link artificial neural network (LFLANN) based post-processing scheme is proposed here to minimize the cost function so as to reduce the blurring in a 2-D up-scaled image. Legendre polynomials are used for functional expansion of the input pattern-vector and provide high degree of nonlinearity to the system. So, the requirement of multiple layers can be replaced by single layer LFLANN architecture to reduce the cost function effectively for better restoration performance. The presence of a single layer reduces the computational complexity and may be suitable for various real-time applications. Simulation results, presented at the end of the chapter, are quite encouraging.

The organisation of this chapter is given below.

- *Local Adaptive Laplacian (LAL) based Post-processing Algorithm*
- *Fuzzy Weighted Laplacian (FWL)based Post-processing Algorithm*
- *Legendre Functional Link Artificial Neural Network (LFLANN) based Post-processing Algorithm*
- *Results and Discussion*
- *Conclusion*

5.1 Local Adaptive Laplacian (LAL) based Post-processing Scheme [P7]

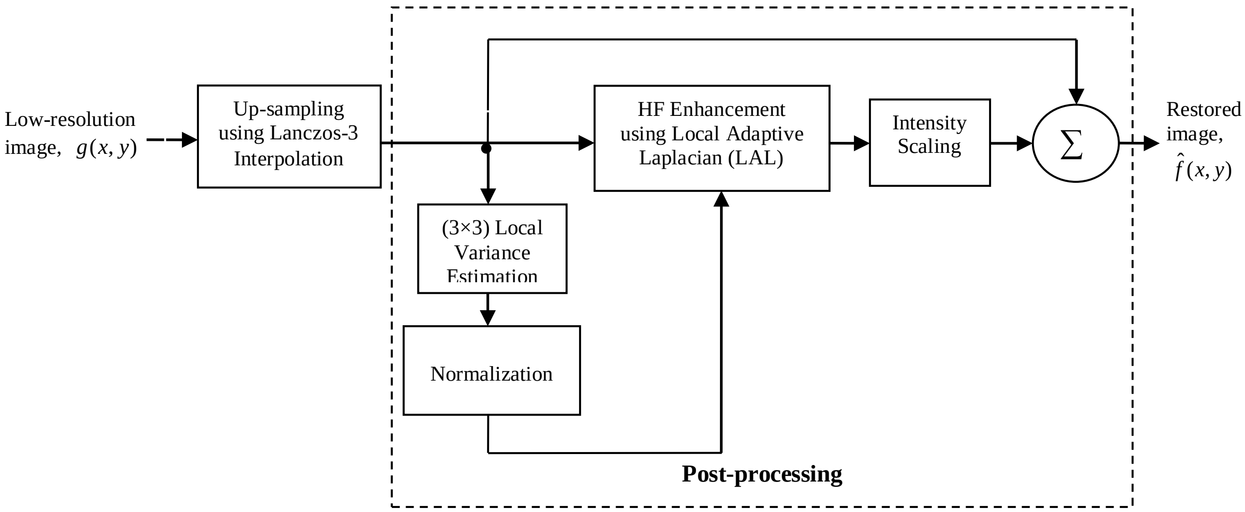


Fig. 5.1 Block Diagram of Local Adaptive Laplacian based Post-processing Scheme

The local adaptive Laplacian (LAL) is a local post-processing scheme, used for the HF enhancement of an up-scaled image so as to reduce its blurring effectively. The blurring is significant at the edges and fast changing regions with high variance as compared to the slowly varying, flat regions with low variance. Therefore, the blurring can be countered effectively by locally enhancing HF regions more than the medium and low frequency regions through local post-processing. To meet the requirement, a 2-D local adaptive Laplacian kernel is employed whose weights are updated as per the normalized local variance of a 3×3 neighbourhood. Therefore, if the local variance is more, the central kernel weight becomes proportionately high and vice versa based on the direct mapping basis. The remaining coefficients of the kernel are attuned as per the central weight so that the sum of all coefficients in the adaptive Laplacian kernel becomes zero.

In case of a slowly varying or a flat region, the slope of the kernel is low resulting in less degree of HF enhancement. In contrast, for a high variance region, the slope of the kernel becomes stiffer resulting in more degree of HF enhancement or sharpening. Hence, with the adaptability of the local adaptive Laplacian (LAL) kernel with respect to the local statistical parameter, the post-processing algorithm efficiently enhances the fast changing and edge regions and at the same time retains the smoothness of flat and slowly varying regions.

HF enhancement using Local adaptive Laplacian is a two-pass scheme. Initially, the maximum local variance of an image is computed during the first pass and the same is used to determine the normalized local variance during the second pass. The normalized local variance of a neighborhood updates the center co-efficient of the Laplacian kernel. The remaining filter co-efficients are attuned according to the central weight so that the sum of all the filter coefficients becomes zero. The local HF extract, is obtained by linearly convolving the up-sampled image with LAL kernel. Finally, the weighted version of the local HF extract is superimposed on the up-sampled image to obtain the restored image with significantly less degree of blurring. The details of HF restoration using LAL is given in the following algorithm.

5.1.1 LAL algorithm

Let $g(x, y)$ and $g_u(x, y)$ denote a low resolution image and its corresponding up-scaled image using Lanczos-3 interpolation. Let them be of size $(P \times Q)$ and $(2P \times 2Q)$, respectively. Let $\sigma^2(x, y)$ and $\sigma^2_{\max}$ denote 3×3 local variance and maximum local variance (global maximum) of an image respectively. The two pass local adaptive Laplacian (LAL) algorithm is given below.

First Pass (Determination of maximum local variance):

Step-1. Select a 3×3 window, w in an up-scaled image, $g_u(x, y)$.

$$w_{s,t}(x, y), \quad -1 \le s, t \le 1$$

Step-2. Determine the local variance, $\sigma^2(x, y)$ in a 3×3 neighborhood as:

$$\sigma^2(x, y) = \frac{1}{(2p+1)^2} \sum_{s=-p}^{p} \sum_{t=-p}^{p} [g_u(x+s, y+t) - m]^2 \tag{5.1}$$

where,

$$m = \frac{1}{(2p+1)^2} \sum_{s=-p}^{p} \sum_{t=-p}^{p} w(x+s, y+t) \tag{5.2}$$

and $p=1$ for a 3×3 neighbourhood.

Step-3. Store the local variance in a null vector.

Step-4. Repeat step-1 to step-3 for all (x, y) locations of the up-scaled image, $g_u(x, y)$.

Step-5. Determine the maximum local variance within the up-scaled image $g_u(x, y)$ of size $(2P \times 2Q)$ using the following expression.

$$\begin{aligned} &\sigma_{\max}^2 = \max\left(\left\{ \sigma^2(x, y) \right\} \right) \\ &\forall \; x \in \{2, 3, 4, \cdots, 2P-1\} \\ &\forall \; y \in \{2, 3, 4, \cdots, 2Q-1\} \end{aligned} \tag{5.3}$$

Second Pass (HF enhancement using local adaptive Laplacian):

Step-6. Determine the 3×3 local variance of the up-scaled image, $g_u(x, y)$ using step-2.

Step-7. Determine the normalize local variance, $\sigma_N^2(x, y)$ to a scale of 10 using the following expression.

$$\sigma_N^2(x, y) = 10 \times \frac{\sigma^2(x, y)}{\sigma_{\max}^2} \tag{5.4}$$

Step-8. Update the filter coefficients of LAL kernel locally using normalized local variance, $\sigma_N^2(x, y)$ as per the following so that the sum of all filter coefficients becomes zero.

$$h_{LAL}(x, y) = \begin{bmatrix} 0 & -\frac{\sigma_N^2(x, y)}{4} & 0 \\ -\frac{\sigma_N^2(x, y)}{4} & \sigma_N^2(x, y) & -\frac{\sigma_N^2(x, y)}{4} \\ 0 & -\frac{\sigma_N^2(x, y)}{4} & 0 \end{bmatrix} \tag{5.5}$$

$$\sum_{Z=1}^{9} \alpha_Z = 0 \tag{5.6}$$

where, α_Z denotes the filter coefficients of LAL kernel.

Step-9. Obtain the HF extracts, $\psi_{LAL}(x, y)$ of the up-scaled image, $g_u(x, y)$ by linearly convolving $g_u(x, y)$ with LAL kernel, $h_{LAL}(x, y)$.

$$\psi_{LAL}(x,y)=\sum_{s=-1}^{1}\sum_{t=-1}^{1}h_{LAL}(s,t)\,g_u(x+s,y+t) \tag{5.7}$$

Step-10. Superimpose the weighted version of the HF extract, $\psi_{LAL}(x,y)$ on the up-scaled image, $g_u(x,y)$ to obtain the restored, de-blurred image, $\hat{f}(x,y)$.

$$\hat{f}(x,y)=g_u(x,y)+K\,\psi_{LAL}(x,y) \tag{5.8}$$

Step-11. Repeat step-6 to step-10 for all (x,y) locations to obtain the HF restored up-scaled image $\hat{f}(x,y)$.

The details of weight factor, K estimation is given in Section 5.2.3.

5.2 Fuzzy Weighted Laplacian (FWL) based Post-processing Scheme [P8]

In this scheme, an attempt has been made to enhance the nonlinearity of a direct normalized mapping approach, LAL by incorporating a fuzzy rule base to improve its HF restoration performance. A fuzzy-based post-processing algorithm with a higher level of nonlinearity is proposed here to alleviate non-uniform blurring effectively in an up-scaled image.

The proposed fuzzy weighted Laplacian (FWL) algorithm is evaluated and compared with a non-fuzzy, direct mapping based LAL scheme to show its performance improvement as a function of nonlinearity.

The FWL is almost similar to LAL algorithm with a difference in the mapping scheme between input and output variables. In case of LAL, normalized mapping technique is used between the input and output variables; whereas, in case of FWL algorithm, fuzzy based mapping technique is used, which is more nonlinear. The remaining parts of both of the algorithms are same.

5.2.1 Fuzzy based Mapping

FWL scheme is a single-input-single-output (SISO) fuzzy inference system (FIS) as shown in Fig. 5.2. In this case, a normalized local variance, $\sigma_N^2(x,y)$ in a 3×3 neighborhood is taken as the input variable whereas the central weight of the local adaptive Laplacian kernel, w_{tO} is taken

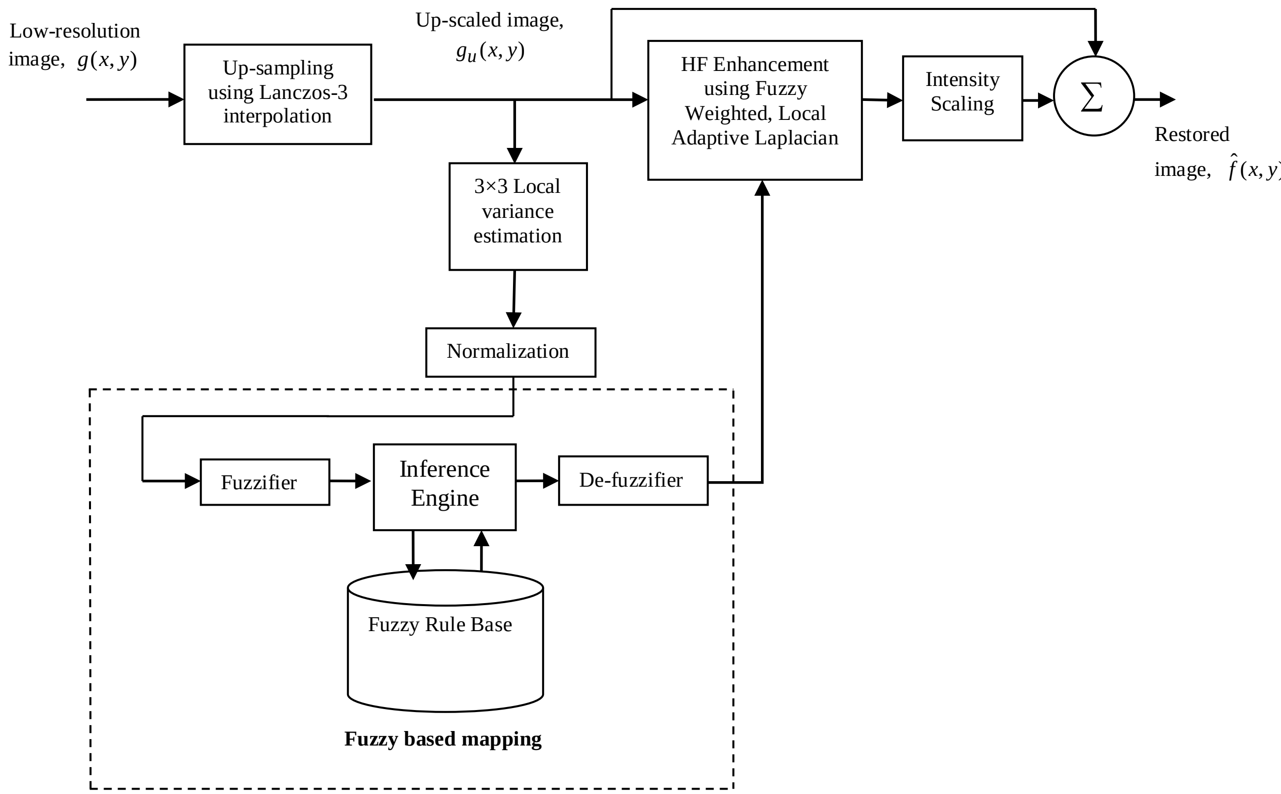


Fig. 5.2 Fuzzy Weighted Laplacian (FWL) based Post-processing Scheme

as the output variable. Triangular input and output membership functions are taken for our suitability. The number of input and output membership functions is taken as 3 and are shown in Fig. 5.3. The range of both the input and output variables is kept within (0-10) scale for our suitability.

Let μ_{iA}, μ_{iB} *and* μ_{iC} denote the low, medium and high membership function of input variable, σ_N^2. Similarly, μ_{OA}, μ_{OB} *and* μ_{OC} denote the low, medium and high membership function of the output variable, respectively. The expressions for input membership functions are given by,

$$\mu_{iA}(\sigma_N^2; 0, 0, 5) = \begin{cases} 0, & \sigma_N^2 \le 0 \\ \dfrac{5 - \sigma_N^2}{5}, & 0 \le \sigma_N^2 \le 5 \\ 0, & 5 \le \sigma_N^2 \end{cases} \tag{5.9a}$$

$$\mu_{iB}(\sigma_N^2; 2.5, 5, 7.5) = \begin{cases} 0, & \sigma_N^2 \le 2.5 \\ \dfrac{\sigma_N^2 - 2.5}{5 - 2.5}, & 2.5 \le \sigma_N^2 \le 5 \\ \dfrac{7.5 - \sigma_N^2}{7.5 - 5}, & 5 \le \sigma_N^2 \le 7.5 \\ 0, & 7.5 \le \sigma_N^2 \end{cases} \tag{5.9b}$$

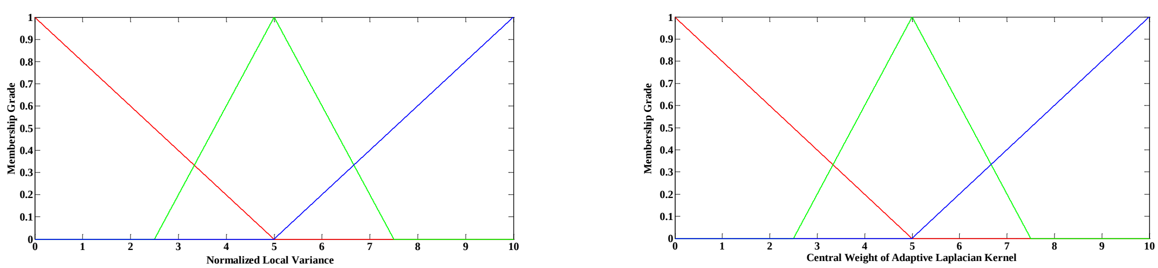
$$\mu_{iC}(\sigma_N^2; 5, 10, 10) = \begin{cases} 0, & \sigma_N^2 \le 5 \\ \dfrac{\sigma_N^2 - 5}{10 - 5}, & 5 \le \sigma_N^2 \le 10 \\ 0, & 10 \le \sigma_N^2 \end{cases} \tag{5.9c}$$

Similarly, the expressions for output membership functions are given by,

$$\mu_{OA}(w_t; 0, 0, 5) = \begin{cases} 0, & w_t \le 0 \\ \dfrac{5 - w_t}{5}, & 0 \le w_t \le 5 \\ 0, & 5 \le w_t \end{cases} \tag{5.10a}$$

$$\mu_{OB}(w_t; 2.5, 5, 7.5) = \begin{cases} 0, & w_t \le 2.5 \\ \dfrac{w_t - 2.5}{5 - 2.5}, & 2.5 \le w_t \le 5 \\ \dfrac{7.5 - w_t}{7.5 - 5}, & 5 \le w_t \le 7.5 \\ 0, & 7.5 \le w_t \end{cases} \tag{5.10b}$$

$$\mu_{OC}(w_t; 5, 10, 10) = \begin{cases} 0, & w_t \le 5 \\ \dfrac{w_t - 5}{10 - 5}, & 5 \le w_t \le 10 \\ 0, & 10 \le w_t \end{cases} \tag{5.10c}$$



(a) (b)

Fig. 5.3 Graphical representation of membership functions employed in FWL algorithm: (a) Input; (b) output

The input and output membership functions of FWL algorithm are shown in Fig. 5.3. Fuzzy logic controllers are governed by a set of if-then rules known as a knowledge base or rule base. The fuzzy rule base drives the inference engine to produce the output in response to one or a set of inputs. The fuzzy based mapping is based on fuzzy rule base. According to the knowledge base of FWL, if the 3×3 normalized local variance is more, the central weight of the adaptive Laplacian kernel increases and vice versa so as to perform more sharpening in the high variance regions than the slowly varying flat regions for effective high frequency restoration. The knowledge base for FWL, which is established on a heuristic approach, is given as per the following.

Fuzzy If-then Rules:

Rule I: If the normalized local variance, $\sigma_N^2(x,y)$ *is* ***low*** *then the central kernel weight,* w_t *is* ***low***

OR

Rule II: If $\sigma_N^2(x,y)$ *is* ***medium*** *then* w_t *is* ***medium***

OR

Rule III: If $\sigma_N^2(x,y)$ *is* ***high*** *then* w_t *is* ***high***

The rule base contains all the information required to relate the inputs and outputs. The minimum of the input and the output membership function is performed as per the fuzzy if-then rules. Subsequently, the inference engine operates with the min - max operator to generate the output responses. The output responses are then de-fuzzified to produce a crisp output employing the center of gravity de-fuzzification method. The central weight, w_{tO} of the FWL kernel is updated as per the defuzzified output of FIS.

So, the minimum of the two membership functions as per the fuzzy if-then rule 1, 2 and 3 are given by,

$$\mu_{iA\cap OA}(\sigma_N^2, w_t) = \min\left\{\mu_{iA}(\sigma_N^2), \mu_{OA}(w_t)\right\} \tag{5.11a}$$

$$\mu_{iB\cap OB}(\sigma_N^2, w_t) = \min\left\{\mu_{iB}(\sigma_N^2), \mu_{OB}(w_t)\right\} \tag{5.11b}$$

$$\mu_{iC\cap OC}(\sigma_N^2, w_t) = \min\left\{\mu_{iC}(\sigma_N^2), \mu_{OC}(w_t)\right\} \tag{5.11c}$$

To obtain an individual response, $R_A(w_t)$ corresponding to an arbitrary input σ_N^2, AND operation is performed in between $\mu_{iA}(\sigma_N^2)$ and the general result $\mu_{iA\cap OA}(\sigma_N^2, w_t)$ evaluated at σ_N^2 according to fuzzy if-then rule-1. Similarly, $R_B(w_t)$ and $R_C(w_t)$ are determined for rule-2 and rule-3 respectively. The general expressions for the individual responses are given by,

$$R_A(w_t) = \min\{\mu_{iA}(\sigma_N^2),\ \mu_{iA\cap OA}(\sigma_N^2, w_t)\} \tag{5.12a}$$

$$R_B(w_t) = \min\{\mu_{iB}(\sigma_N^2),\ \mu_{iB\cap OB}(\sigma_N^2, w_t)\} \tag{5.12b}$$

$$R_C(w_t) = \min\{\mu_{iC}(\sigma_N^2),\ \mu_{iC\cap OC}(\sigma_N^2, w_t)\} \tag{5.12c}$$

The overall response is obtained by aggregating the individual responses using OR operator. The overall response $R(w_t)$ is given by,

$$R(w_t) = R_A(w_t) \cup R_B(w_t) \cup R_C(w_t) \tag{5.13}$$

The crisp output, w_{tO} from the fuzzy set, R is obtained through centre of gravity de-fuzzification method. Since $R(w_t)$ can have k possible values, the centre of gravity of $R(w_t)$ is given by,

$$w_{tO} = \frac{\sum_{w_t=1}^{k} w_t\ R(w_t)}{\sum_{w_t=1}^{k} R(w_t)} \tag{5.14}$$

The de-fuzzified output, w_{tO} updates the central weight of the local adaptive Laplacian kernel $h_{FWL}(x, y)$ for enhancing local high frequency information and is given by,

$$h_{FWL}(x, y) = \begin{bmatrix} 0 & -\frac{w_{tO}}{4} & 0 \\ -\frac{w_{tO}}{4} & w_{tO} & -\frac{w_{tO}}{4} \\ 0 & -\frac{w_{tO}}{4} & 0 \end{bmatrix} \tag{5.15}$$

The remaining filter coefficients of the FWL kernel are attuned so that the sum of all the filter coefficients becomes zero. Let the filter coefficients of FWL kernel are $\alpha_1, \alpha_2, \cdots, \alpha_9$ respectively. The sum of all filter coefficients is given by,

$$\sum_{Z=1}^{9} \alpha_Z = 0 \tag{5.16}$$

The HF extracts, $\psi_{FWL}(x,y)$ of the up-scaled image, $g_u(x,y)$ is obtained by linearly convolving $g_u(x,y)$ with adaptive FWL kernel, $h_{FWL}(x,y)$.

$$\psi_{FWL}(x,y) = \sum_{s=-1}^{1} \sum_{t=-1}^{1} h_{FWL}(s,t)\, g_u(x+s, y+t) \tag{5.17}$$

Intensity scaling is performed on the high frequency extracts according to the estimated weight factor so as to perform precise sharpening to counter the blurring effectively in the up-sampled image. So, the weighted version of the HF extract, $\psi_{FWL}(x,y)$ is superimposed on the up-scaled image, $g_u(x,y)$ to obtain the restore, de-blurred image, $\hat{f}(x,y)$ and is given by,

$$\hat{f}(x,y) = g_u(x,y) + K\,\psi_{FWL}(x,y) \tag{5.18}$$

Precise estimation of weight factor is very essential for the optimum performance of the algorithm. Estimation of weight factor, K for FWL algorithm is illustrated in Section 5.2.3.

The overall input output mapping of FWL scheme is obtained by plotting the de-fuzzified output, w_{tO} with respect to the normalized local variance, σ_N^2. The plot is obtained using Lena (512×512) image. The overall input output curve give an idea about the characteristics of fuzzy based mapping techniques and is shown below. It is quite apparent from the figure that the FWL performs nonlinear mapping between input and output variable. However, the mapping in case of LAL scheme is linear as per (5.4). Hence, because of the nonlinearity, FWL scheme provides much better performance than LAL in terms of HF restoration.

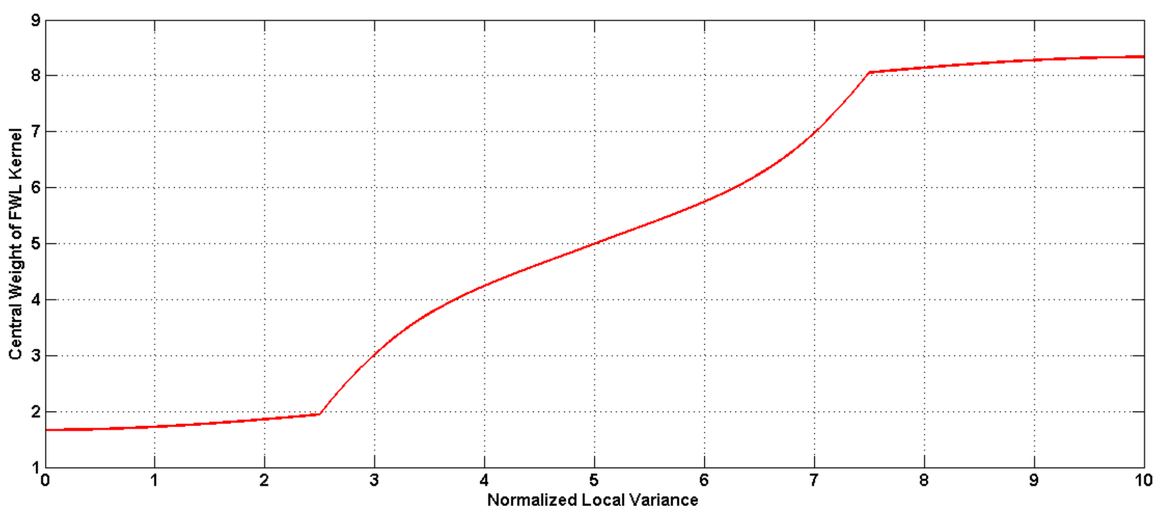
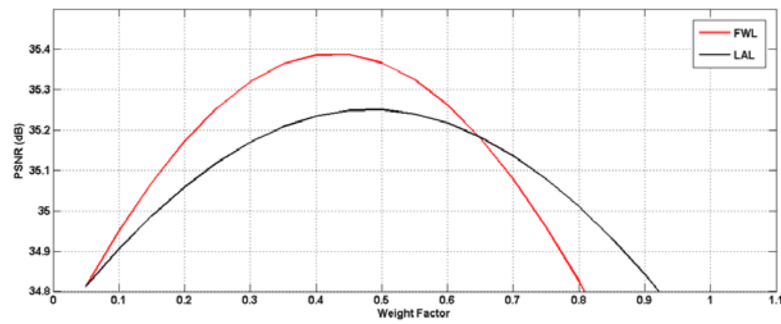
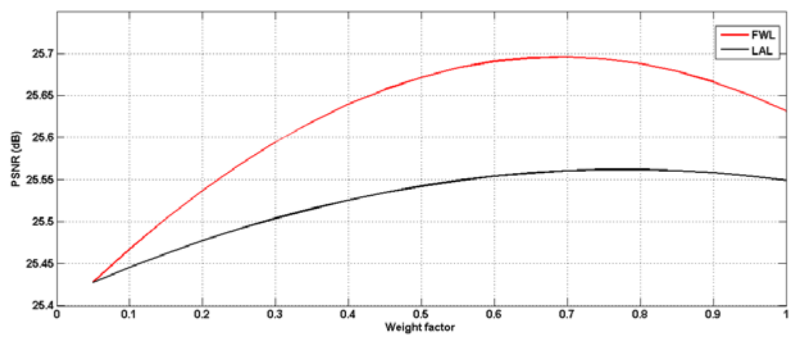


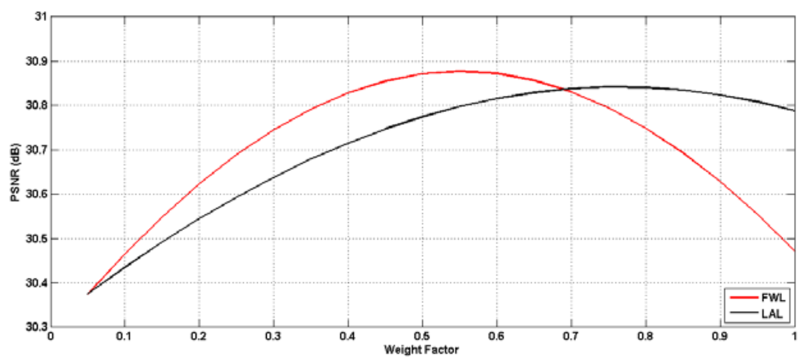
Fig. 5.4 Overall input output curve of Lena (512×512) image using FWL algorithm



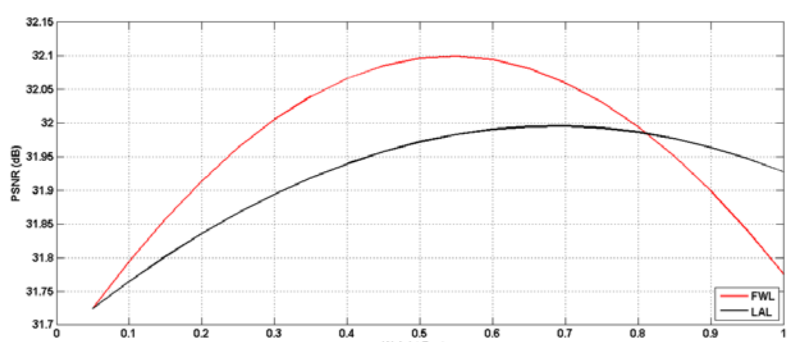
(a)



(b)



(c)



(d)

Fig. 5.5 PSNR (dB) vs. weight factor characteristics of the proposed LAL and FWL algorithms at 4:1 compression ratio for various test images: (a) Lena; (b) Barbara; (c) Boat; (d) Goldhill.

Table 5.1 Weight factor, K estimation of the proposed LAL and FWL schemes

Algorithm	Weight factors corresponding to the maximum PSNR for different images					Average Weight Factor
	Lena	Boat	Barbara	Goldhill	Livingroom	
LAL	0.5	0.75	0.75	0.7	0.8	0.70
FWL	0.43	0.55	0.7	0.55	0.57	0.56

5.2.2 FWL algorithm

The FWL algorithm is also a two-pass scheme and is almost same as LAL algorithm. The only difference is that the direct mapping in between the input and output variables is replaced by fuzzy based mapping scheme in case of FWL algorithm using (5.14) and (5.15) respectively. The remaining part of the FWL algorithm is same as LAL algorithm.

5.2.3 Weight Factor Estimation

The precise estimation of weight factor is very essential for overall performance of the algorithm. To determine the overall weight factor, the performance characteristics of PSNR (dB) vs. weight factor for different test images are obtained as illustrated in Fig. 5.5. The weight factor corresponding to maximum PSNR is determined for various images. The overall weight factor is obtained by averaging those weight factors as illustrated in Table 5.1.

5.3 Legendre Functional Link Artificial Neural Network (LFLANN) based Post-processing [P9]

The blurring in up-sampled images is non-uniform and depends on the local statistical properties of an image. The blurring is more in high variance regions in comparison to the regions with low variance. Since the artificial neural networks (ANN) are specifically developed to handle various nonlinear problems [118], Legendre functional link artificial neural network (LFLANN) [119] based image restoration scheme is exploited here to resolve the non-uniform blurring issues in an up-sampled image. LFLANN is used here for such de-blurring operation because of its faster rate of convergence and lesser computational complexity due to its single layer architecture. The need for multilayer structure is resolved by enhancing the input pattern using Legendre polynomials based nonlinear function expansion.

In case of multi layer perceptron (MLP), the inclusion of hidden layer increases its nonlinearity. However, it also increases the computational complexities of the system. So, an MLP takes longer time to optimize the weight vector and is not suitable for any real-time applications. In contrast, the requirement of hidden layer is eliminated using a single layer LFLANN structure. A great deal of nonlinearity is introduced in the algorithm by incorporating Legendre polynomials based nonlinear function expansion of input pattern vector. Since the proposed method operates on an up-scaled image, it is considered as a post-processing operation.

In this proposed algorithm, the input pattern vector is a 3×3 neighbourhood of an up-scaled, blurred image which is introduced to the input node of the LFLANN structure. The associated desired value is the corresponding pixel value of the original reference image. The enhanced input pattern vector is obtained using Legendre polynomials based functional expansion. The structure of the LFLANN is {9-1} and each pixel value of a 3×3 neighbourhood is expanded N times using Legendre function expansion schemes where, N denotes the degree of expansion. The degree of expansion may vary for improved HF restoration performance. If the degree of expansion increases, the nonlinearity of the input pattern is also increased so as to provide better HF restoration performance. However, there is a limit to the rise in the degree of expansion because, beyond a certain value, the performance of the algorithm starts declining as shown in Table 5.2. The values in bold letter in Table 5.2 signify maximum objective performance. So, in case of the proposed LFLANN algorithm, the degree of expansion is kept four to obtain the optimum performance.

Table 5.2 PSNR and UQI variations with respect to the degree of expansion, N for LFLANN

Images	Metric	PSNR (dB) and UQI corresponding to different degree of expansion					Estimation of N
		$N=1$	$N=2$	$N=3$	$N=4$	$N=5$	
Boat	PSNR (dB)	28.141	29.896	30.792	**30.848**	27.303	
	UQI	0.9782	0.9853	0.9875	**0.9878**	0.9700	4
Barbara	PSNR(dB)	24.421	25.401	25.620	**25.630**	24.889	
	UQI	0.9607	0.9673	0.9690	**0.9692**	0.9612	4
Goldhill	PSNR (dB)	28.174	31.234	31.895	**32.027**	30.422	
	UQI	0.9791	0.9896	0.9911	**0.9914**	0.9876	4

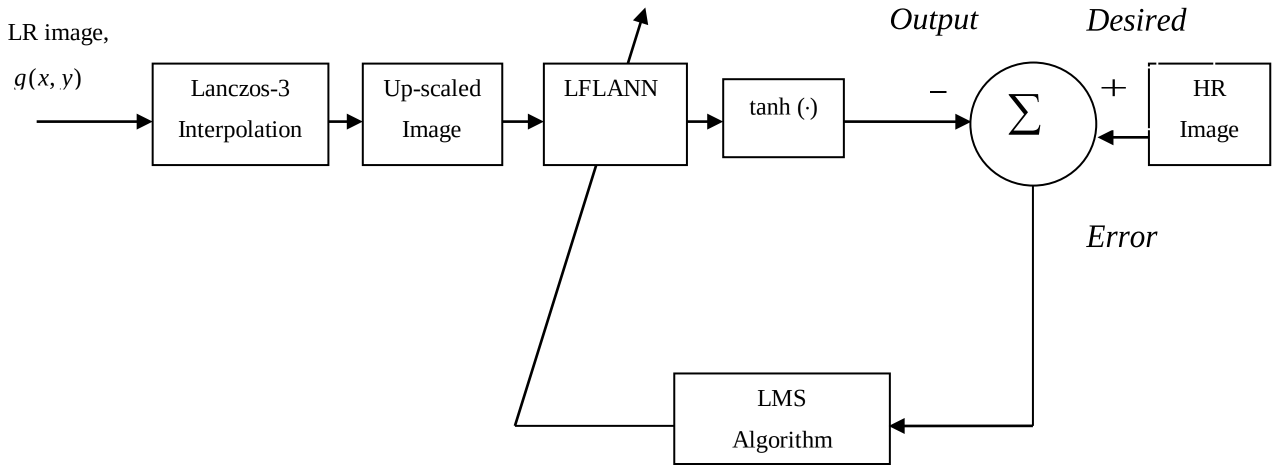


Fig. 5.6a Training phase of LFLANN

Fig. 5.6b Testing Phase of LFLANN

Since the degree of expansion is taken as four, the total number of expanded input pattern corresponding to a 3×3 neighbourhood becomes 36. Accordingly, there will be 36 number of weights that are to be updated using an adaptive algorithm. The weighted sum of all the inputs is passed through the nonlinear $\tanh(\cdot)$ function to produce the output. The output is compared with the corresponding pixel value of the target reference image and hence, the network is learned through supervised learning. The error between the target value and the output value is used to update the weights of the network. The training input and the corresponding target pixel value are normalized to fall within the interval (0-1). The mean square error (MSE) is taken as the cost function. Least mean square (LMS) algorithm is used to minimize the cost function by updating all the weights of the network.

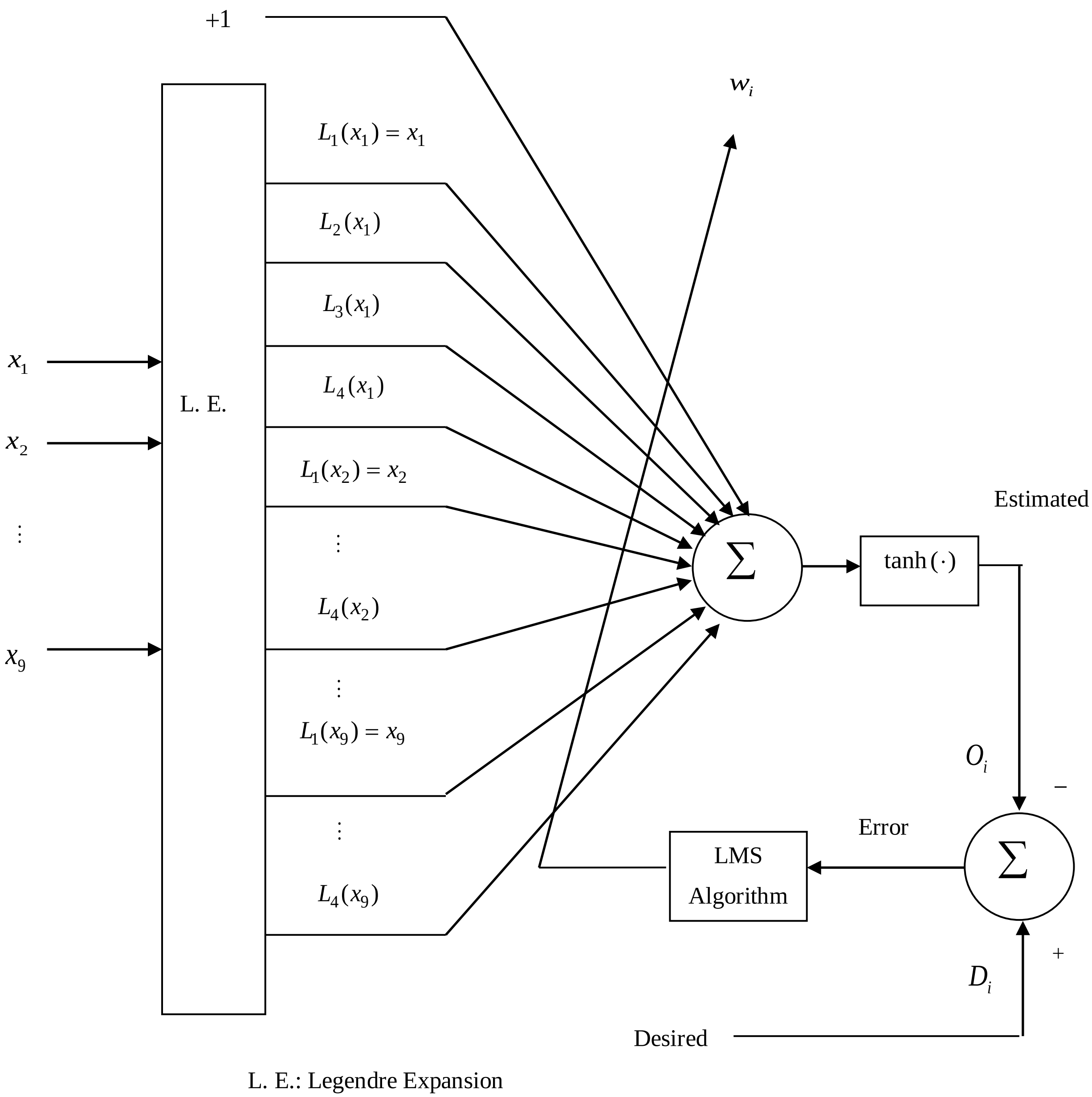


Fig. 5.6c Detail Structure of LFLANN

The learning rate, μ for LFLANN is set at 0.02. The number of iterations is kept low i.e. 10 due to a faster rate of convergence because of the single layer architecture. The LFLANN algorithm almost converges after 10 iterations as illustrated in Fig. 5.7. The number of iterations is considerably less than MLP which is in the order of 1000. Therefore, the proposed algorithm requires much less computational time for training than MLP and other ANN structures.

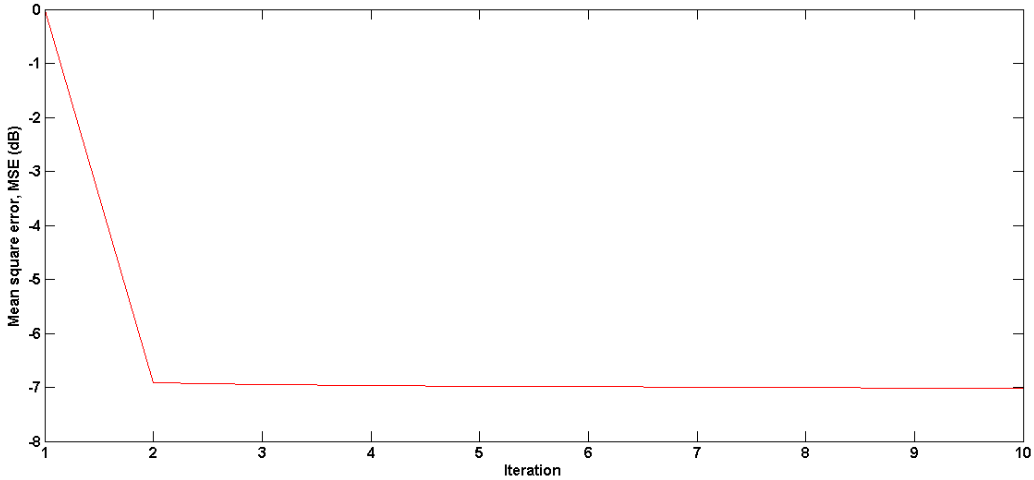


Fig. 5.7 Convergence characteristics of LFLANN algorithm

For proper training of the neural network, an image which is rich in different patterns is taken. Therefore, Lena image is used for training purpose which is rich in different low, medium and high frequency patterns. But in general, this neural network can be trained with any image and can be tested with any image. The functional expansion of the input pattern vector using Legendre polynomials is explained below.

In this proposed scheme, LFLANN is used for predicting the HF information in an up-sampled, blurred image. A 3×3 neighbourhood of an up-scaled image is taken as input. So, there will be nine pixel values in the input pattern. So, the nine dimensional input pattern vector is given by,

$$X_i = [\ x_1 \quad x_2 \ \cdots \quad x_9\ ] \tag{5.19}$$

The input vector is expanded using Legendre polynomial based nonlinear function expansion. It may be observed from the Table 5.2 that, for different images, peak performance are attained at the four degree of expansion. Hence, the overall degree of expansion is kept four. The Legendre polynomials are denoted by $L_n(X)$, where n is the order and $-1 < x < 1$ is the argument of the polynomial. The generalized mathematical expression of the Legendre polynomial expansion is given by,

$$L_{n+1}(x) = \frac{1}{n+1}[\ (2n+1)x\, L_n(x) - n\, L_{n-1}(x)\ ] \tag{5.20}$$

The zero and first-order Legendre polynomials are:

$$L_0(x) = 1 \tag{5.21a}$$

$$L_1(x) = x \tag{5.21b}$$

The higher order Legendre polynomials are given by,

$$L_2(x) = \frac{1}{2}(3x^2 - 1) \tag{5.22a}$$

$$L_3(x) = \frac{1}{2}(5x^3 - 3x) \tag{5.22b}$$

$$L_4(x) = \frac{1}{8}(35x^4 - 30x^2 + 3) \tag{5.22c}$$

$$L_5(x) = 7.875x^5 - 15.312x^3 + 1.875x \tag{5.22d}$$

In this case, the degree of expansion is taken as four for optimum performance. The functional expansion of the nine dimensional input pattern using Legendre polynomial is shown in Figure 5.6 and is given by,

$$X_{iL} = [1\ x_1\ L_2(x_1)\ L_3(x_1)\ L_4(x_1)\ x_2\ L_2(x_2)\ L_3(x_2)\ L_4(x_2) \ldots x_9\ L_2(x_9)\ L_3(x_9)\ L_4(x_9)\ ] \tag{5.23}$$

5.3.1 LFLANN Algorithm

Let $g_u(x, y)$ denote an up-scaled image using Lanczos-3 interpolation. Let $\hat{f}(x, y)$ denote the de-blurred, restored image using LFLANN algorithm. Let them of size $(2P \times 2Q)$. X_i denotes the nine-dimensional input pattern vector which represents nine pixels of a 3×3 neighbourhood of a blurred up-scaled image. The enhanced pattern with Legendre expansion is denoted by X_{iL}. The degree of expansion, N for LFLANN is taken as four so as to provide optimum performance. The LFLANN algorithm consists of two phases namely training and testing phase which are given below.

Training Phase:

Step-1. Initialize the number of iterations to ten.

$$iteration = 10$$

Step-2. Initialize the learning rate parameter, μ.

$$\mu = 0.02$$

Step-3. Select a 3×3 window, w in an up-scaled image, $g_u(x, y)$.

$$w_{s,t}(x, y), \quad -1 \le s, t \le 1$$

Step-4. All the elements of the weight vector, w_i are initialized to 1.

Step-5. Generate the enhanced input pattern vector, X_{iL} from the nine-dimensional input pattern vector, X_i using (5.20) and (5.23).

Step-6. Calculate the output of the LFLANN structure by:

$$O_i = \tanh\left(\sum_{i=1}^{37} w_i \, X_{iL}\right) \quad (5.24)$$

Step-7. Calculate the output error by:

$$e_i = D_i - O_i \quad (5.25)$$

where, D_i is the desired output (pixel value).

Step-8. Update the weight vector using the least mean square (LMS) algorithm given by:

$$w_i(l+1) = w_i(l) + \mu \, e_i(l) \, X_i(l) \quad (5.26)$$

where, l is the time index or iteration.

Step-9. Repeat Step-3 to Step-8 for all the (x, y) locations of the up-scaled image, $g_u(x, y)$.

Step-10. After the completion of the learning phase, the final updated weights are kept fixed and are subsequently used in the testing phase for image de-blurring operation.

Testing Phase:

Step-11. Select a 3×3 window, w in another up-scaled image, $g_{u1}(x, y)$.

Step-12. Generate the enhanced input pattern vector, X_{iL} from the nine-dimensional input pattern vector, X_i using (5.20) and (5.23).

Step-13. Calculate the output of the LFLANN structure using (5.24).

Step-14. Repeat Step-11 to Step-13 for all the (x, y) locations of the up-scaled image, $g_{u1}(x, y)$ to obtain the restored, de-blurred image, $\hat{f}(x, y)$.

Table 5.3 PSNR (dB) and UQI comparison of different schemes at 4:1 compression ratio for various (512×512) images

Image	**Image Metric**	**Bilinear [5]**	**Bicubic [4]**	**Lanczos3 [12]**	**DCT [74]**	**LAL [P7]**	**FWL [P8]**	**LFLANN [P9]**
Mandril	PSNR(dB)	23.045	23.630	23.859	23.925	24.066	24.178	**24.180**
	UQI	0.8957	0.9114	0.9170	0.9187	0.9228	**0.9260**	0.9250
Lena	PSNR(dB)	32.704	34.148	34.813	35.023	35.080	35.250	**35.454**
	UQI	0.9922	0.9945	0.9953	0.9955	0.9956	0.9958	**0.9960**
Barbara	PSNR(dB)	24.925	25.352	25.428	25.183	25.562	**25.692**	25.630
	UQI	0.9631	0.9669	0.9675	0.9657	0.9687	**0.9698**	0.9692
Boat	PSNR(dB)	28.940	29.952	30.375	30.466	30.841	**30.870**	30.848
	UQI	0.9801	0.9845	0.9860	0.9863	0.9876	0.9877	**0.9878**
Goldhill	PSNR(dB)	30.574	31.405	31.725	31.716	31.993	**32.092**	32.027
	UQI	0.9880	0.9901	0.9909	0.9909	0.9915	**0.9917**	0.9914
Pirate	PSNR(dB)	30.027	31.058	31.490	31.606	31.952	31.986	**32.039**
	UQI	0.9853	0.9885	0.9897	0.9899	0.9908	0.9909	**0.9909**
Livingroom	PSNR(dB)	28.617	29.557	29.977	30.128	30.311	**30.439**	30.308
	UQI	0.9761	0.9811	0.9829	0.9835	0.9843	**0.9850**	0.9845
Fingerprint	PSNR(dB)	28.045	30.632	31.722	**32.133**	31.138	31.345	31.966
	UQI	0.9785	0.9889	0.9915	0.9922	0.9908	0.9913	**0.9924**
Baboon	PSNR(dB)	33.588	35.014	35.662	35.890	35.850	**36.042**	35.737
	UQI	0.9924	0.9946	0.9954	0.9956	0.9956	**0.9958**	0.9955
Bridge	PSNR(dB)	25.728	26.504	26.826	26.919	27.160	**27.255**	27.102
	UQI	0.9693	0.9748	0.9767	0.9772	0.9787	**0.9793**	0.9782
Cameraman	PSNR(dB)	33.214	35.757	37.216	**37.832**	36.963	37.381	35.979
	UQI	0.9959	0.9977	0.9984	**0.9986**	0.9983	0.9985	0.9982
Cat	PSNR(dB)	30.949	31.982	32.427	32.563	32.584	**32.826**	31.842
	UQI	0.9920	0.9937	0.9943	0.9945	0.9946	**0.9949**	0.9938
Crowd	PSNR(dB)	30.984	32.666	33.451	33.768	**34.035**	34.026	33.664
	UQI	0.9894	0.9930	0.9942	0.9946	0.9949	**0.9950**	0.9944
Cycle	PSNR(dB)	21.208	21.895	22.154	22.129	22.452	22.543	**22.960**
	UQI	0.9323	0.9437	0.9475	0.9474	0.9526	0.9538	**0.9567**
F16	PSNR(dB)	30.379	31.543	32.104	32.722	32.353	32.319	**32.790**
	UQI	0.9848	0.9885	0.9900	0.9913	0.9907	0.9907	**0.9916**
House	PSNR(dB)	29.248	30.314	30.807	30.862	30.980	**31.304**	30.855
	UQI	0.9719	0.9785	0.9809	0.9812	0.9820	**0.9835**	0.9817
Lake	PSNR(dB)	28.945	30.022	30.495	30.793	30.617	30.700	**30.930**
	UQI	0.9895	0.9919	0.9928	0.9933	0.9930	0.9932	**0.9935**
Peppers	PSNR(dB)	31.180	31.991	32.328	**32.747**	32.509	32.407	32.157
	UQI	0.9923	0.9937	0.9942	**0.9947**	0.9944	0.9943	0.9941
Elaine	PSNR(dB)	32.534	33.117	33.309	33.284	33.315	**33.413**	33.244
	UQI	0.9913	0.9924	0.9928	0.9927	0.9928	**0.9930**	0.9927
Ruler	PSNR(dB)	12.335	12.613	12.673	12.600	12.584	12.823	**13.415**
	UQI	0.5188	0.5735	0.5898	0.5920	0.6088	0.6371	**0.6484**

Table 5.4 PSNR (dB) and UQI comparison of different schemes at 16:1 compression ratio for various (512×512) images

Image	**Image Metric**	**Bilinear [5]**	**Bicubic [4]**	**Lanczos3 [12]**	**DCT [74]**	**LAL [P7]**	**FWL [P8]**	**LFLANN [P9]**
Mandril	PSNR(dB)	20.883	21.085	21.156	21.167	21.181	21.210	**21.282**
	UQI	0.8191	0.8309	0.8351	0.8362	0.8365	0.8383	**0.8424**
Lena	PSNR(dB)	28.053	28.848	29.183	29.296	29.389	29.407	**29.794**
	UQI	0.9767	0.9810	0.9825	0.9829	0.9833	0.9835	**0.9848**
Barbara	PSNR(dB)	23.351	23.607	23.708	23.747	23.755	23.765	**23.834**
	UQI	0.9457	0.9496	0.9510	0.9515	0.9516	0.9519	**0.9526**
Boat	PSNR(dB)	25.041	25.538	25.739	25.773	25.849	25.880	**26.137**
	UQI	0.9493	0.9557	0.9580	0.9584	0.9592	0.9596	**0.9624**
Goldhill	PSNR(dB)	27.166	27.628	27.798	27.776	27.878	27.913	**28.002**
	UQI	0.9731	0.9761	0.9771	0.9770	0.9775	0.9778	**0.9779**
Pirate	PSNR(dB)	26.286	26.861	27.083	27.127	27.228	27.243	**27.503**
	UQI	0.9641	0.9691	0.9708	0.9712	0.9719	0.9721	**0.9737**
Livingroom	PSNR(dB)	24.932	25.385	25.560	25.563	25.634	25.674	**25.800**
	UQI	0.9420	0.9488	0.9511	0.9513	0.9521	0.9527	**0.9544**
Fingerprint	PSNR(dB)	20.920	22.633	23.779	24.202	24.218	24.329	**24.468**
	UQI	0.8791	0.9192	0.9407	0.9467	0.9478	0.9496	**0.9542**
Baboon	PSNR(dB)	28.812	29.604	29.931	30.007	30.018	30.118	**30.207**
	UQI	0.9766	0.9808	0.9823	0.9826	0.9827	0.9831	**0.9835**
Bridge	PSNR(dB)	22.676	23.066	23.230	23.263	23.297	23.331	**23.371**
	UQI	0.9357	0.9423	0.9449	0.9454	0.9459	0.9465	**0.9466**
Cameraman	PSNR(dB)	26.633	27.546	27.946	28.094	28.264	28.261	**28.681**
	UQI	0.9809	0.9847	0.9861	0.9866	0.9871	0.9872	**0.9887**
Cat	PSNR(dB)	27.295	27.838	28.059	28.105	28.141	**28.201**	28.191
	UQI	0.9812	0.9835	0.9844	0.9845	0.9847	0.9849	**0.9852**
Crowd	PSNR(dB)	25.759	26.601	26.974	27.066	27.205	27.217	**27.359**
	UQI	0.9634	0.9706	0.9733	0.9739	0.9748	0.9750	**0.9756**
Cycle	PSNR(dB)	18.693	18.996	19.109	19.135	19.167	19.198	**19.330**
	UQI	0.8727	0.8842	0.8883	0.8893	0.8903	0.8915	**0.8935**
F16	PSNR(dB)	25.767	26.457	26.773	26.939	26.965	26.963	**27.150**
	UQI	0.9540	0.9617	0.9647	0.9661	0.9664	0.9665	**0.9684**
House	PSNR(dB)	25.401	25.863	26.050	26.063	26.155	26.183	**26.345**
	UQI	0.9285	0.9373	0.9406	0.9409	0.9423	0.9429	**0.9463**
Lake	PSNR(dB)	24.783	25.466	25.746	25.834	25.911	25.924	**26.207**
	UQI	0.9720	0.9764	0.9780	0.9785	0.9789	0.9790	**0.9803**
Peppers	PSNR(dB)	27.359	28.045	28.330	28.524	28.547	28.496	**28.604**
	UQI	0.9811	0.9841	0.9852	0.9859	0.9860	0.9858	**0.9863**
Elaine	PSNR(dB)	29.771	30.496	30.785	30.884	**30.984**	30.959	30.883
	UQI	0.9832	0.9860	0.9870	**0.9878**	0.9876	0.9857	0.9872
Ruler	PSNR(dB)	10.707	10.804	10.873	10.876	10.873	**10.884**	10.843
	UQI	0.2036	0.2394	0.2608	0.2636	0.2638	**0.2691**	0.2252

Table 5.5 Average PSNR (dB) and UQI comparison of different interpolation techniques at 4:1 compression ratio for various sequences over 50 frames

Sequence	Image Metric	Bilinear [5]	Bicubic [4]	Lanczos3 [12]	DCT [74]	LAL [P7]	FWL [P8]	LFLANN [P9]
Salesman	PSNR(dB)	28.107	28.979	29.327	29.440	29.744	**29.804**	29.723
	UQI	0.9649	0.9719	0.9742	0.9750	0.9770	**0.9776**	0.9771
Bus	PSNR(dB)	24.243	25.262	25.716	25.794	26.177	26.362	**26.393**
	UQI	0.9491	0.9610	0.9652	0.9660	0.9698	**0.9713**	0.9706
Akiyo	PSNR(dB)	31.686	32.911	33.450	33.647	**34.187**	34.110	33.963
	UQI	0.9927	0.9945	0.9952	0.9954	**0.9959**	0.9959	0.9957
City	PSNR(dB)	26.827	27.592	27.879	27.852	28.248	28.358	**28.380**
	UQI	0.9104	0.9277	0.9333	0.9333	0.9407	**0.9433**	0.9421
Container	PSNR(dB)	24.663	25.567	26.008	26.256	26.376	**26.448**	26.261
	UQI	0.9592	0.9676	0.9709	0.9727	0.9739	**0.9746**	0.9729
Football	PSNR(dB)	27.106	28.568	29.366	29.687	29.983	**30.178**	30.164
	UQI	0.9668	0.9769	0.9810	0.9824	0.9840	**0.9849**	0.9846
Mobile	PSNR(dB)	20.310	21.197	21.596	21.758	22.199	22.239	**22.400**
	UQI	0.9373	0.9502	0.9550	0.9569	0.9622	0.9628	**0.9635**
Soccer	PSNR(dB)	29.271	30.254	30.664	30.748	31.020	**31.166**	30.943
	UQI	0.9828	0.9863	0.9876	0.9878	0.9887	**0.9891**	0.9884
Coast	PSNR(dB)	25.585	26.500	26.939	27.080	27.255	**27.575**	27.389
	UQI	0.9692	0.9754	0.9779	0.9787	0.9797	**0.9814**	0.9804

Table 5.6 Execution time of the proposed and existing algorithms at 4:1 CR

Images of different size (M×N)	Execution time in Seconds						
	Bilinear [5]	**Bicubic [4]**	**Lanczos3 [12]**	**DCT [74]**	**LAL [P7]**	**FWL [P8]**	**LFLANN [P9]**
Clock (200×200)	0.0141	0.0150	0.0151	0.0518	3.1982	8.6076	0.8933
Lena (256×256)	0.0147	0.0152	0.0157	0.0633	7.7949	21.5439	1.4455
Fruit (377×321)	0.0153	0.0163	0.0168	0.1629	56.9265	82.9838	2.6542
Lena (512×512)	0.0201	0.0215	0.0222	0.1773	335.1430	384.0769	5.6765
Pentagon (1024×1024)	0.0354	0.0409	0.0442	0.6372	6140.0514	6399.0850	22.7194

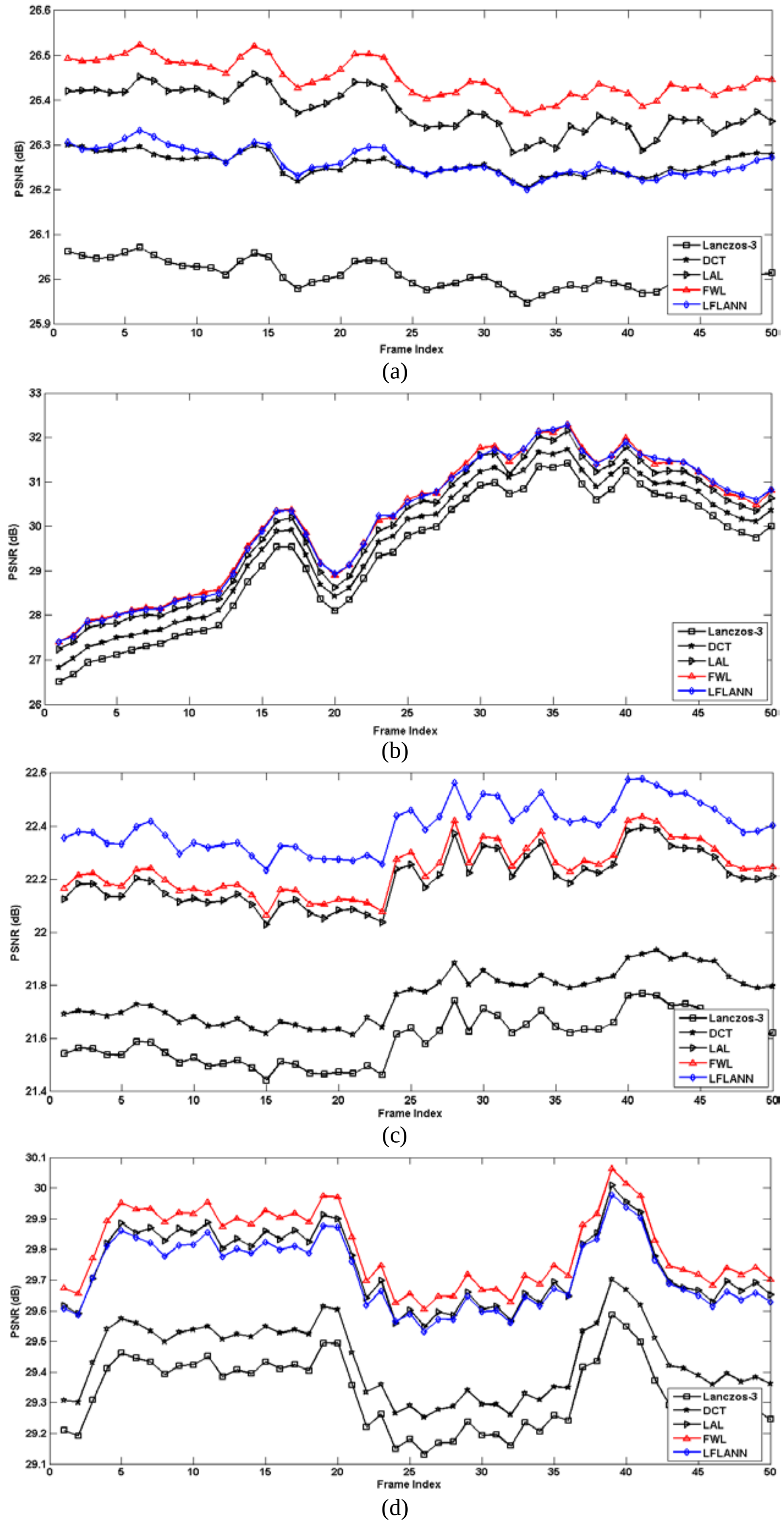


Fig. 5.8 PSNR (dB) comparisons of various up-sampling schemes at 4:1 CR, meant for different sequences: (a) Container; (b) Football; (c) Mobile; (d) Salesman

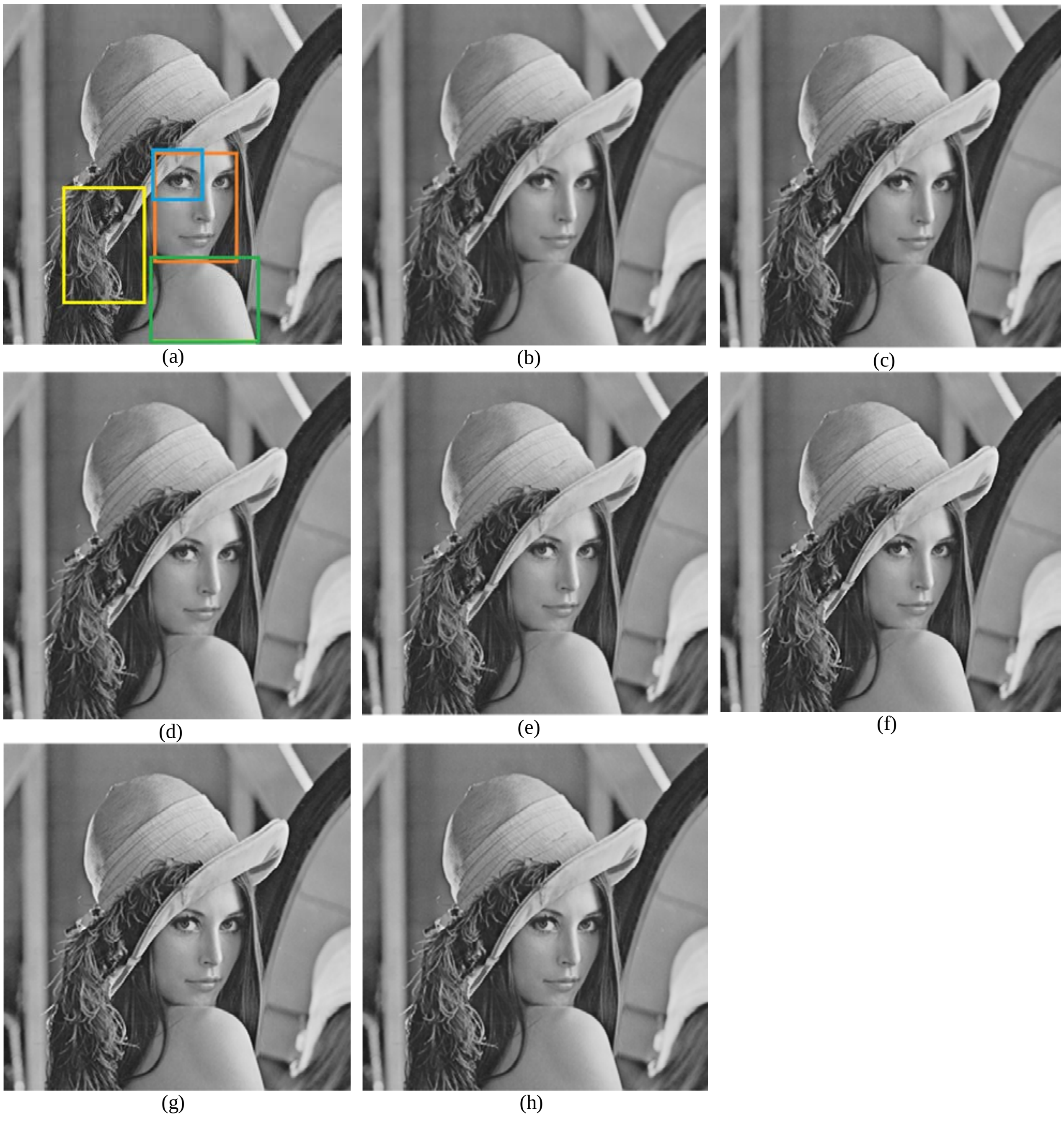


Fig. 5.9 Subjective evaluation of Lena (512×512) image using various up-sampling schemes at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) LAL (P7); (g) FWL (P8) ; (h) LFLANN (P9)

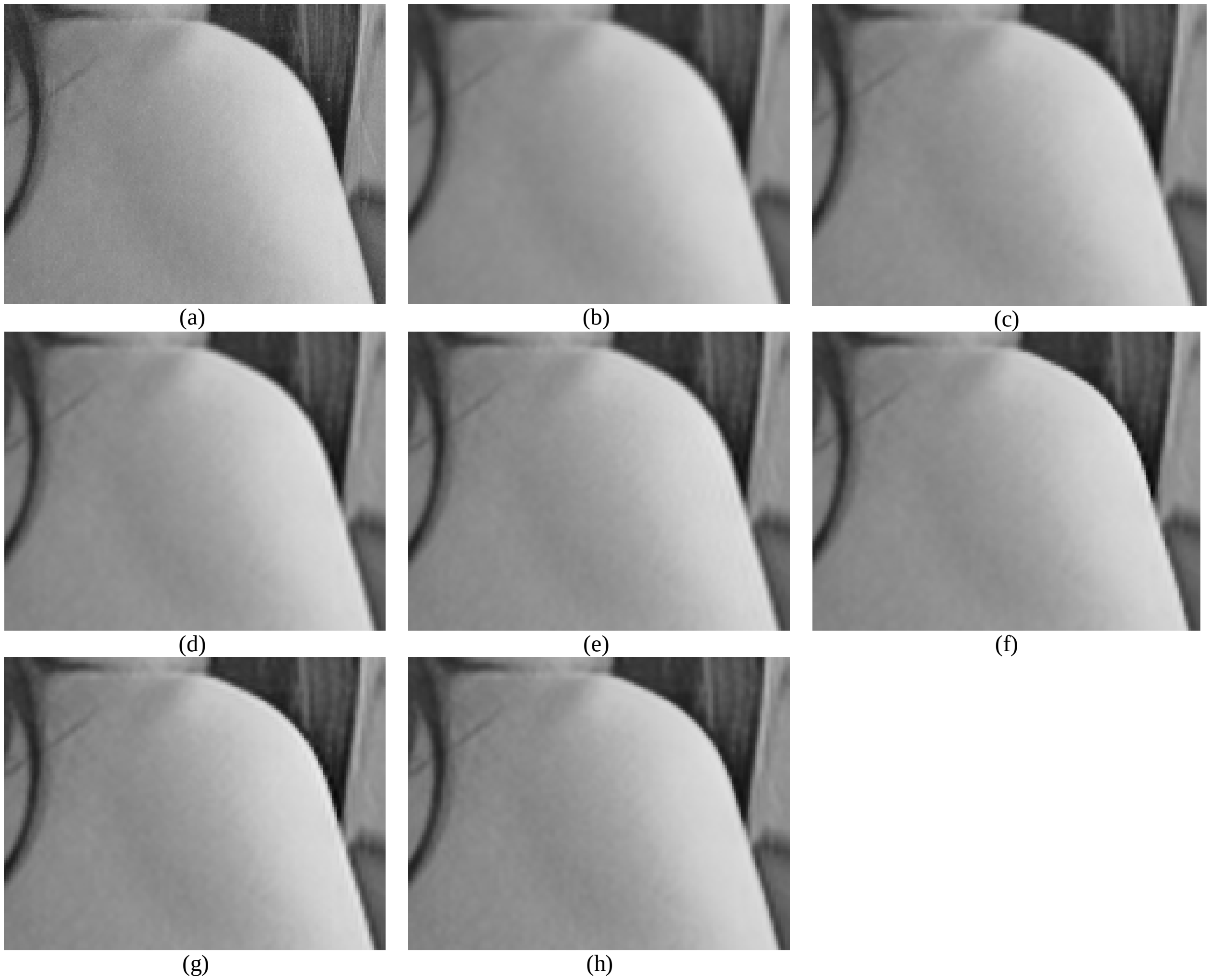


Fig. 5.10 Subjective evaluation of the selected low frequency green rectangular region (127×164) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) LAL (P7); (g) FWL (P8); (h) LFLANN (P9)

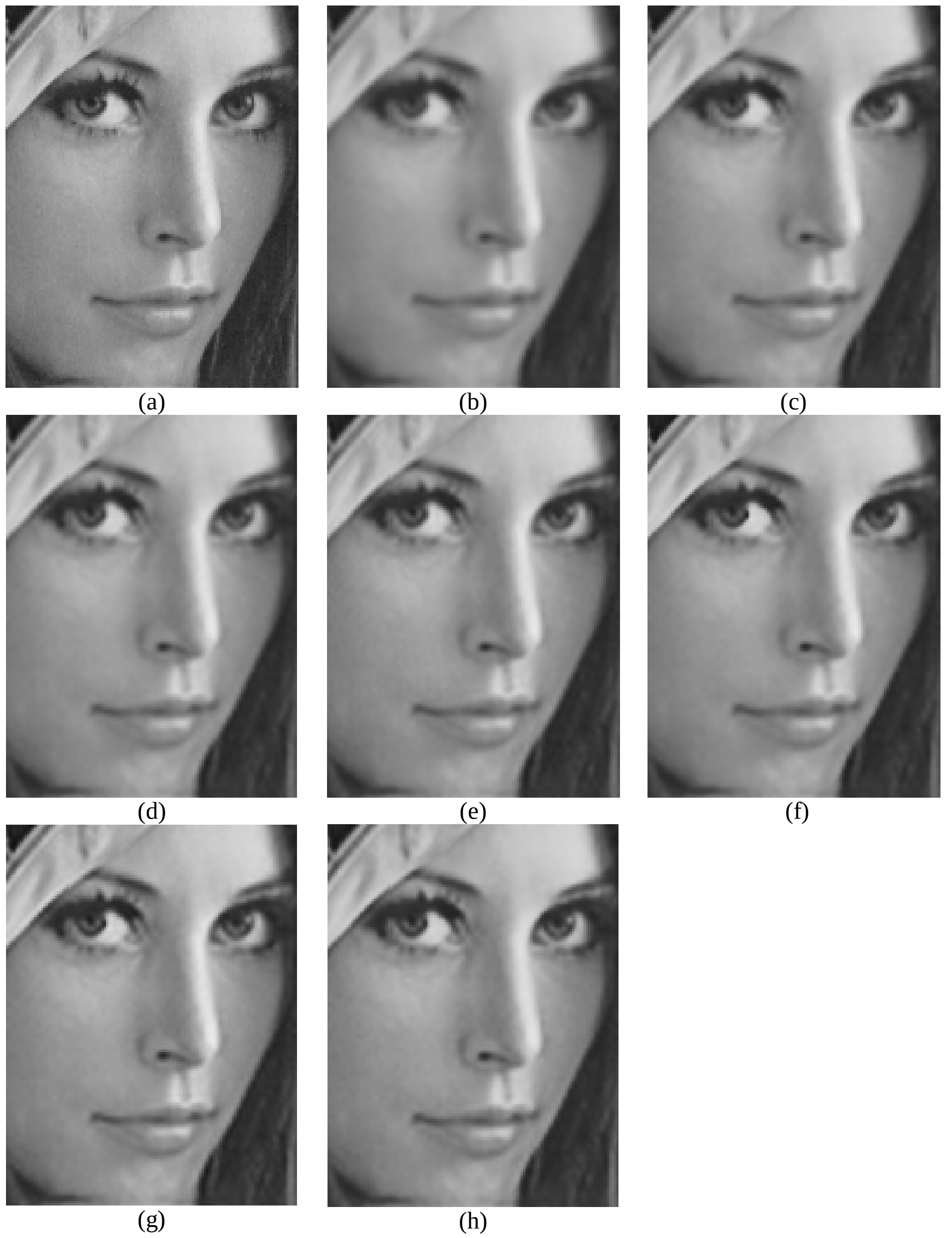


Fig. 5.11 Subjective evaluation of the selected medium frequency orange rectangular region (164×125) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) LAL (P7); (g) FWL (P8); (h) LFLANN (P9)

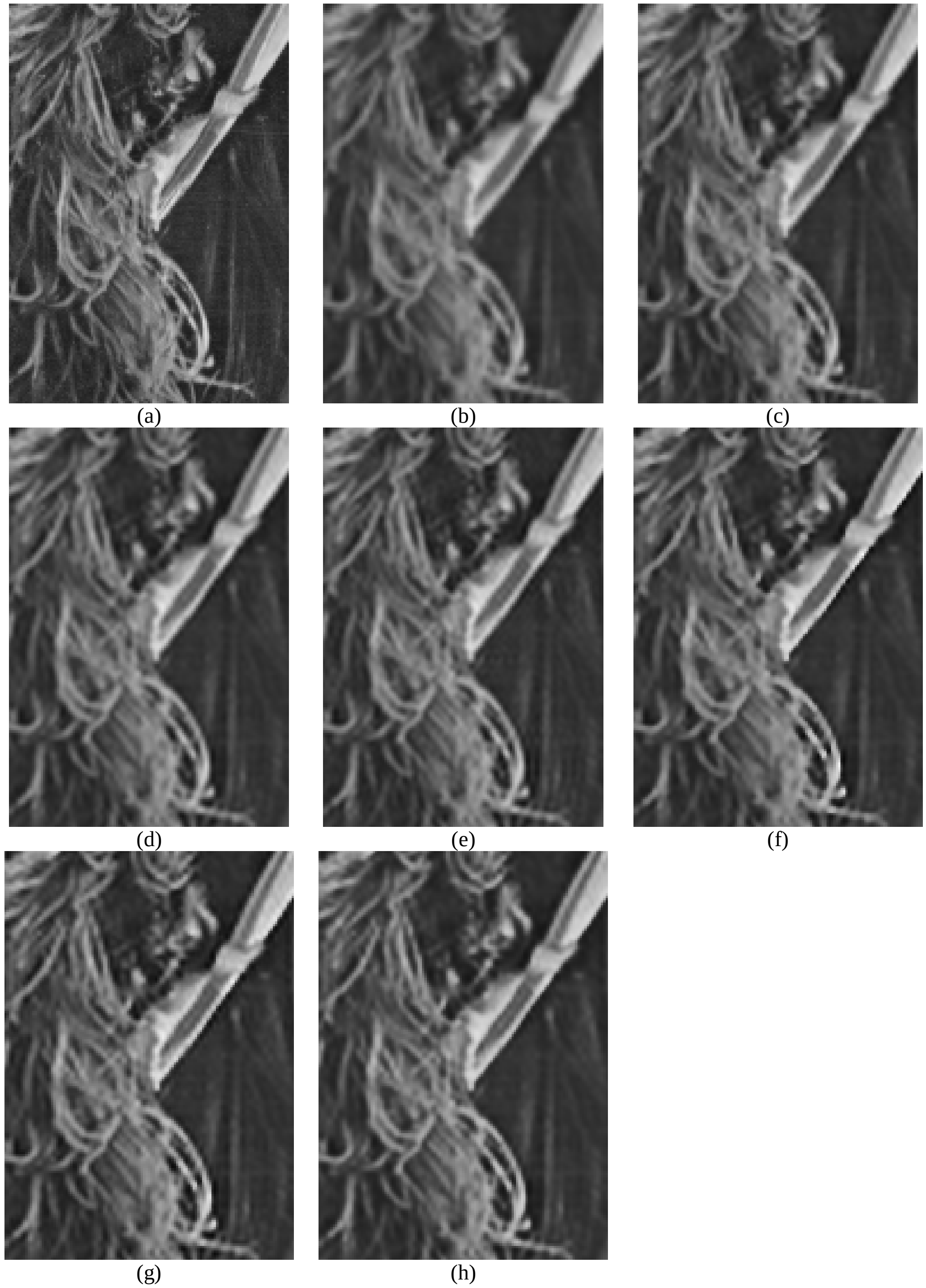


Fig. 5.12 Subjective evaluation of the selected high frequency yellow rectangular region (123×174) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) LAL (P7); (g) FWL (P8); (h) LFLANN (P9)

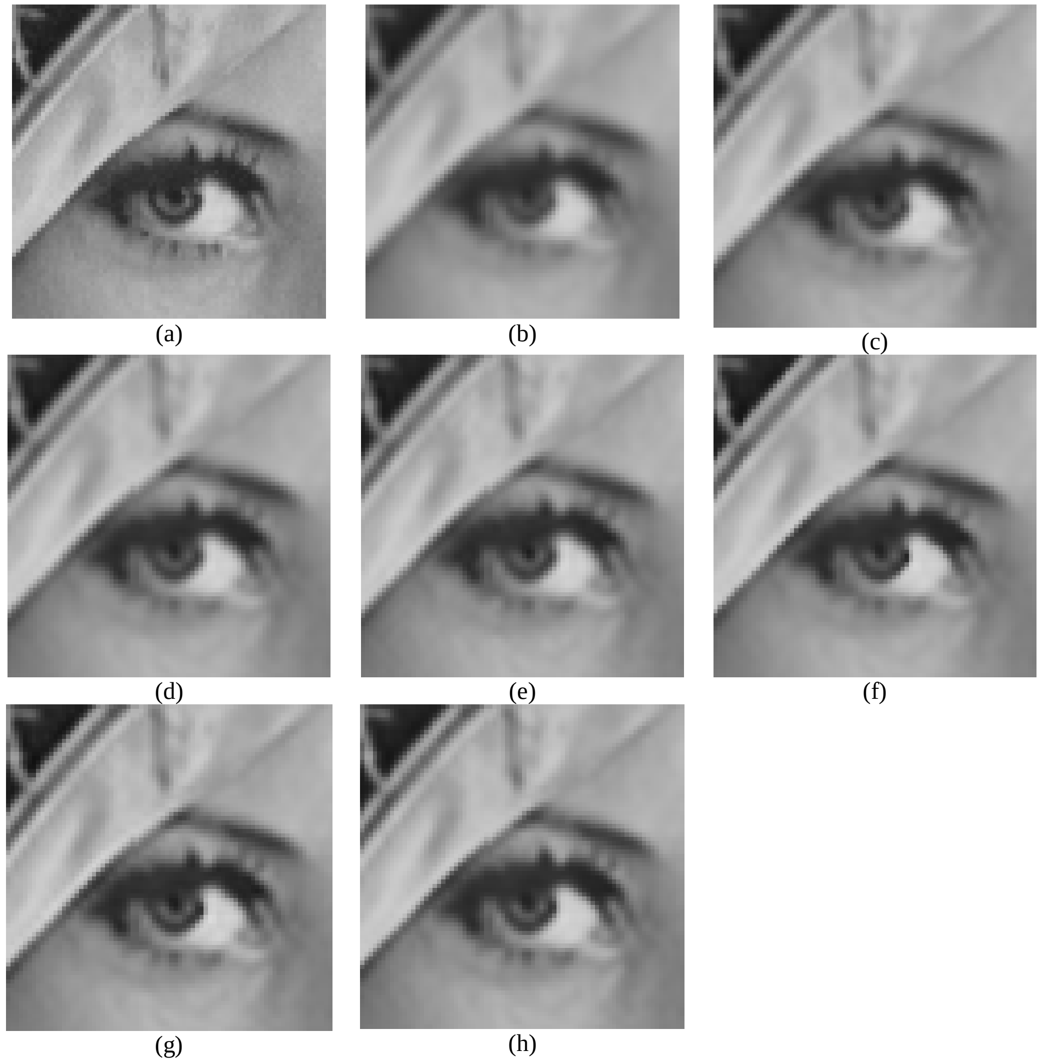


Fig. 5.13 Subjective evaluation of the selected blue rectangular (76×76) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) LAL (P7); (g) FWL (P8); (h) LFLANN (P9)

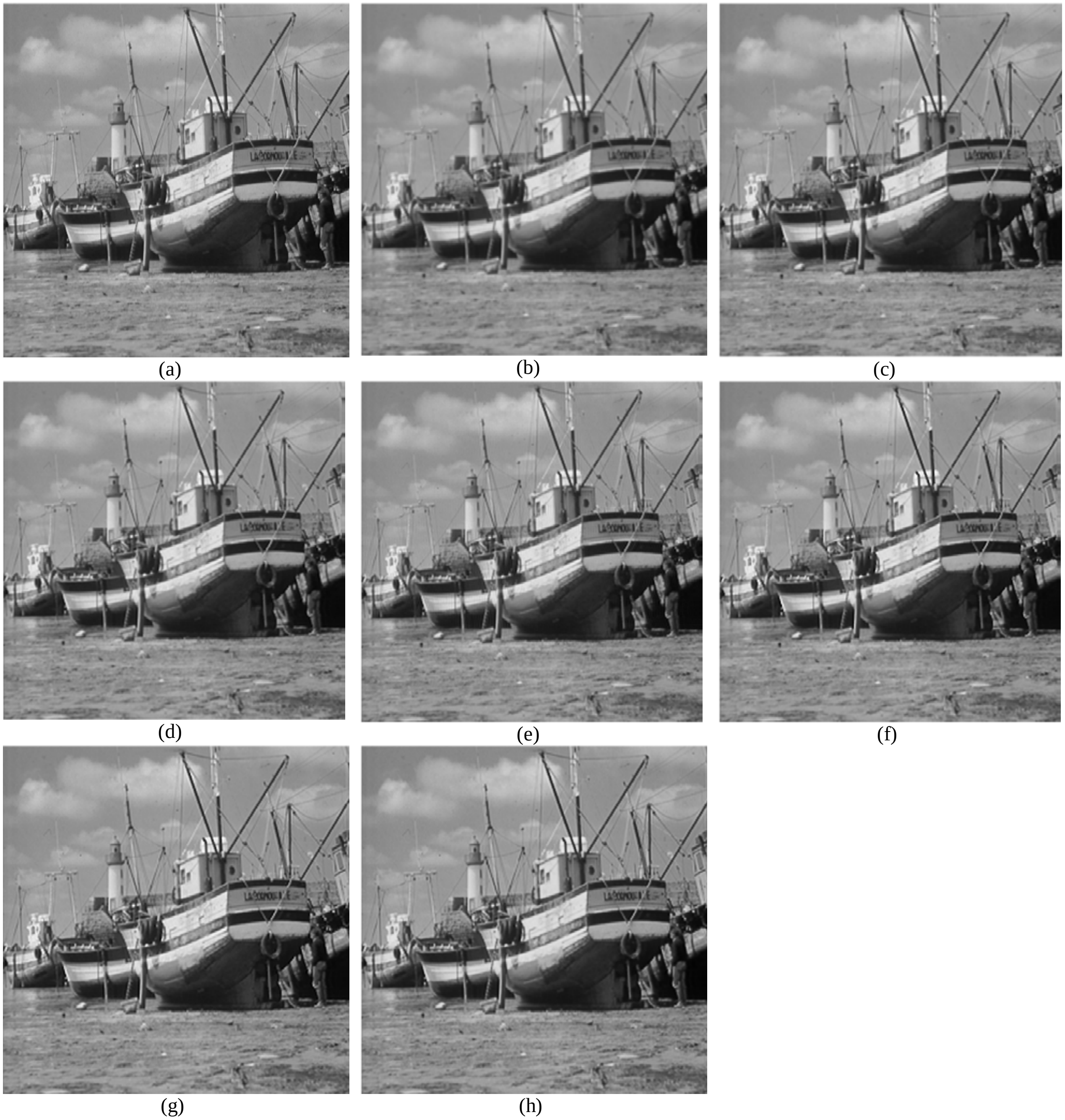


Fig. 5.14 Subjective evaluation of Boat (512×512) image using various up-sampling schemes at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) LAL (P7); (g) FWL (P8); (h) LFLANN (P9)

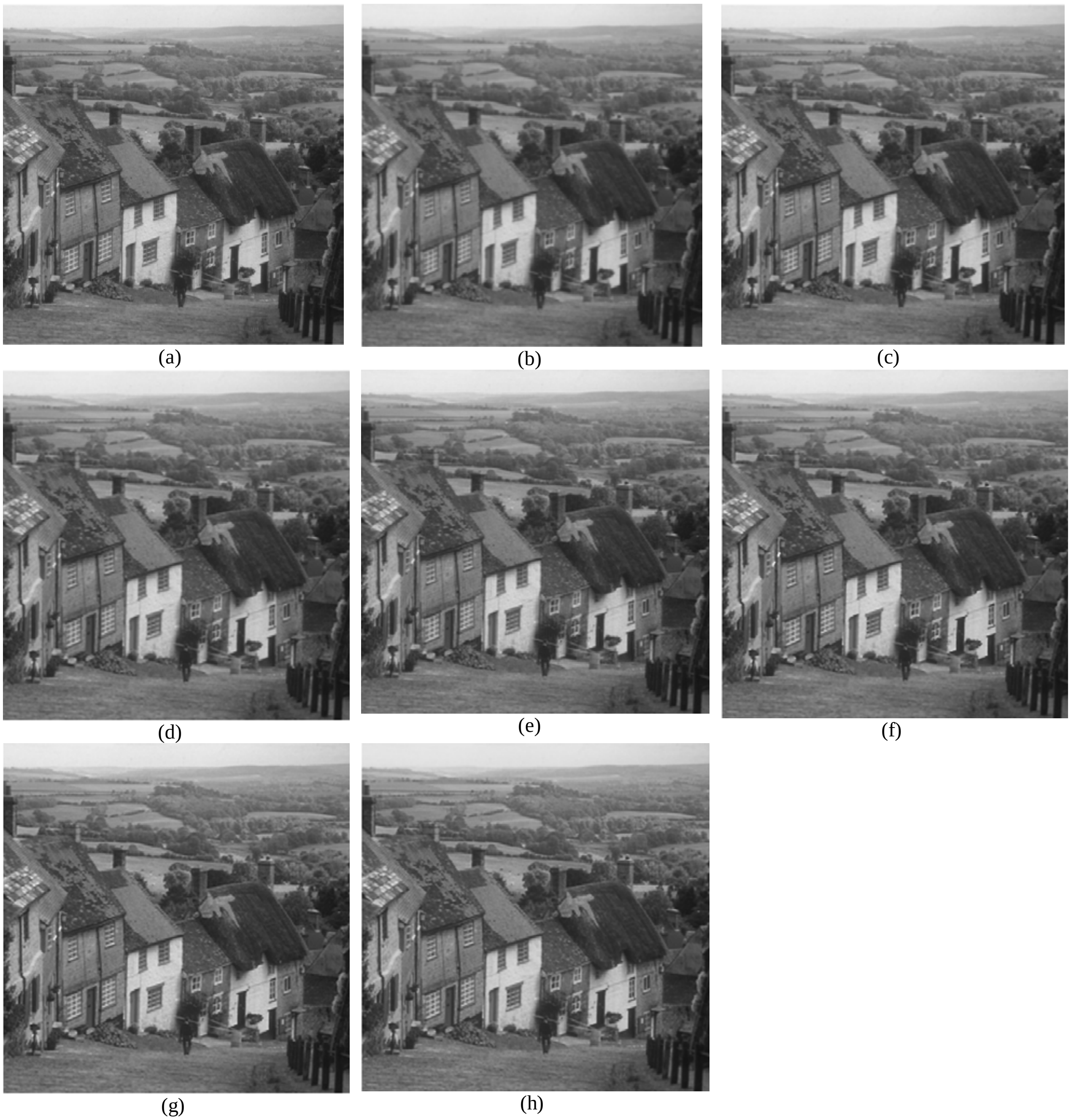


Fig. 5.15 Subjective evaluation of Goldhill (512×512) image using various up-sampling schemes at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) LAL (P7); (g) FWL (P8); (h) LFLANN (P9)

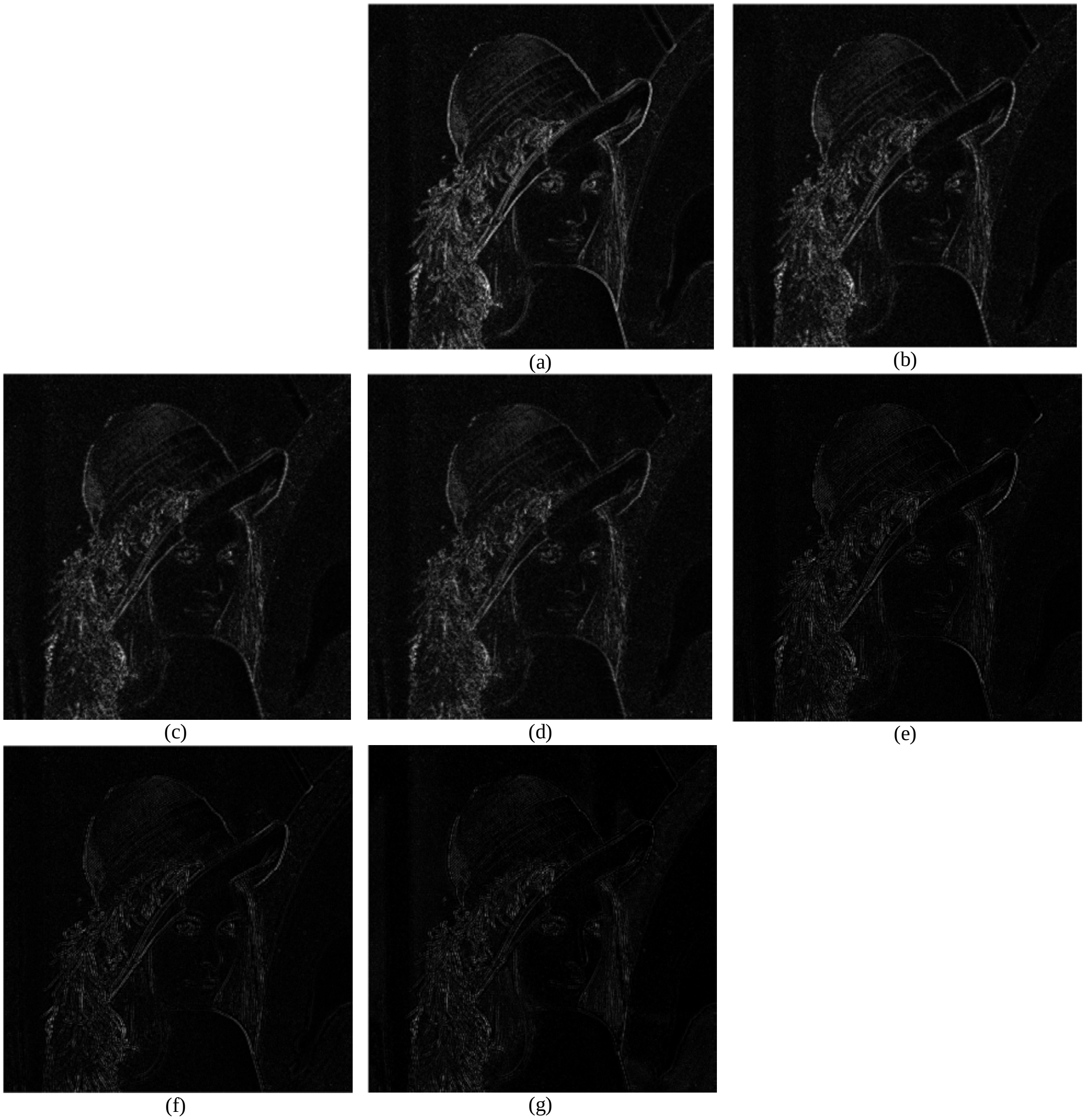


Fig. 5.16 Error image of Lena (512×512) using various up-sampling schemes at 4:1 compression ratio: (a) Bilinear; (b) Bicubic; (c) Lanczos-3; (d) DCT; (e) LAL (P7); (f) FWL (P8); (g) LFLANN (P9)

5.4 Results and Discussion

The performance of the proposed algorithms is analyzed in terms of objective and subjective measures. The computational complexities of the algorithms are evaluated in terms of CPU execution time for images of different dimensions. The error images corresponding to different algorithms are generated to show the degree of HF restoration. The figures and tables showing the performance of the existing and proposed algorithms are explained below.

Table 5.3 and Table 5.4 show the objective performance in terms of PSNR (dB) and UQI of the proposed and existing algorithms at 4:1 and 16:1 compression ratio, respectively. Table 5.5 reveals the average PSNR and UQI comparison of different algorithms for various video sequences at 4:1 compression ratio. PSNR (dB) vs. frame index plot corresponding to different sequences is given in Fig. 5.8. The execution time of different algorithms is shown in Table 5.6.

The subjective performance of Lena (512×512), Boat (512×512) and Goldhill (512×512) images at 4:1 compression ratio is shown in Fig. 5.9, Fig. 5.14 and Fig. 5.15 using various up-sampling schemes. In case of Fig. 5.9, four distinct regions with different features and thus different signal characteristics such as low, medium, high and their combinations are marked. Performance at these regions is analyzed. For this purpose, the output images at these regions are enlarged and shown in Fig. 5.10, Fig. 5.11, Fig. 5.12 and Fig. 5.13. The error images of Lena (512×512) corresponding to various schemes are given in Fig. 5.16.

The proposed algorithms show noticeable PSNR gain over the existing algorithms at 4:1 compression ratio. The LAL scheme achieves better objective performance than DCT and Lanczos-3 for most of the images. It achieves the maximum PSNR gain of 0.584 dB and 0.267 dB than Lanczos-3 and DCT in case of crowd image. Similarly for other images, LAL shows improved PSNR gain because of the employment a local adaptive post-processing algorithm whose performance varies as per the local statistics of an image. It shows PSNR improvement irrespective of variation in compression ratio because of its adaptability to varying constraints.

The performance of FWL scheme is better than LAL because of the deployment of fuzzy based mapping scheme as illustrated in Table 5.3 and Table 5.4. The FWL scheme shows peak performance amongst various existing and proposed techniques for most of the images. In case of Boat, Living-room, House and Lena, FWL shows PSNR improvement of 0.404 dB, 0.311dB, 0.442 dB and 0.23 dB, respectively than DCT at 4:1 compression ratio. It also shows better performance than LAL for rest of the images except the Crowd image. FWL achieves further improvement is due to its fuzzy based nonlinear mapping instead of the linear mapping. LAL algorithm makes use of normalized mapping which is linear. In contrast, FWL uses fuzzy based mapping, which is nonlinear and predicts the HF components more accurately than LAL.

In addition, the LFLANN scheme shows better performance than FWL at 4:1 compression ratio for images like Mandril, Lena, Pirate, Cycle, F16, Lake and Ruler which are rich in HF components. It achieves better PSNR improvement over Lanczos-3 and DCT for almost all images. In case of images like Lena, Pirate, Cycle and Ruler, the LFLANN scheme achieves PSNR gain of 0.431dB, 0.433 dB, 0.831dB and 0.815 dB, respectively over DCT at 4:1 compression ratio. At 16:1 compression ratio, LFLANN outperforms almost all existing and proposed algorithms in terms of PSNR and UQI as depicted in Table 5.4. So, the performance of LFLANN algorithm is better than FWL and LAL at higher compression ratio. The performance improvement is due to the nonlinearity introduced in the single layer LFLANN structure because of the functional expansion of input vector using Legendre polynomials. In case of various sequences, the proposed algorithms show better performance than the existing schemes in terms of average PSNR and UQI as illustrated in Table 5.5.

The computational complexity of the proposed and existing algorithms in terms of execution time is illustrated in Table 5.6. It may be well observed from the table that, all the proposed algorithms are computationally more complex than the existing algorithms. It is because these algorithms are two-pass post-processing schemes and operate on up-scaled images of larger dimensions. Since LAL and FWL are local based two-pass schemes, are computationally more complex than LFLANN. The FWL algorithm is computationally more complex than LAL because of deploying fuzzy based mapping. The LFLANN is computationally less complex amongst the proposed algorithms because of its single layer structure. The requirement of multiple layer is resolved by using nonlinear function expansion of the input vector. So, LFLANN is preferred because of its reduced computational complexity. Furthermore, in case of LAL and FWL, there is a significant rise in execution time with respect to image dimension as shown in Table 5.6. At lower dimension, the execution time of LAL and FWL is less and even comparable to LFLANN. However, at a higher image dimension, the execution time increases exponentially and far exceeds the execution time of LFLANN.

Fig. 5.16 shows the absolute error image of various existing and proposed algorithms. The absolute error shows about the degree of HF restoration performance of the algorithms. Reduced error employs better restoration performance and vice versa. It is apparent from the figure that the absolute error is much reduced in case of the proposed algorithms which indicate better restoration performance. The absolute error in case of FWL is less than LAL showing better restoration performance. Likewise, the absolute error is least in case of LFLANN indicating the best HF restoration performance amongst all the algorithms.

Lena (512×512) image, which has different regions of low, medium and high frequencies, is a suitable candidate for subjective evaluation of the proposed algorithms. The low, medium, high frequency regions of Lena image are considered for performance evaluation of the algorithms. The region rich in combination of different patterns is also considered for performance evaluation.

The green rectangular region (127×164) containing the shoulder portion of Lena image is considered as a low frequency region. The enlarged version of it using various algorithms is given in Fig. 5.10. Similarly, the enlarged versions of medium and high frequency regions are given in Fig. 5.11 and 5.12, respectively. The face and hair regions represent the medium and high frequencies, respectively. Fig 5.13 shows the eye and its surrounding region comprising low, medium and high frequencies.

In all these cases, it may be perceived that low frequency regions are well preserved, the mid frequency regions are moderately enhanced and the HF regions are highly emphasized so as to compensate the HF loss during the up-sampling process. The low frequency regions are well preserved because these regions are not degraded during the up-sampling process. In addition, the central kernel weight and hence the slope of LAL and FWL kernel becomes automatically low for low variance regions due to the employed mapping scheme, resulting in less degree of enhancement. However, in case of medium and high frequency regions, the central kernel weight and slope of LAL and FWL kernel becomes moderate and high using the mapping schemes so as to provide medium and high degree of HF enhancement, respectively. The performance of FWL is found to be better than LAL in terms of HF restoration and visual quality because of the improved nonlinearity. Nevertheless, LAL and FWL schemes show some blocking artifacts in the HF regions because of the adaptive kernels employed. These artifacts are removed in case of LFLANN algorithm. Though LFLANN scheme gives less PSNR gain than FWL in majority of images at 4:1 compression ratio, it has better subjective quality because of reduced blocking artifacts as demonstrated in Fig. 5.12 and Fig. 5.13.

In case of the proposed algorithms, the fine details and edge information, which represent the high frequency, are effectively enhanced resulting in a better visual quality. So, the overall subjective performance of the proposed schemes is more satisfactory than the existing schemes.

5.5 Conclusion

The proposed algorithms: LAL, FWL and LFLANN are local post-processing schemes which are employed for local HF enhancement of up-scaled images so as to lessen blurring at the edges and fast changing regions. Being the local based schemes, these algorithms tackle the local HF degradation more effectively than the existing schemes. The image degradation due to up-scaling is non-uniform and depends on the degree of local HF content. So, to deal with such situations, the feature of the LAL and FWL kernels are varied as per the local statistics to provide a more improved local HF restoration performance.

LAL being a local post-processing scheme, adaptively enhances high variance regions more than the low variance regions resulting in improved objective and subjective performance than DCT and other existing schemes as per the experimental results. However, the mapping technique introduced in this scheme is linear and therefore, the performance is further improved in FWL by employing a nonlinear fuzzy mapping. The incorporation of fuzzy rules into FWL makes the HF prediction more accurate than LAL resulting in better objective and subjective performance.

LFLANN, being a soft-computing technique, is exploited here for HF restoration in Lanczos-3 interpolated, up-scaled images. In case of almost all the images, it shows better objective performance than Lanczos-3 and DCT. It shows improved PSNR gain of 0.831dB and 0.815 dB over DCT in case of Cycle and Ruler images at 4:1 compression ratio. The PSNR gain is due to the nonlinearity introduced into the system because of nonlinear function expansion using Legendre polynomials.

All the proposed algorithms show better performance than the existing schemes as per the experimental results. Amongst the proposed algorithms, the overall objective and subjective performance of FWL is found to be superior than LAL and LFLANN in most of the images. In case of certain images which are rich in HF pattern like Lena, Pirate, Cycle and Ruler, LFLANN outperforms FWL at 4:1 compression ratio. Besides, at 16:1 compression ratio, LFLANN outperforms almost all the existing and proposed algorithms in terms of PSNR and UQI because of its high degree of nonlinearity and adaptability.

The proposed algorithms are computationally more complex than the given existing schemes. The proposed FWL and LAL are two-pass, local post-processing schemes and operate on up-scaled images of larger dimensions and hence are computationally more complex than various existing schemes like DCT and Lanczos-3. FWL consumes more computation time than LAL because of incorporating a fuzzy based mapping scheme. Hence, these post-processing schemes are not suitable for real-time applications. However, these are suitable for various off-line applications.

In contrast, LFLANN is computationally quite less complex than LAL and FWL because of its single layer architecture. It takes very less time for training due to its high convergence characteristics. It is due to its single layer LFLANN structure and LMS algorithm. The training time is reduced considerably by the substituting the requirement of multiple layers by a single layer structure by employing function expansion using Legendre polynomials. Likewise, the testing time of LFLANN is considerably reduced than FWL and LAL because of the single layer LFLANN structure and hence is suitable for real-time applications.

In addition, the proposed adaptive algorithms work efficiently for different types of images under variation in compression ratio and image resolution and hence are considered to be more versatile.

Chapter 6

Development of Some Spatial-domain Composite Algorithms

Preview

This chapter presents two spatial domain composite algorithms to tackle non-uniform blurring in an up-scaled image. Various pre-processing and post-processing algorithms, proposed in the previous chapters, show considerably high de-blurring performance under different conditions. However, there is a scope for further improvement by combining the pre-processing and post-processing algorithms. The pre-processing algorithms are basically based on boosting of HF details using various global HF enhancement schemes. On the other hand, the post-processing algorithms are primarily based on HF prediction using various local adaptive schemes. Hence, it is presumed that the fusion of both the processes would give better HF restoration performance than the proposed preceding algorithms.

One of the proposed composite schemes (CS-I) is developed by combining global iterative Laplacian (GIL) based pre-processing scheme with the local adaptive Laplacian (LAL) post-processing scheme. In this method, the HF detail of an input image is boosted up iteratively using Laplacian operator to counter blurring in the corresponding up-scaled image. Furthermore, local adaptive Laplacian operator locally predicts the HF details as per the local statistics to effectively restore the HF contents in the up-scaled image.

Another composite scheme (CS-II) is proposed which combines iterative Laplacian of Laplacian based global pre-processing (ILLGP) scheme with a newly proposed local fuzzy weighted Laplacian, CFWL post-processing scheme for more improved performance. The incorporation of fuzzy rules makes the composite scheme, CS-II more nonlinear than the former composite scheme, CS-I for better performance. The fuzzy based post-processing scheme is made more nonlinear through the variations of various parameters of the fuzzy inference system

(FIS) such as slope, width and the number of input-, and output membership functions. The effective fusion of pre-processing and post-processing operations makes the proposed scheme much effective to tackle the non-uniform blurring than the standalone pre-processing and post-processing algorithms.

Simulation results, presented at the end of the chapter, are quite encouraging.

The organisation of this chapter is given below.

- *Composite Scheme (CS-I) using Iterative Laplacian and Local Adaptive Laplacian*
- *Composite Scheme (CS-II) using Iterative Laplacian of Laplacian and Fuzzy Weighted Laplacian*
- *Results and Discussion*
- *Conclusion*

6.1 Composite Scheme (CS-I) using Iterative Laplacian and Local Adaptive Laplacian [P10]

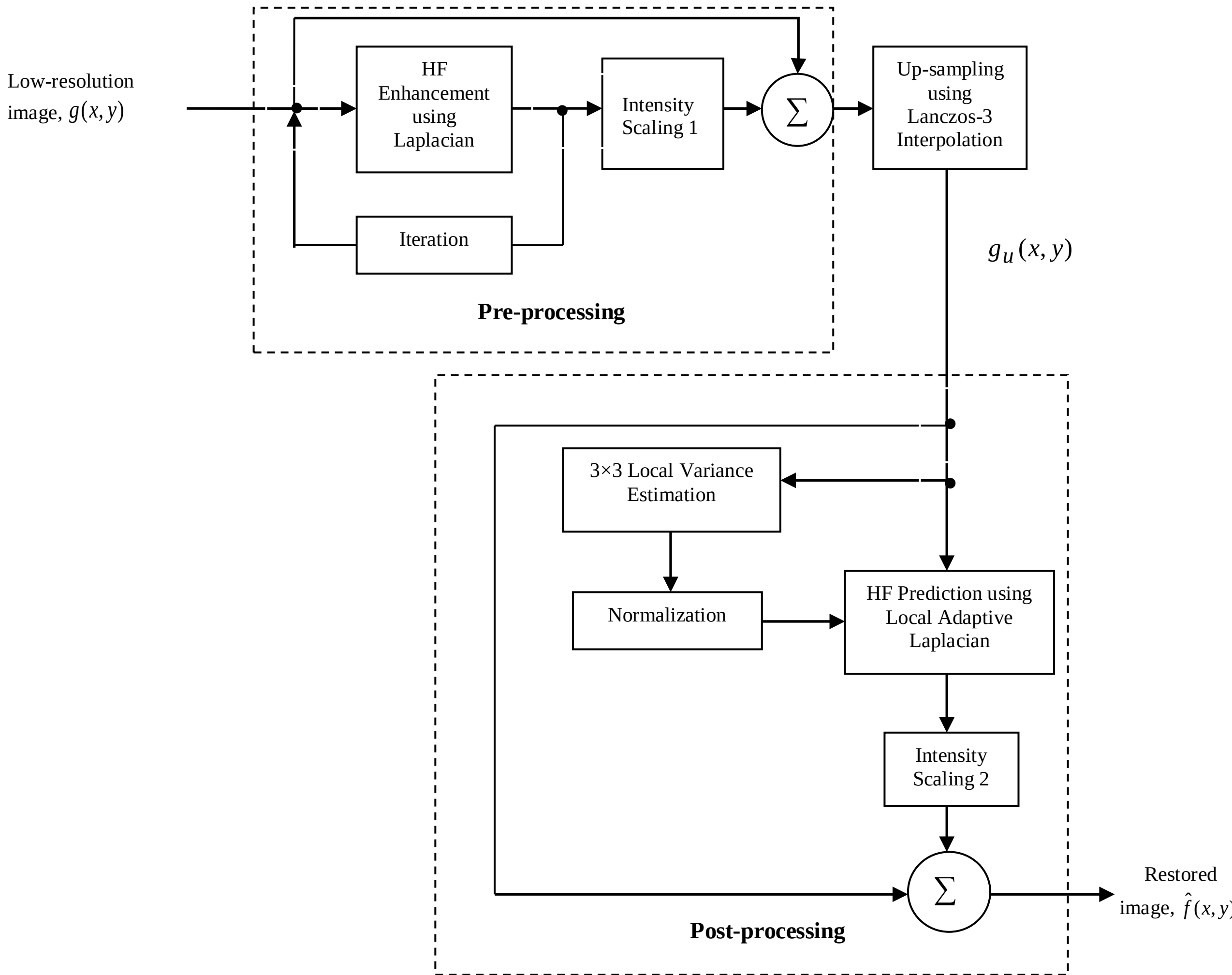


Fig. 6.1 Composite Scheme (CS-I) using Iterative Laplacian and Local Adaptive Laplacian

The proposed composite scheme (CS-I) is developed by combining a pre-processing and a post-processing operation to effectively restore HF and VHF information in an up-scaled image. CS-I exploits the advantages of either of the techniques for more improved restoration performance. The pre-processing operation makes use of global iterative Laplacian sharpening scheme prior to image up-scaling to boost up the high frequency information so as to alleviate the extent of blurring in the up-scaled images. The post-processing scheme is operated on the up-scaled images to enhance and predict the HF information using a local statistics based local adaptive Laplacian filter. The appropriate fusion of pre-processing and post-processing operations results in more accurate prediction and enhancement of high frequency to counter blurring in the up-scaled images. It is presumed that the effective fusion of both the schemes gives better HF and VHF restoration performance than the standalone schemes. The composite scheme CS-I comprises the following three steps.

1. HF enhancement using global iterative Laplacian (GIL)
2. Image up-scaling using Lanczos-3 interpolation
3. HF prediction using local adaptive Laplacian (LAL)

HF enhancement using global iterative Laplacian (GIL) is explained in the subsequent section. The Lanczos-3 interpolation and the Local adaptive Laplacian based post-processing are explained in Section 2.5.1 and Section 5.1.

6.1.1 HF Enhancement using Global Iterative Laplacian (GIL)

The composite scheme performs HF enhancement in the pre-processing step by boosting the VHF and HF details which are much more degraded than the medium and low frequency details during the up-sampling process. To reduce such HF degradation, the sub-sampled image is superimposed with HF extracts which are obtained using global iterative Laplacian operation prior to image up-scaling. The Laplacian operator being based on 2^{nd} order derivative is a high pass filter and is used to obtain the HF extract of an image. Higher the order of the derivative operator, better will be its capability to extract the finer and subtler details of an image that corresponds to VHF. The iterative Laplacian uses the Laplacian for three iterations and hence is based on 6^{th} order derivative. So, it is more capable of extracting the very fine and subtler details of an image as compare to Laplacian.

The number of iterations of the pre-processing algorithm plays a major role in determining the restoration performance of the algorithm. The high frequency restoration performance of the algorithm is found to be maximum during the third iteration and then gradually reduces towards the higher order iterations as shown in Fig. 6.2. Since the VHF and HF extracts too undergo deformation towards further higher order derivatives, the restoration performance declines after third iteration. Hence, the number of iterations is limited to three to have a better restored image quality with less high frequency deformation as illustrated in Fig. 6.2. The details of global iterative Laplacian pre-processing scheme is discussed as follows.

Let $g(x,y)$ be a sub-sampled low resolution input image. During the pre-processing operation, the sub-sampled image is linearly convolved with the Laplacian operator for three iterations to generate the VHF extracts. Let $\nabla^2 g(x,y)$ be the output obtained after convolving the sub-sampled image $g(x,y)$ with the Laplacian operator $h_{La}(x,y)$ after the first iteration, which is expressed as

$$\nabla^2 g(x,y) = \sum_{s=-1}^{1} \sum_{t=-1}^{1} h_{La}(s,t)\, g(x+s,\, y+t) \tag{6.1}$$

where, the 3×3 Laplacian operator, $h_{La}(x,y)$ is given by,

$$h_{La}(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{6.2}$$

Similarly, $\nabla^4 g(x,y)$ and $\nabla^6 g(x,y)$ are the outputs after two and three iterations, respectively. So, the HF extracts after two and three iterations are given by,

$$\nabla^4 g(x,y) = \sum_{s=-1}^{1} \sum_{t=-1}^{1} h_{La}(s,t)\, \nabla^2 g(x+s, y+t) \tag{6.3}$$

$$\nabla^6 g(x,y) = \sum_{s=-1}^{1} \sum_{t=-1}^{1} h_{La}(s,t)\, \nabla^4 g(x+s,\, y+t) \tag{6.4}$$

Let $g_s(x,y)$ be the sharpened sub-sampled image which is obtained by the weighted superimposition of HF extract after three iterations with the sub-sampled image $g(x,y)$ and is given by,

$$g_s(x,y) = g(x,y) + K_1\, \nabla^6 g(x,y) \tag{6.5}$$

where, $K_1 = 0.0028$. The term K_1 is the intensity scaling factor or the weight factor that determines the degree of sharpening in the pre-processing phase of the composite algorithm. However, the weight factor, K_1 in the standalone GIL pre-processing scheme is different from the K_1 in the composite scheme. The detail estimation of weight factor is given in Section 6.1.3.

Subsequent to image sharpening, the sharpened image, $g_s(x,y)$ is up-scaled using Lanczos-3 interpolation. The HF detail in the up-scaled image is further enhanced using local adaptive Laplacian (LAL) scheme which is illustrated in Section 5.1. However, the weight factor, K_2 of LAL post-processing scheme which is employed in the composite algorithm, is different and generally less than that of the standalone LAL scheme for effective HF prediction. It is because the desired amount of sharpening in composite scheme is shared by both of its constituent pre-processing and post-processing algorithms. The detailed estimation of weight factors K_1 and K_2 is given in Section 6.1.3.

6.1.2 CS-I Algorithm

The proposed CS-I composite algorithm comprises a pre-processing and a post-processing algorithm which are given below.

Pre-processing Algorithm:

Let $g(x,y)$ and $\nabla^6 g(x,y)$ denote the sub-sampled image and its filtered version using Laplacian after three iterations. Let them be of size $P \times Q$. The sharpened version of the sub-sampled image is denoted by $g_s(x,y)$. The Laplacian kernel used for the iterative operation is denoted by $h_L(x,y)$. The pre-processing algorithm is given below. K_1 and K_2 denote the weight factors of the pre-processing and post-processing algorithms, respectively.

Step 1. Apply the Laplacian operation, given by (6.1), on the sub-sampled image $g(x,y)$ and iterate it 3 times to obtain $\nabla^6 g(x,y)$.

Step 2. Add the weighted version of the filtered output $\nabla^6 g(x,y)$ to the original sub-sampled image $g(x,y)$ to generate a sharpened sub-sampled image, $g_s(x,y)$ using (6.5). where, $K_1 = 0.0028$

Step 3. Interpolate sub-sampled image $g_s(x,y)$ by Lanczos-3 interpolation to obtain the up-scaled image, $g_u(x,y)$ of size $(2P \times 2Q)$ using (2.3).

Post-processing Algorithm:

The post-processing algorithm is same as local adaptive Laplacian (LAL) algorithm which is given in Section 5.1.1. However, the weight factor, K_2 is different from the standalone LAL scheme and its estimation is illustrated in Section 6.1.3. The estimated value of weight factor, K_2 for the composite scheme, CS-I is given by, $K_2 = 0.2866$.

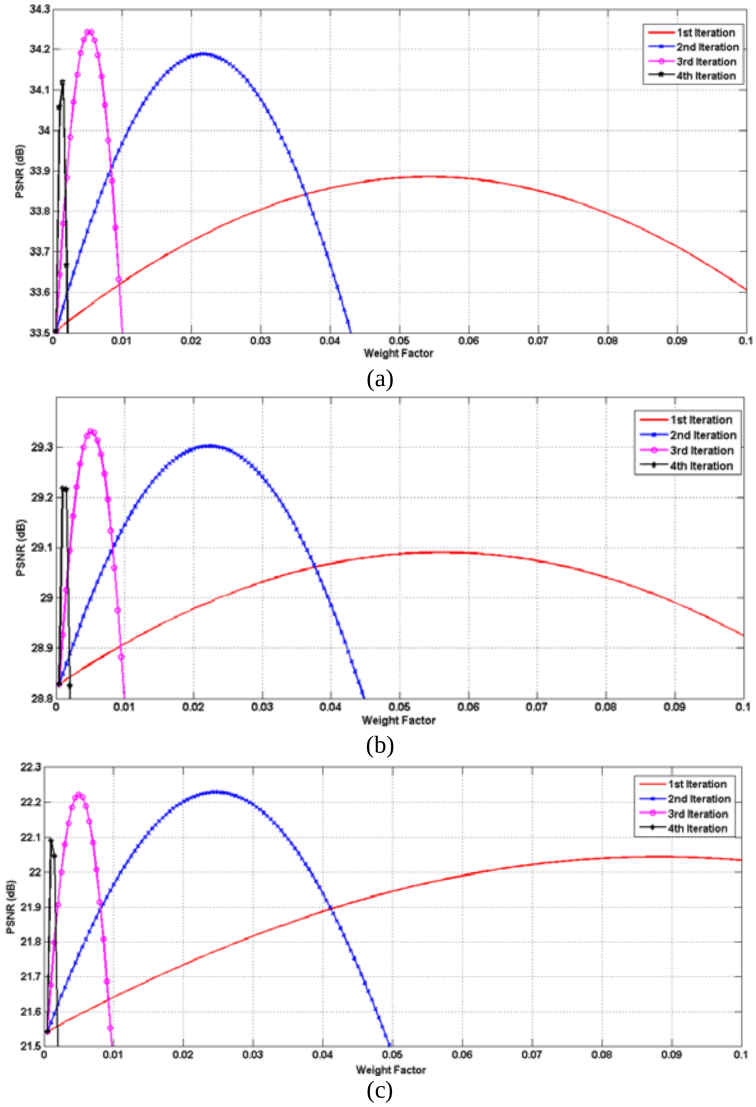


Fig. 6.2 PSNR (dB) vs. K_1 characteristic plot for 1st frame of different sequences for different iterations at 4:1 compression ratio: (a) Akiyo; (b) Soccer; (c) Mobile

6.1.3 Estimation of Weight Factors K_1 and K_2 for CS-I

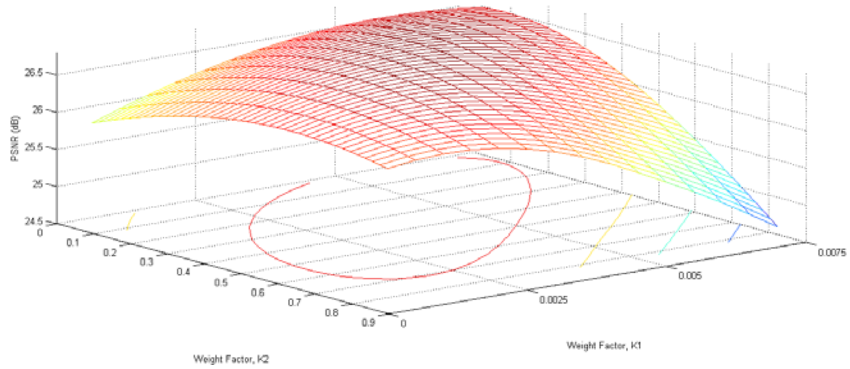
In case of the composite scheme, the PSNR (dB) gain is a function of weight factors K_1 and K_2. Therefore, the precise estimation of weight factors is imperative for optimum performance of the composite algorithm. In order to obtain the optimized weight factors, simulation studies are carried to observe the variation in PSNR (dB) with respect to the variations in the weight factors K_1 and K_2, respectively. Hence, PSNR (dB) vs. K_1 and K_2 surface plots are generated for three different types of images as illustrated in Fig. 6.3. The weight factors K_1 and K_2 corresponding to maximum PSNR are determined from the 3-D characteristic plots for each image type. The contours corresponding to high PSNR region in the 3-D plots don't vary much in all the three different cases. Hence, the optimized weight factors K_1 and K_2 are calculated by averaging the corresponding weight factors in all the three different cases as illustrated in Table 6.1. In case of the composite scheme, the weight factors are found to be less i.e. almost half the value of the standalone schemes. It is because, the HF enhancement is performed in two phases unlike the standalone schemes where the HF enhancement is done in a single phase. In addition, the weight factors corresponding to maximum PSNR doesn't vary much for different images because of the improved nonlinearity and adaptability of the composite algorithm.

The estimated weight factors K_1 and K_2 are different for different images but their variations for the optimum performance is confined to a narrow range because of adopting higher order derivative operator in the pre-processing operation and by opting a highly adaptive, local statistics based Laplacian filter for the post-processing operation. Hence, their estimation become more precise for better HF restoration performance.

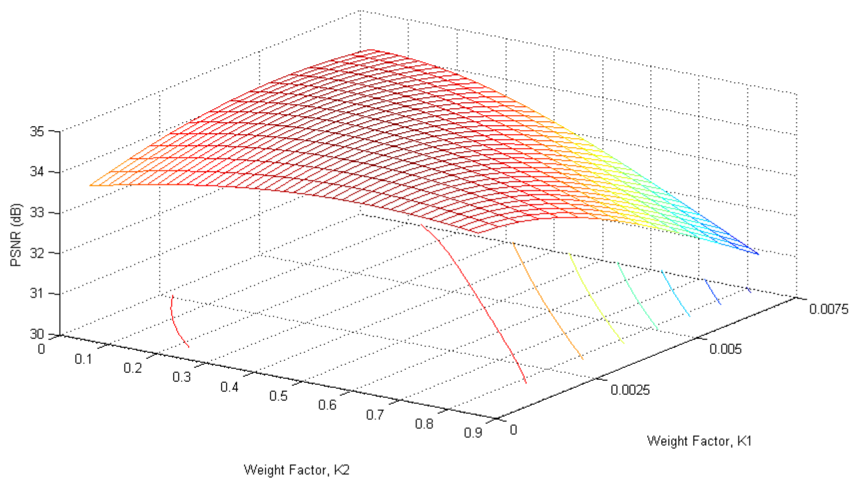
Moreover, the weight factors are different in the standalone pre-processing and post-processing schemes. The standalone pre-processing weight factor, K_1 is determined from the PSNR vs. K_1 characteristics plots of 1st frame of different sequences which are given in Fig. 6.2. The weight factor, K_1 corresponding to maximum PSNR is taken for three different frames. The overall weight factor is estimated by averaging these weight factors and is taken as 0.005. Likewise, the estimation standalone post-processing weight factor, K_2 is given in Section 5.2.3.

Table 6.1 Weight factors, K_1 and K_2 estimation for the proposed composite scheme, CS-I

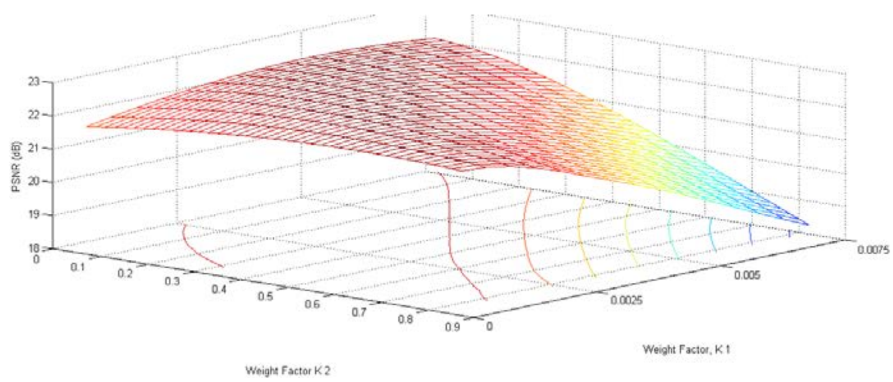
	1st frame of different Sequences			
Weight Factors	**Akiyo**	**Bus**	**Mobile**	**Average Weight Factors**
K_1	0.0025	0.0035	0.0025	0.0028
K_2	0.32	0.3	0.24	0.2866



(a)



(b)



(c)

Fig. 6.3 3-D characteristic plot of PSNR (dB) vs. K_1 and K_2 for the 1st frame of different sequences using the proposed composite scheme CS-I: (a) Bus; (b) Akiyo; (c) Mobile

6.2 Composite Scheme (CS-II) using Iterative Laplacian of Laplacian and Fuzzy Weighted Laplacian [P11]

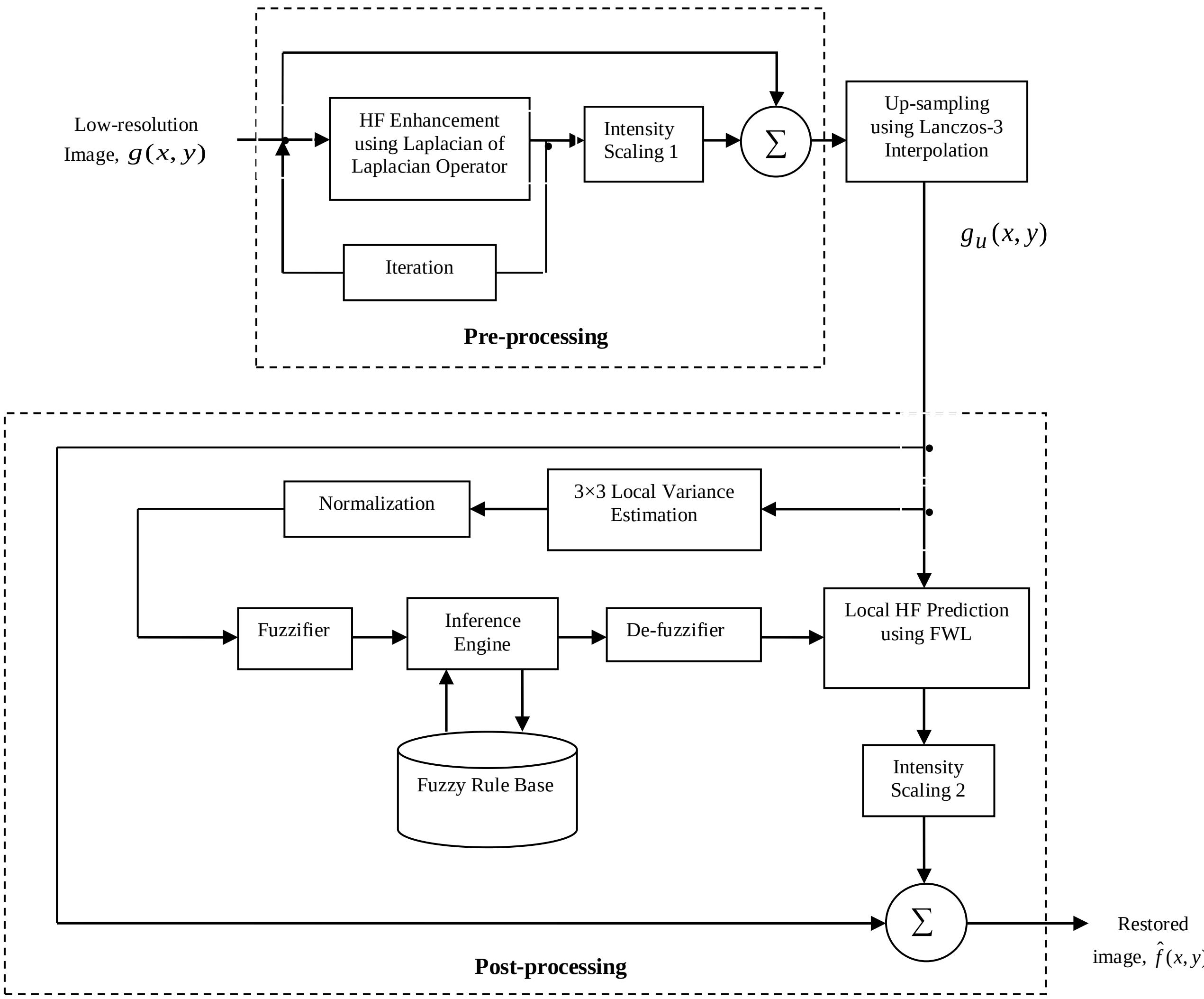


Fig. 6.4 Composite Scheme (CS-II) using Iterative Laplacian of Laplacian and Fuzzy weighted Laplacian

There is a scope to improve the performance of the previous composite algorithm, CS-I through incorporation of fuzzy rule base. Here, a highly nonlinear, fuzzy logic based, composite scheme, CS-II is proposed by combining a pre-processing and a post-processing operation to efficiently restore high frequency (HF) and very high frequency (VHF) details in an up-scaled image. The fuzzy composite scheme is developed on the basis of inverse process of HF degradation to resolve the blurring problem effectively.

During the pre-processing operation, the HF and VHF components of an image are boosted up using recursive Laplacian of Laplacian (LOL) operator prior to image up-scaling. Subsequent to the image up-scaling, a local adaptive, composite fuzzy weighted Laplacian post-processing scheme is used for further quality improvement of the up-scaled image. The post-processing operation is a local statistics based fuzzy weighted Laplacian scheme which enhances the high variance regions more than the low variance regions based on fuzzy rule base so as to effectively restore the HF and VHF components.

The HF restoration performance of the fuzzy based composite scheme is enhanced by improving its nonlinearity through the variations of different parameters of the fuzzy inference system (FIS) such as slope, width and the number of input-, and output membership functions. The effective fusion of pre-processing and post-processing operations makes the proposed scheme much effective to tackle the non-uniform blurring than the standalone pre-processing and post-processing techniques.

The proposed composite scheme, CS-II comprises the following three steps.

1. HF enhancement using iterative Laplacian of Laplacian global pre-processing (ILLGP)
2. Image up-scaling using Lanczos-3 interpolation
3. HF prediction using composite fuzzy weighted Laplacian (CFWL) post-processing

The pre-processing scheme (ILLGP) is discussed in Section 3.3 of Chapter-3. The CFWL scheme is same as FWL post-processing scheme discussed in Section 5.2 of Chapter-5. However, the fuzzy mapping technique, employed in CFWL is different from the fuzzy mapping technique used in standalone FWL post-processing scheme discussed in Section 5.2. The remaining part of both of the algorithms is same. A different fuzzy mapping technique is employed in CFWL scheme to further raise the nonlinearity of FWL for better restoration performance. The fuzzy mapping technique employed in CFWL scheme is discussed in the subsequent section.

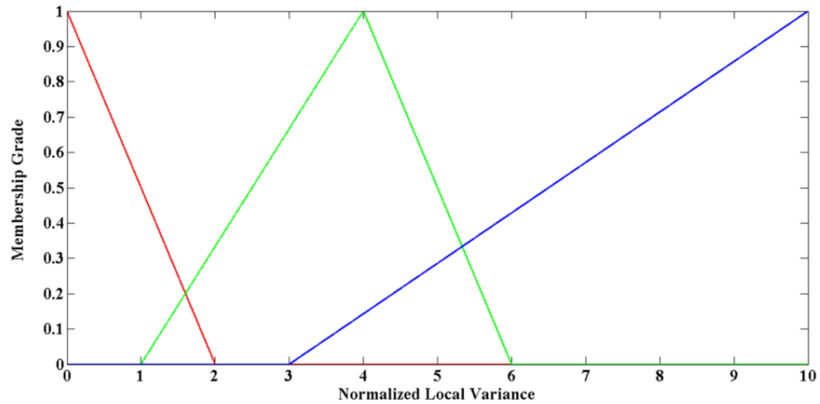
6.2.1 Fuzzy Mapping Technique Employed in CFWL

In case of CFWL post-processing scheme, the algorithm is optimized by making its mapping more nonlinear. The nonlinearity of the algorithm is enhanced by varying the parameters such as slope, width of each membership function and by keeping the number of input and output membership functions uneven. The CFWL post-processing scheme is a single-input-single-output (SISO) fuzzy inference system (FIS) that improves the nonlinearity of the process for better restoration performance.

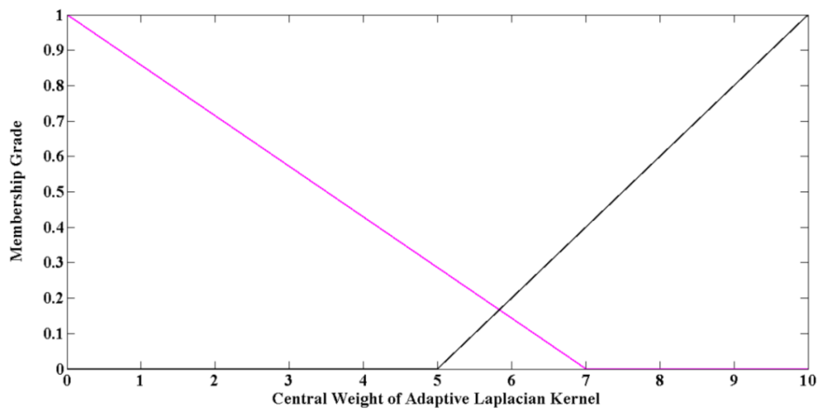
Selection of the number and type of membership functions plays a major role in deciding the degree of nonlinearity of a fuzzy system. As per our convenience, triangular membership functions are taken for fuzzyfication of input-, and output-variables. In this algorithm, the numbers of input and output membership functions are kept uneven i.e. three and two respectively to raise the nonlinearity of FIS for better restoration performance. Moreover, the slope and width of each input and output triangular membership functions are kept different in order to make the system more nonlinear. The input and output variables are normalized to a scale of 10.

In this CFWL algorithm, 3×3 normalized local variance, σ_N^2 is taken as the input variable and the central weight, w_{tO} of the fuzzy weighted Laplacian kernel is taken as the output variable of FIS. The central weight, w_{tO} is updated as per the de-fuzzyfied output of the FIS which maps the output according to the normalized 3×3 local variance based on the rule base. Centroid method of de-fuzzification is used to produce a crisp defuzzified output which updates the coefficients of fuzzy weighted Laplacian kernel for enhancement of local HF information. The graphical representation of input and output membership functions is depicted in Fig. 6.5. A sum total of five fuzzy if-then rules are formed for inclusion into the rule base.

Let $\mu_{iX}(\sigma_N^2), \mu_{iY}(\sigma_N^2)$ and $\mu_{iZ}(\sigma_N^2)$ denote the low, medium and high input membership functions of the input-variable, σ_N^2. Let $\mu_{OX}(w_t)$ and $\mu_{OY}(w_t)$ be the low and high membership functions of the output variable respectively. The expressions for the three input membership functions are given by,



(a)



(b)

Fig. 6.5 Graphical representation of membership functions of CFWL post-processing algorithm: (a) Input; (b) output

$$\mu_{iX}(\sigma_N^2; 0, 0, 2) = \begin{cases} 0, & \sigma_N^2 \le 0 \\ \dfrac{2-\sigma_N^2}{2}, & 0 \le \sigma_N^2 \le 2 \\ 0, & 2 \le \sigma_N^2 \end{cases} \tag{6.6a}$$

$$\mu_{iY}(\sigma_N^2; 1, 4, 6) = \begin{cases} 0, & \sigma_N^2 \le 1 \\ \dfrac{\sigma_N^2 - 1}{4-1}, & 1 \le \sigma_N^2 \le 4 \\ \dfrac{6-\sigma_N^2}{6-4}, & 4 \le \sigma_N^2 \le 6 \\ 0, & 6 \le \sigma_N^2 \end{cases} \tag{6.6b}$$

$$\mu_{iZ}(\sigma_N^2;3,10,10)=\begin{cases}0, & \sigma_N^2\le 3\\ \dfrac{\sigma_N^2-3}{10-3}, & 3\le\sigma_N^2\le 10\\ 0, & 10\le\sigma_N^2\end{cases}\tag{6.6c}$$

Similarly, the expressions for output membership functions are given by,

$$\mu_{OX}(w_t;0,0,7)=\begin{cases}0, & w_t\le 0\\ \dfrac{7-w_t}{10-3}, & 0\le w_t\le 7\\ 0, & 7\le w_t\end{cases}\tag{6.7a}$$

$$\mu_{OY}(w_t;5,10,10)=\begin{cases}0, & w_t\le 5\\ \dfrac{w_t-5}{10-5}, & 5\le w_t\le 10\\ 0, & 10\le w_t\end{cases}\tag{6.7b}$$

As per the rule base, the minimum of any two membership functions are given by,

$$\mu_{iX\cap OX}(\sigma_N^2,w_t)=\mathbf{min}\left(\left\{\mu_{iX}(\sigma_N^2),\mu_{OX}(w_t)\right\}\right)\tag{6.8a}$$

$$\mu_{iY\cap OX}(\sigma_N^2,w_t)=\mathbf{min}\left(\left\{\mu_{iY}(\sigma_N^2),\mu_{OX}(w_t)\right\}\right)\tag{6.8b}$$

$$\mu_{iY\cap OY}(\sigma_N^2,w_t)=\mathbf{min}\left(\left\{\mu_{iY}(\sigma_N^2),\mu_{OY}(w_t)\right\}\right)\tag{6.8c}$$

$$\mu_{iZ\cap OX}(\sigma_N^2,w_t)=\mathbf{min}\left(\left\{\mu_{iZ}(\sigma_N^2),\mu_{OX}(w_t)\right\}\right)\tag{6.8d}$$

$$\mu_{iZ\cap OY}(\sigma_N^2,w_t)=\mathbf{min}\left(\left\{\mu_{iZ}(\sigma_N^2),\mu_{OY}(w_t)\right\}\right)\tag{6.8e}$$

To obtain an individual response, $R_1(w_t)$ corresponding to an arbitrary input σ_N^2, AND operation is performed in between $\mu_{iX}(\sigma_N^2)$ and the general result $\mu_{iX\cap OX}(\sigma_N^2,w_t)$ evaluated at σ_N^2 according to fuzzy if-then rule-1. Similarly, $R_2(w_t)$, $R_3(w_t)$, $R_4(w_t)$ and $R_5(w_t)$ are obtained as per the knowledge base. The expressions for all the individual responses according to the rule base are given by,

$$R_1(w_t)=\mathbf{min}\left(\left\{\mu_{iX}(\sigma_N^2),\mu_{iX\cap OX}(\sigma_N^2,w_t)\right\}\right)\tag{6.9a}$$

$$R_2(w_t)=\mathbf{min}\left(\left\{\mu_{iY}(\sigma_N^2),\mu_{iY\cap OX}(\sigma_N^2,w_t)\right\}\right)\tag{6.9b}$$

$$R_3(w_t)=\mathbf{min}\left(\left\{\mu_{iY}(\sigma_N^2),\mu_{iY\cap OY}(\sigma_N^2,w_t)\right\}\right)\tag{6.9c}$$

$$R_4(w_t) = \mathbf{min}\left(\left\{ \mu_{iZ}(\sigma_N^2), \mu_{iZ \cap OX}(\sigma_N^2, w_t) \right\} \right) \tag{6.9d}$$

$$R_5(w_t) = \mathbf{min}\left(\left\{ \mu_{iZ}(\sigma_N^2), \mu_{iZ \cap OY}(\sigma_N^2, w_t) \right\} \right) \tag{6.9e}$$

The overall response is obtained by aggregating all the individual responses using OR operator and is given by,

$$\begin{aligned} R(w_t) &= \max\{R_j(w_t)\} \\ &\quad where,\ j \in \{1, 2, \cdots, 5\} \end{aligned} \tag{6.10}$$

The crisp output, w_{tO} from the fuzzy set, $R(w_t)$ is obtained through centre of gravity de-fuzzification method. Since $R(w_t)$ can have k possible values, the centre of gravity of $R(w_t)$ is given by,

$$w_{tO} = \frac{\sum_{w_t=1}^{k} w_t \ R(w_t)}{\sum_{w_t=1}^{k} R(w_t)} \tag{6.11}$$

The de-fuzzified output, w_{tO} updates the central weight of the fuzzy weighted Laplacian kernel $h_{FWL}(x, y)$ for enhancing local high frequency information and is given by,

$$h_{FWL}(x, y) = \begin{bmatrix} 0 & -\frac{w_{tO}}{4} & 0 \\ -\frac{w_{tO}}{4} & w_{tO} & -\frac{w_{tO}}{4} \\ 0 & -\frac{w_{tO}}{4} & 0 \end{bmatrix} \tag{6.12}$$

The remaining filter coefficients of the FWL kernel are attuned so that the sum of all the filter coefficients becomes zero. The expression of central weight, w_{tO} shows nonlinear mapping between input and output variables.

The overall input-output curve of FIS is obtained by plotting the de-fuzzified output, w_{tO} with respect to the normalized local variance, σ_N^2. The plot is obtained using the 1st frame of salesman sequence and is shown in Fig. 6.6. The overall input-output characteristic gives an idea about the characteristics of composite fuzzy mapping technique. It is quite apparent from the figure that the characteristic is nonlinear and hence is responsible for nonlinear mapping between input and output variables for better HF restoration performance.

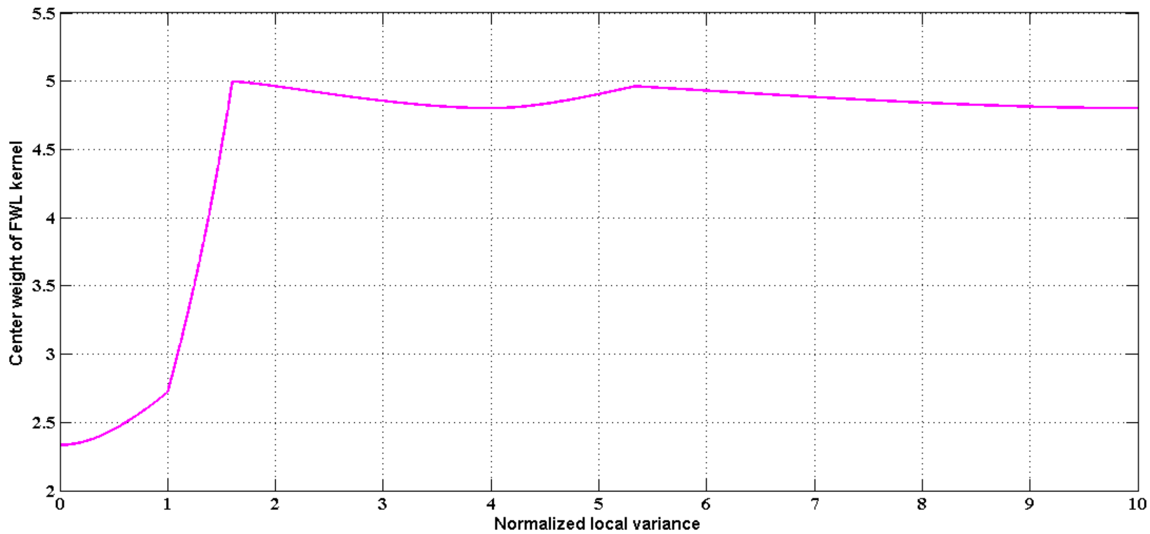


Fig. 6.6 Overall input output curve of 1st frame of salesman sequence using CFWL algorithm

HF extracts from the up-scaled image is obtained using local adaptive FWL filter. Let $g_u(x,y)$ and $\psi(x,y)$ be the up-scaled image and the HF extracts respectively. The HF extract obtained by FWL filter is given by,

$$\psi(x,y) = \sum_{s=-1}^{1}\sum_{t=-1}^{1} h_{FWL}(s,t)\, g_u(x+s, y+t) \tag{6.13}$$

Intensity scaling is performed on the HF extracts according to the estimated weight factor, K_2 so as to perform precise sharpening to counter blurring effectively in an up-sampled image. Precise estimation of weight factor plays a major role for the improved performance of the proposed composite algorithm. The weight factor, K_2 at 4:1 compression ratio is different for composite and standalone scheme and is illustrated in Section 6.2.3. The weighted version of the HF extracts is then superimposed on the up-sampled image to obtain the restored image and is given by,

$$\hat{f}(x,y) = g_u(x,y) + K_2\, \psi(x,y) \tag{6.14}$$

where, $K_2 = 0.186$. The detail estimation of K_2 is illustrated in Section 6.2.3.

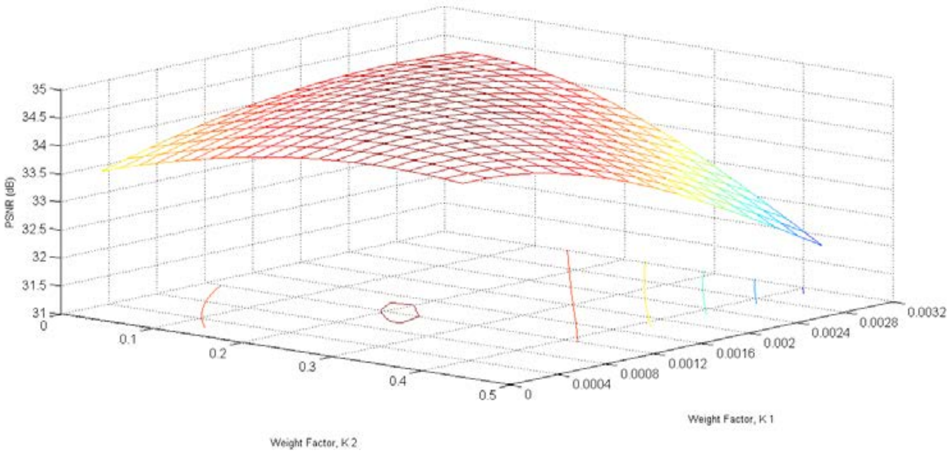
6.2.2 CS-II Algorithm

The proposed CS-II composite algorithm comprises a pre-processing and a post-processing phase. The pre-processing phase completely utilizes ILLGP algorithm to enhance the HF details in an input image. However, the weight factor, K_1 used in this pre-processing phase of the composite scheme is different from the weight factor used in standalone ILLGP algorithm. The weight factor used in the pre-processing phase is estimated as $K_1 = 0.0012$. The ILLGP algorithm is illustrated in Section 3.3.1. The estimation of weight factor, K_1 is given in Section 6.2.3.

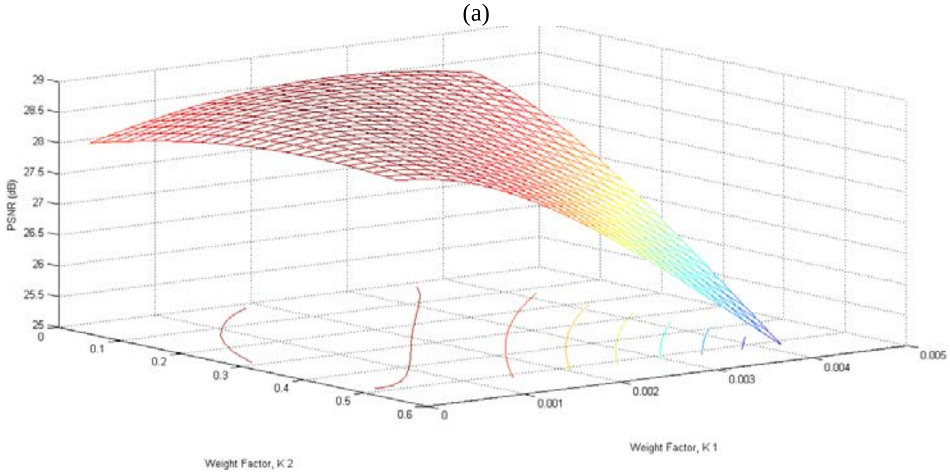
The post-processing phase utilizes FWL algorithm to enhance and predict HF information in the up-scaled image. The FWL post-processing algorithm is illustrated in Section 5.2.2. But the fuzzy mapping technique employed in this composite scheme is different from the fuzzy mapping technique employed in FWL standalone post-processing scheme. The fuzzy mapping technique employed in CS-II algorithm is illustrated in Section 6.2.1. The weight factor, K_2 used in the post-processing phase of the composite algorithm is different from the standalone FWL algorithm. The weight factor, corresponding to post-processing phase of the composite algorithm is given by, $K_2 = 0.186$. The detail estimation of weight factor, K_2 is given in Section 6.2.3.

6.2.3 Estimation of Weight Factors K_1 and K_2 for CS-II

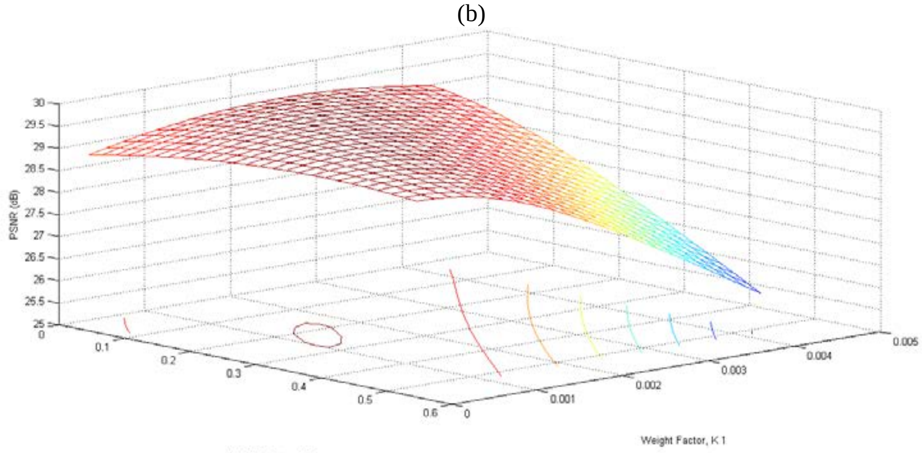
The proposed CS-II composite algorithm comprises a pre-processing and post-processing phase and hence has two weight factors to perform intensity scaling in both of the phases. In case of the composite scheme, the PSNR (dB) gain is a function of weight factors K_1 and K_2. Therefore, the precise estimation of weight factors is imperative for optimum performance of the composite algorithm. In order to obtain the optimized weight factors, simulation studies are carried to observe the variation in PSNR (dB) with respect to the variations in the weight factors K_1 and K_2 respectively. Hence, PSNR (dB) vs. K_1 and K_2 surface plots are generated for three different types of images as illustrated in Fig. 6.7. The weight factors K_1 and K_2 corresponding to maximum PSNR are determined from the 3-D characteristic plots for each image. The contours corresponding to high PSNR region in the 3-D plots don't vary much in all the three



(a)



(b)



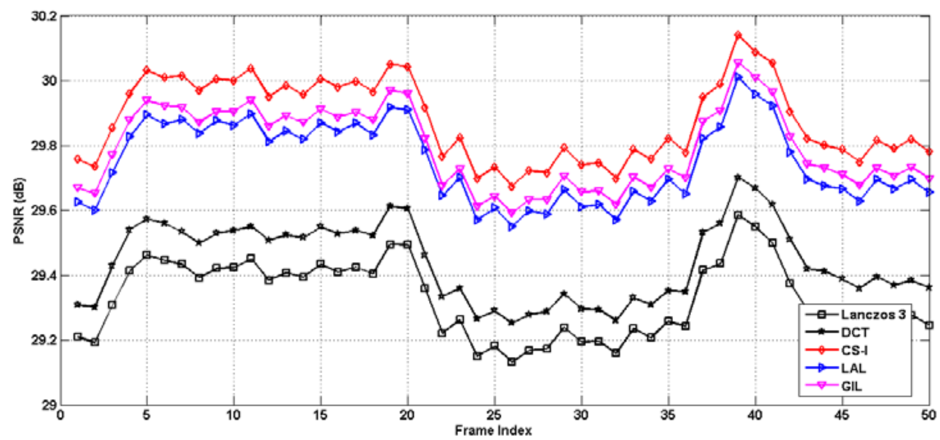
(c)

Fig. 6.7 3-D characteristic plot of PSNR (dB) vs. K_1 and K_2 for the 1st frame of different sequences using the proposed composite scheme, CS-II: (a) Akiyo; (b) City; (c) Soccer

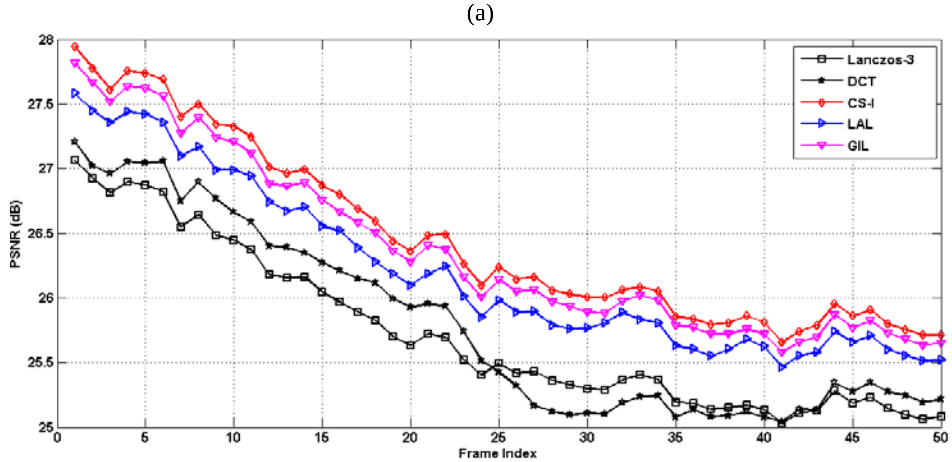
Table 6.2 Weight factor, K_1 and K_2 estimation for the proposed composite scheme, CS-II

Weight Factors	1st frame of different Sequences			Average Weight Factors
	Akiyo	City	Soccer	
K_1	0.001	0.0014	0.0012	0.0012
K_2	0.2	0.16	0.2	0.186

different cases. Hence, the optimized weight factors K_1 and K_2 are calculated by averaging the corresponding weight factors in all the three different cases as illustrated in Table 6.2. K_1 is estimated to be 0.0012 and the weight factor K_2 is estimated as 0.186 for the composite scheme, CS-II for better HF restoration performance.



(a)



(b)

Fig. 6.8 PSNR (dB) comparison of composite scheme, CS-I with pre-processing scheme, GIL and post-processing scheme, LAL for different video sequences at 4:1 compression ratio: (a) Salesman; (b) Bus

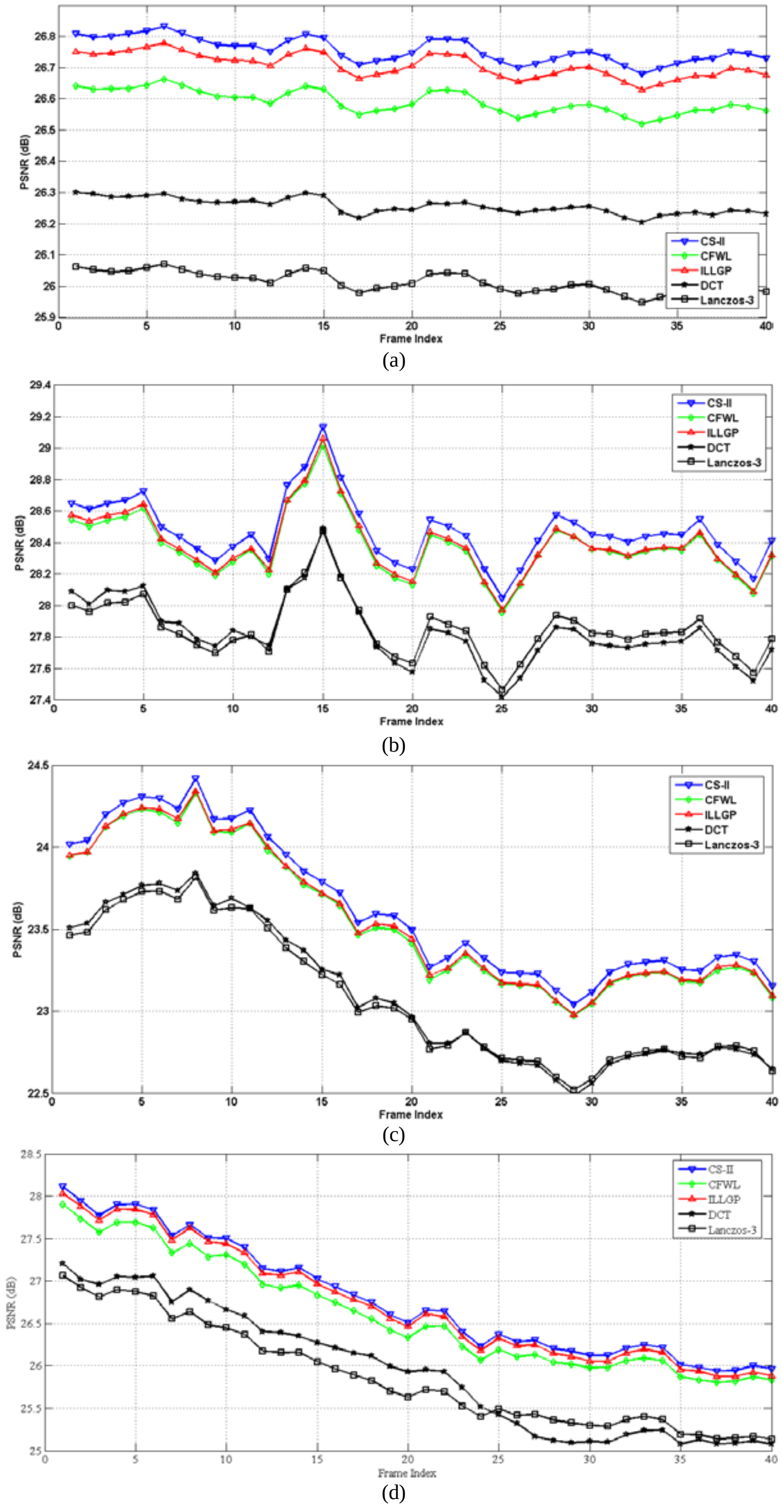


Fig. 6.9 PSNR (dB) comparison of composite scheme, CS-II with pre-processing scheme, ILLGP and fuzzy post-processing scheme, CFWL for different video sequences at 4:1 compression ratio: (a) Container; (b) City; (c) Stefan; (d) Bus

Table 6.3 PSNR (dB) and UQI comparison of different schemes at 4:1 compression ratio for various (512×512) images

Image	**Image Metric**	**Bicubic [4]**	**Lanczos3 [12]**	**DCT [74]**	**LAL [P7]**	**FWL [P8]**	**ILLGP [P3]**	**GIL**	**CS-I [P10]**	**CS-II [P11]**
Mandril	PSNR(dB)	23.630	23.859	23.925	24.066	24.178	**24.288**	24.218	24.232	24.282
	UQI	0.9114	0.9170	0.9187	0.9228	0.9260	0.9271	0.9263	0.9263	**0.9278**
Lena	PSNR(dB)	34.148	34.813	35.023	35.080	35.250	35.439	35.444	35.531	**35.562**
	UQI	0.9945	0.9953	0.9955	0.9956	0.9958	0.9959	0.9959	0.9960	**0.9961**
Barbara	PSNR(dB)	25.352	25.428	25.183	25.562	25.692	25.744	25.706	25.748	**25.781**
	UQI	0.9669	0.9675	0.9657	0.9687	0.9698	0.9701	0.9699	0.9701	**0.9705**
Boat	PSNR(dB)	29.952	30.375	30.466	30.841	30.870	30.879	30.825	30.958	**30.983**
	UQI	0.9845	0.9860	0.9863	0.9876	0.9877	0.9876	0.9875	0.9879	**0.9880**
Goldhill	PSNR(dB)	31.405	31.725	31.716	31.993	32.092	32.080	32.087	32.157	**32.197**
	UQI	0.9901	0.9909	0.9909	0.9915	0.9917	0.9916	0.9916	0.9918	**0.9919**
Pirate	PSNR(dB)	31.058	31.490	31.606	31.952	31.986	32.047	31.997	32.140	**32.161**
	UQI	0.9885	0.9897	0.9899	0.9908	0.9909	0.9910	0.9909	0.9912	**0.9912**
Livingroom	PSNR(dB)	29.557	29.977	30.128	30.311	30.439	30.524	30.446	30.480	**30.594**
	UQI	0.9811	0.9829	0.9835	0.9843	0.9850	0.9851	0.9848	0.9849	**0.9854**
Fingerprint	PSNR(dB)	30.632	31.722	32.133	31.138	31.345	**32.798**	32.668	32.457	32.458
	UQI	0.9889	0.9915	0.9922	0.9908	0.9913	**0.9934**	0.9933	0.9930	0.9931
Baboon	PSNR(dB)	35.014	35.662	35.890	35.850	36.042	36.142	36.214	36.240	**36.271**
	UQI	0.9946	0.9954	0.9956	0.9956	0.9958	0.9959	0.9959	**0.9960**	**0.9960**
Bridge	PSNR(dB)	26.504	26.826	26.919	27.160	27.255	27.333	27.245	27.343	**27.367**
	UQI	0.9748	0.9767	0.9772	0.9787	0.9793	0.9795	0.9791	0.9796	**0.9798**
Cameraman	PSNR(dB)	35.757	37.216	37.832	36.963	37.381	38.349	38.385	38.243	**38.463**
	UQI	0.9977	0.9984	0.9986	0.9983	0.9985	**0.9988**	0.9988	0.9987	**0.9988**
Cat	PSNR(dB)	31.982	32.427	32.563	32.584	32.826	**33.046**	32.937	32.941	33.037
	UQI	0.9937	0.9943	0.9945	0.9946	0.9949	**0.9951**	0.9950	0.9950	**0.9951**
Crowd	PSNR(dB)	32.666	33.451	33.768	34.035	34.026	34.415	34.246	34.405	**34.430**
	UQI	0.9930	0.9942	0.9946	0.9949	0.9950	0.9953	0.9952	0.9954	**0.9954**
Cycle	PSNR(dB)	21.895	22.154	22.129	22.452	22.543	22.633	22.570	22.597	**22.661**
	UQI	0.9437	0.9475	0.9474	0.9526	0.9538	0.9541	0.9536	0.9541	**0.9551**
F16	PSNR(dB)	31.543	32.104	**32.722**	32.353	32.319	32.484	32.536	32.638	32.576
	UQI	0.9885	0.9900	**0.9913**	0.9907	0.9907	0.9909	0.9910	0.9912	0.9912
House	PSNR(dB)	30.314	30.807	30.862	30.980	31.304	**31.596**	31.423	31.412	31.585
	UQI	0.9785	0.9809	0.9812	0.9820	0.9835	0.9843	0.9837	0.9837	**0.9844**
Lake	PSNR(dB)	30.022	30.495	30.793	30.617	30.700	**31.021**	30.942	30.960	30.965
	UQI	0.9919	0.9928	0.9933	0.9930	0.9932	**0.9936**	0.9935	0.9936	**0.9936**
Peppers	PSNR(dB)	31.991	32.328	**32.747**	32.509	32.407	32.652	32.644	32.713	32.625
	UQI	0.9937	0.9942	**0.9947**	0.9944	0.9943	0.9946	0.9946	0.9947	0.9946
Elaine	PSNR(dB)	33.117	33.309	33.284	33.315	33.413	33.371	33.479	**33.523**	33.502
	UQI	0.9924	0.9928	0.9927	0.9928	0.9930	0.9929	0.9931	**0.9932**	0.9931
Ruler	PSNR(dB)	12.613	12.673	12.600	12.584	12.823	**12.890**	12.823	12.749	12.765
	UQI	0.5735	0.5898	0.5920	0.6088	0.6371	**0.6351**	0.6211	0.6165	0.6337

Table 6.4 PSNR (dB) and UQI comparison of different schemes at 16:1 compression ratio for various (512×512) images

Image	Image Metric	Bicubic [4]	Lanczos3 [12]	DCT [74]	LAL [P7]	FWL [P8]	ILLGP [P3]	CS-I [P10]	CS-II [P11]
Mandril	PSNR(dB)	21.085	21.156	21.167	21.181	21.210	**21.292**	21.280	21.286
	UQI	0.8309	0.8351	0.8362	0.8365	0.8383	**0.8440**	0.8423	0.8424
Lena	PSNR(dB)	28.848	29.183	29.296	29.389	29.407	29.556	**29.635**	29.612
	UQI	0.9810	0.9825	0.9829	0.9833	0.9835	0.9841	**0.9843**	**0.9843**
Barbara	PSNR(dB)	23.607	23.708	23.747	23.755	23.765	23.829	**23.843**	23.842
	UQI	0.9496	0.9510	0.9515	0.9516	0.9519	0.9528	**0.9528**	**0.9528**
Boat	PSNR(dB)	25.538	25.739	25.773	25.849	25.880	**26.024**	26.000	26.020
	UQI	0.9557	0.9580	0.9584	0.9592	0.9596	**0.9612**	0.9608	0.9610
Goldhill	PSNR(dB)	27.628	27.798	27.776	27.878	27.913	27.973	28.009	**28.020**
	UQI	0.9761	0.9771	0.9770	0.9775	0.9778	0.9782	0.9783	**0.9784**
Pirate	PSNR(dB)	26.861	27.083	27.127	27.228	27.243	27.379	27.395	**27.397**
	UQI	0.9691	0.9708	0.9712	0.9719	0.9721	0.9731	0.9731	**0.9731**
Livingroom	PSNR(dB)	25.385	25.560	25.563	25.634	25.674	**25.815**	25.781	25.812
	UQI	0.9488	0.9511	0.9513	0.9521	0.9527	**0.9547**	0.9540	0.9544
Fingerprint	PSNR(dB)	22.633	23.779	24.202	24.218	24.329	**24.948**	24.773	24.875
	UQI	0.9192	0.9407	0.9467	0.9478	0.9496	**0.9589**	0.9567	0.9570
Baboon	PSNR(dB)	29.604	29.931	30.007	30.018	30.118	30.318	30.341	**30.353**
	UQI	0.9808	0.9823	0.9826	0.9827	0.9831	0.9840	0.9840	**0.9841**
Bridge	PSNR(dB)	23.066	23.230	23.263	23.297	23.331	**23.473**	23.433	23.453
	UQI	0.9423	0.9449	0.9454	0.9459	0.9465	**0.9487**	0.9479	0.9482
Cameraman	PSNR(dB)	27.546	27.946	28.094	28.264	28.261	28.532	28.513	**28.540**
	UQI	0.9847	0.9861	0.9866	0.9871	0.9872	0.9880	0.9879	**0.9880**
Cat	PSNR(dB)	27.838	28.059	28.105	28.141	28.201	**28.417**	28.371	28.390
	UQI	0.9835	0.9844	0.9845	0.9847	0.9849	**0.9857**	0.9855	0.9856
Crowd	PSNR(dB)	26.601	26.974	27.066	27.205	27.217	**27.497**	27.451	27.470
	UQI	0.9706	0.9733	0.9739	0.9748	0.9750	**0.9767**	0.9764	0.9764
Cycle	PSNR(dB)	18.996	19.109	19.135	19.167	19.198	**19.302**	19.286	19.296
	UQI	0.8842	0.8883	0.8893	0.8903	0.8915	**0.8959**	0.8945	0.8948
F16	PSNR(dB)	26.457	26.773	26.939	26.965	26.963	27.055	**27.124**	27.116
	UQI	0.9617	0.9647	0.9661	0.9664	0.9665	0.9675	**0.9678**	0.9678
House	PSNR(dB)	25.863	26.050	26.063	26.155	26.183	**26.322**	26.268	26.291
	UQI	0.9373	0.9406	0.9409	0.9423	0.9429	**0.9452**	0.9442	0.9445
Lake	PSNR(dB)	25.466	25.746	25.834	25.911	25.924	**26.139**	26.105	26.124
	UQI	0.9764	0.9780	0.9785	0.9789	0.9790	**0.9801**	0.9799	0.9800
Peppers	PSNR(dB)	28.045	28.330	28.524	28.547	28.496	28.578	28.630	**28.617**
	UQI	0.9841	0.9852	0.9859	0.9860	0.9858	0.9861	0.9863	**0.9862**
Elaine	PSNR(dB)	30.496	30.785	30.884	30.984	30.959	30.902	**31.111**	31.058
	UQI	0.9860	0.9870	0.9878	0.9876	0.9857	0.9874	**0.9880**	0.9878
Ruler	PSNR(dB)	10.804	10.873	10.876	10.873	10.884	**10.902**	10.890	10.898
	UQI	0.2394	0.2608	0.2636	0.2638	0.2691	**0.2788**	0.2705	0.2740

Table 6.5 Average PSNR (dB) and UQI comparison of different interpolation techniques at 4:1 compression ratio for various sequences over 50 frames

Sequence	**Image Metric**	**Bicubic [4]**	**Lanczos3 [12]**	**DCT [74]**	**LAL [P7]**	**FWL [P8]**	**ILLGP [P3]**	**CS-I [P10]**	**CS-II [P11]**
Salesman	PSNR(dB)	28.979	29.327	29.440	29.744	29.804	**29.934**	29.871	30.003
	UQI	0.9719	0.9742	0.9750	0.9770	0.9776	**0.9779**	0.9777	0.9287
Bus	PSNR(dB)	25.262	25.716	25.794	26.177	26.362	26.557	26.579	**26.820**
	UQI	0.9610	0.9652	0.9660	0.9698	0.9713	0.9720	0.9716	**0.9733**
Akiyo	PSNR(dB)	32.911	33.450	33.647	34.187	34.110	34.206	**34.413**	34.402
	UQI	0.9945	0.9952	0.9954	0.9959	0.9959	0.9959	**0.9961**	**0.9961**
City	PSNR(dB)	27.592	27.879	27.852	28.248	28.358	28.419	28.402	**28.479**
	UQI	0.9277	0.9333	0.9333	0.9407	0.9433	0.9431	0.9426	**0.9449**
Container	PSNR(dB)	25.567	26.008	26.256	26.376	26.448	26.705	26.622	**26.768**
	UQI	0.9676	0.9709	0.9727	0.9739	0.9746	0.9757	0.9752	**0.9763**
Football	PSNR(dB)	28.568	29.366	29.687	29.983	30.178	**30.604**	29.950	30.123
	UQI	0.9769	0.9810	0.9824	0.9840	0.9849	**0.9860**	0.9843	0.9849
Mobile	PSNR(dB)	21.197	21.596	21.758	22.199	22.239	22.427	22.397	**22.484**
	UQI	0.9502	0.9550	0.9569	0.9622	0.9628	0.9639	0.9636	**0.9646**
Soccer	PSNR(dB)	30.254	30.664	30.748	31.020	31.166	31.217	**31.330**	31.265
	UQI	0.9863	0.9876	0.9878	0.9887	0.9891	0.9892	0.9895	**0.9900**
Coast	PSNR(dB)	26.500	26.939	27.080	27.255	27.575	27.733	27.621	**27.952**
	UQI	0.9754	0.9779	0.9787	0.9797	0.9814	0.9819	0.9813	**0.9828**

Table 6.6 Execution time of the proposed and existing algorithms at 4:1 CR

Images M×N	**Execution time in Seconds**								
	Bicubic [4]	**Lanczos3 [12]**	**DCT [74]**	**LAL [P7]**	**FWL [P8]**	**ILLGP [P3]**	**GIL**	**CS-I [P10]**	**CS-II [P11]**
Clock (200×200)	0.0150	0.0151	0.0518	3.1982	8.6076	0.1300	0.0907	3.4249	11.7605
Lena (256×256)	0.0152	0.0157	0.0633	7.7949	21.5439	0.1328	0.0925	9.8969	27.5542
Fruit (377×321)	0.0163	0.0168	0.1629	56.9265	82.9838	0.1295	0.0986	61.5708	97.2611
Lena (512×512)	0.0215	0.0222	0.1773	335.1430	384.0769	0.1420	0.1041	343.3647	415.4207
Pentagon (1024×1024)	0.0409	0.0442	0.6372	6140.0514	6399.0850	0.1767	0.1260	6158.1726	6462.9129

Fig. 6.10 Subjective evaluation of Lena (512×512) image using various up-sampling schemes at 4:1 compression ratio: (a) Original; (b) Bicubic; (c) Lanczos-3; (d) DCT; (e) LAL (P7); (f) FWL (P8); (g) ILLGP (P3); (h) CS-I (P10); (i) CS-II (P11)

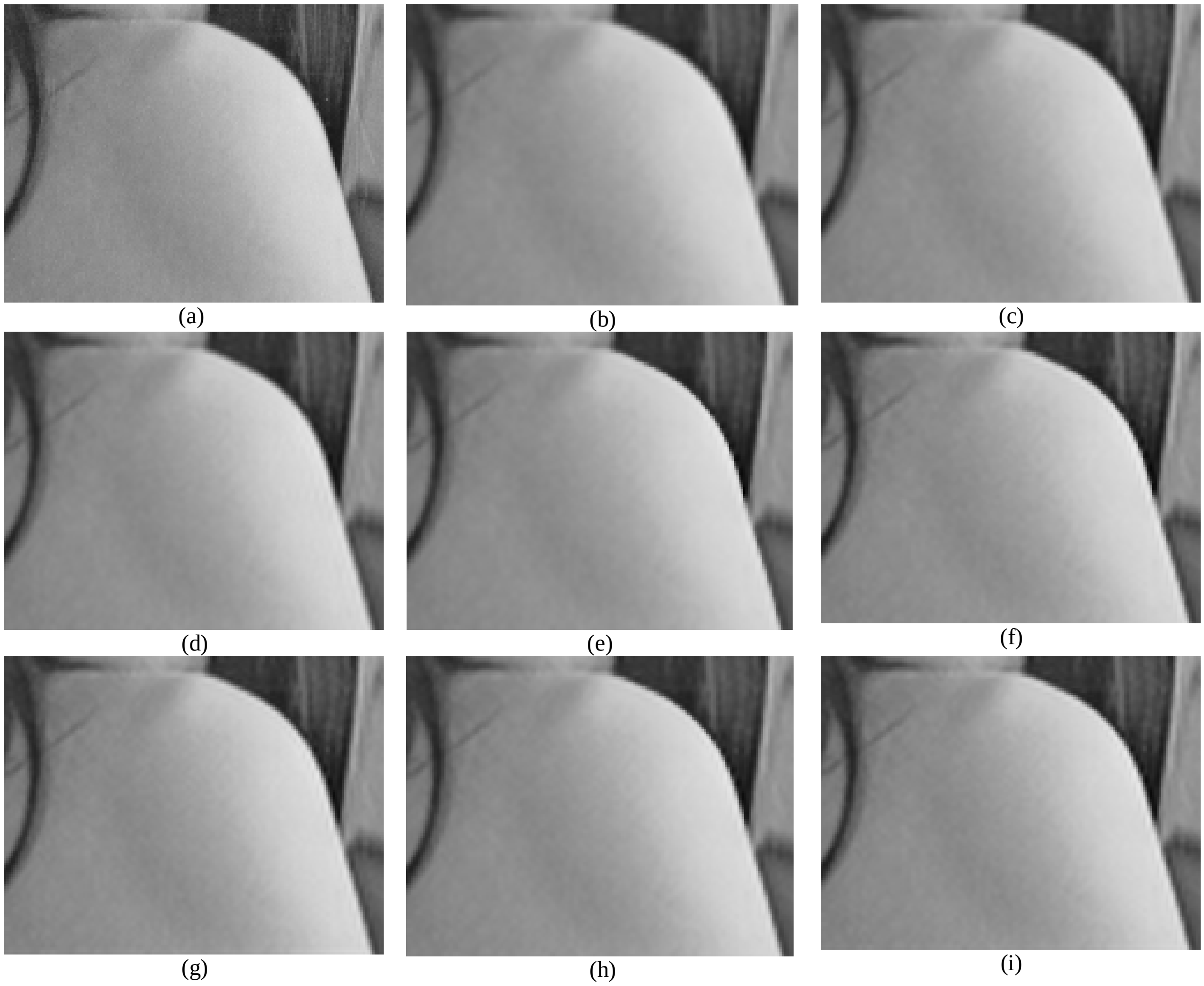


Fig. 6.11 Subjective evaluation of the selected low frequency green rectangular region (127×164) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bicubic; (c) Lanczos-3; (d) DCT; (e) LAL (P7); (f) FWL (P8); (g) ILLGP (P3); (h) CS-I (P10); (i) CS-II (P11)

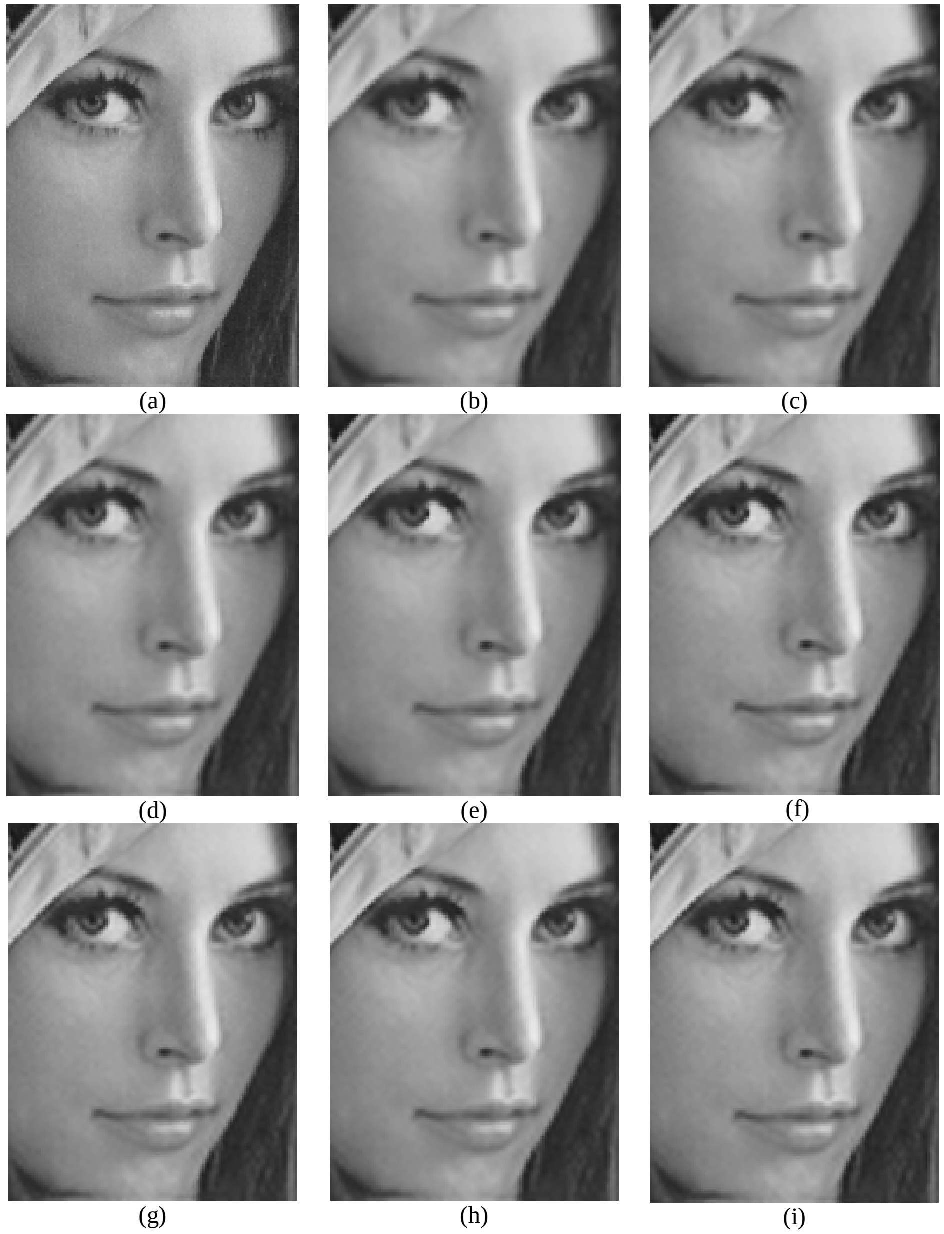


Fig. 6.12 Subjective evaluation of the selected medium frequency orange rectangular region (164×125) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bicubic; (c) Lanczos-3; (d) DCT; (e) LAL (P7); (f) FWL (P8); (g) ILLGP (P3); (h) CS-I (P10); (i) CS-II (P11)

Fig. 6.13 Subjective evaluation of the selected high frequency yellow rectangular region (123×174) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bicubic; (c) Lanczos-3; (d) DCT; (e) LAL (P7); (f) FWL (P8); (g) ILLGP (P3); (h) CS-I (P10); (i) CS-II (P11)

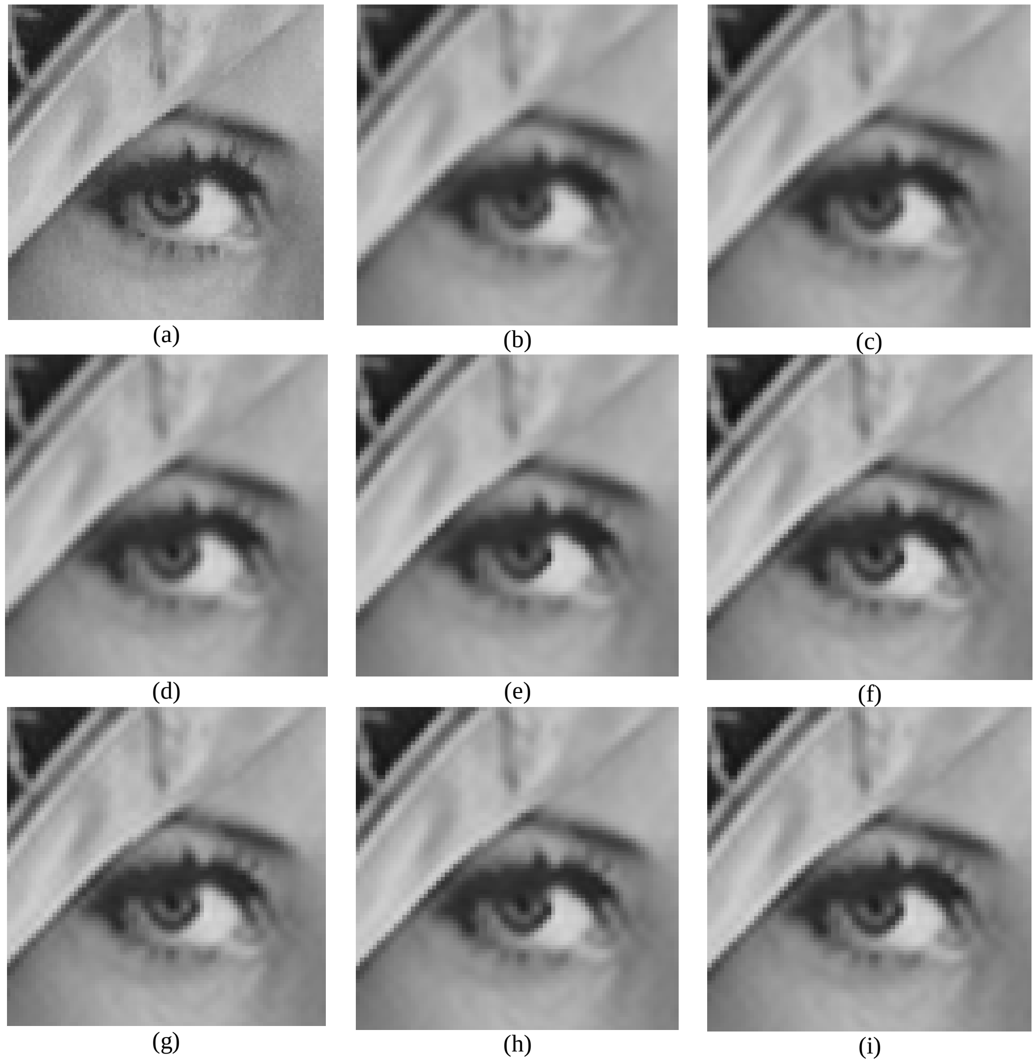
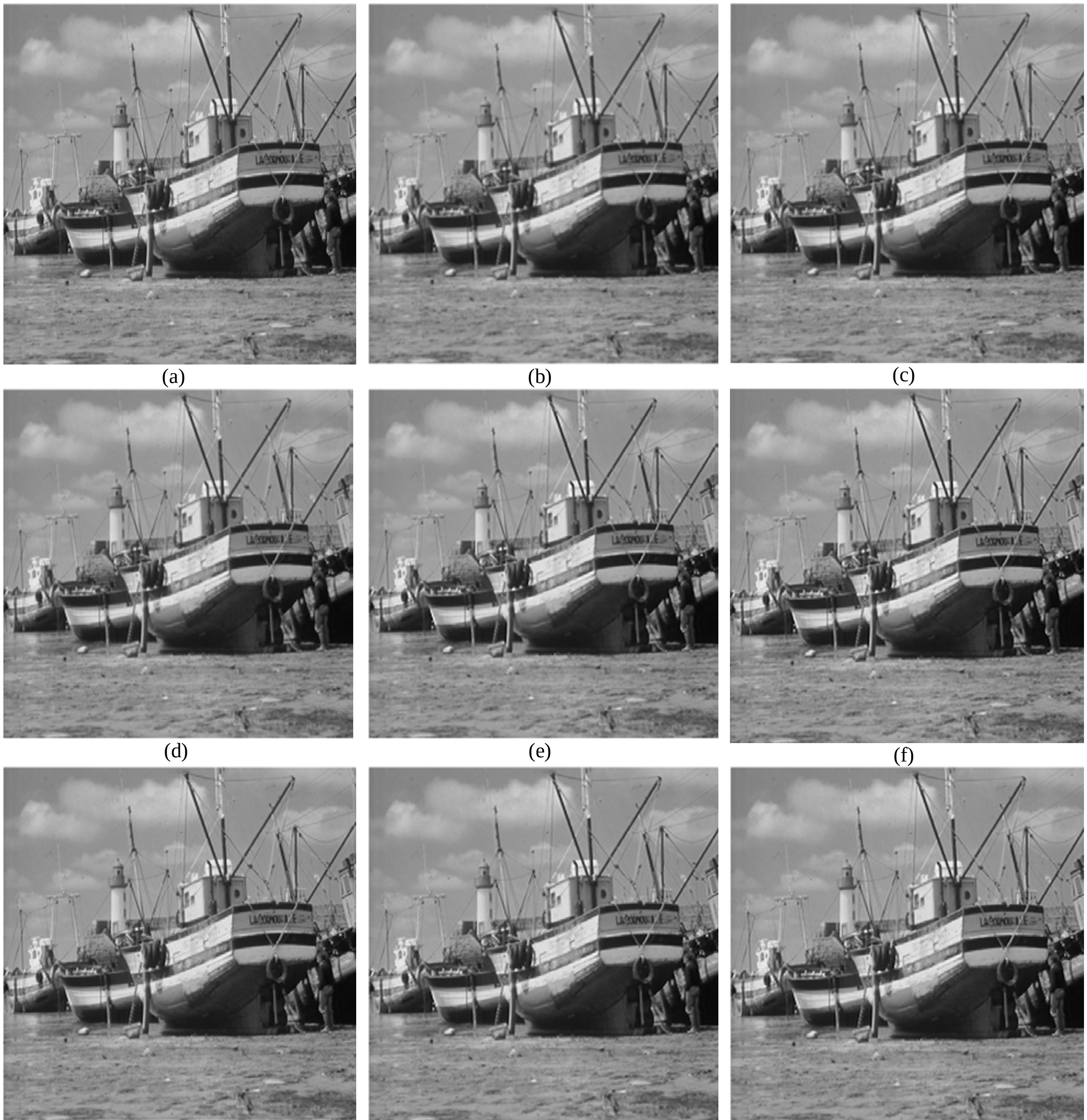


Fig. 6.14 Subjective evaluation of the selected blue rectangular (76×76) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bicubic; (c) Lanczos-3; (d) DCT; (e) LAL (P7); (f) FWL (P8); (g) ILLGP (P3); (h) CS-I (P10); (i) CS-II (P11)

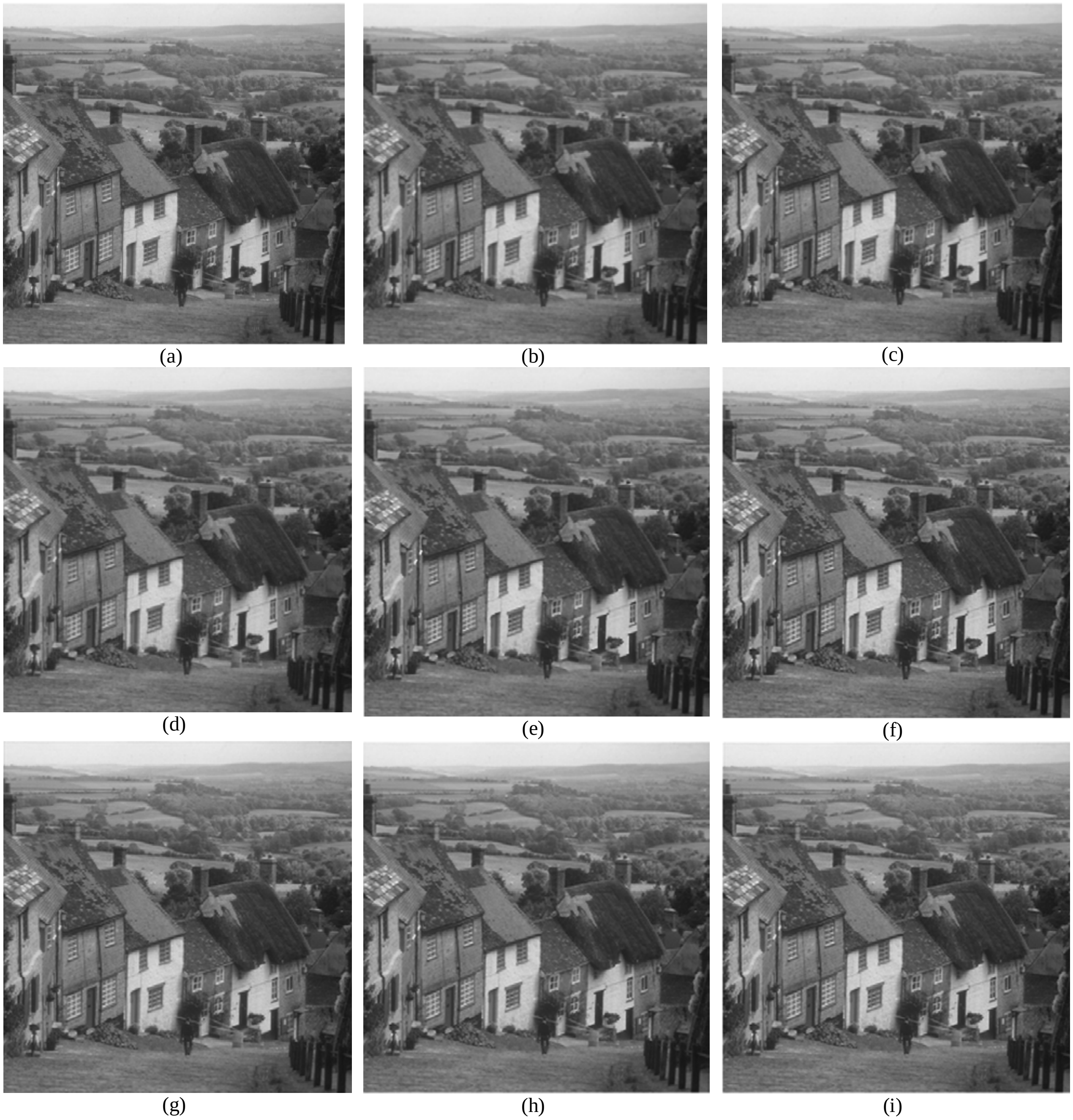


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 6.15 Subjective evaluation of Boat (512×512) image using various up-sampling schemes at 4:1 compression ratio: (a) Original; (b) Bicubic; (c) Lanczos-3; (d) DCT; (e) LAL (P7); (f) FWL (P8); (g) ILLGP (P3); (h) CS-I (P10); (i) CS-II (P11)



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 6.16 Subjective evaluation of Goldhill (512×512) image using various up-sampling schemes at 4:1 compression ratio: (a) Original; (b) Bicubic; (c) Lanczos-3; (d) DCT; (e) LAL (P7); (f) FWL (P8); (g) ILLGP (P3); (h) CS-I (P10); (i) CS-II (P11)

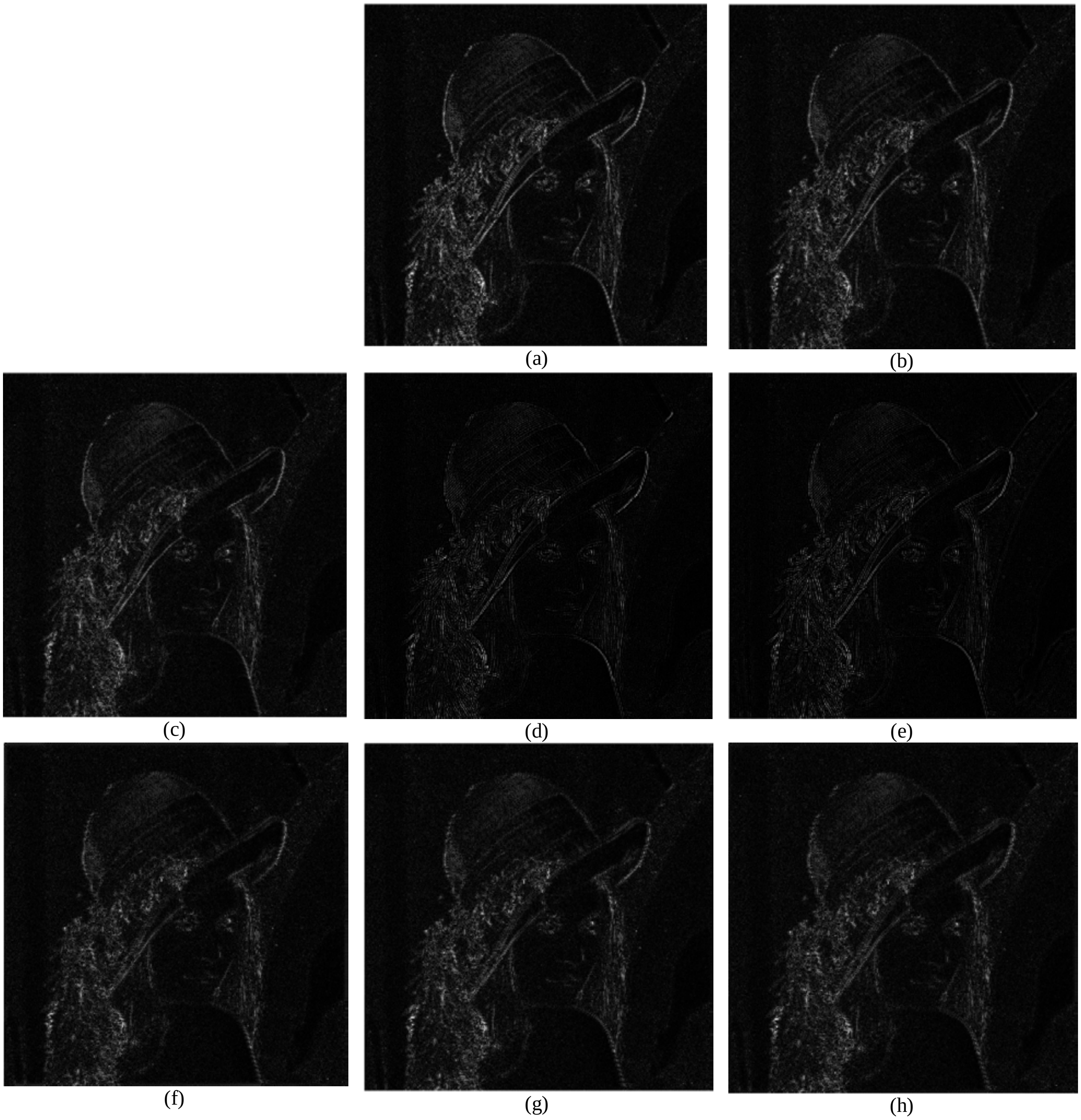


Fig. 6.17 Error image of Lena (512×512) using various up-sampling schemes at 4:1 compression ratio: (a) Bicubic; (b) Lanczos-3; (c) DCT; (d) LAL (P7); (e) FWL (P8); (f) ILLGP (P3); (g) CS-I (P10); (h) CS-II (P11)

6.3 Results and Discussion

The proposed composite schemes in this chapter utilize the advantages of pre-processing and post-processing algorithms to provide better HF restoration performance. The performance of the composite schemes are compared with their constituent pre-processing and post-processing algorithms to reveal their performance improvements. The figures and tables showing the performance of the existing and proposed algorithms are explained below.

Table 6.3 and 6.4 show the objective performance in terms of PSNR (dB) and UQI of the proposed and existing algorithms at 4:1 and 16:1 compression ratios respectively. Table 6.5 reveals the average PSNR and UQI comparison of different algorithms for various video sequences at 4:1 compression ratio. PSNR (dB) vs. frame index plot corresponding to different sequences is given in Fig. 6.8 and Fig 6.9. The performance of the composite scheme CS-I in terms of PSNR is compared with its constituent pre-processing and post-processing algorithms, GIL and LAL, is shown in Fig. 6.8. Likewise, in Fig. 6.9, the performance of the composite scheme CS-II is compared with its constituent pre-processing and post-processing algorithms, ILLGP and CFWL respectively. The computational complexity in terms of CPU execution time of various algorithms is shown in Table 6.6.

For subjective evaluation, results of Lena (512×512), Boat (512×512) and Goldhill (512×512) images at 4:1 compression ratio are shown in Fig. 6.10, Fig. 6.15 and Fig. 6.16 using various up-sampling schemes. In case of Fig. 6.10, four distinct regions with different features and thus different signal characteristics such as low, medium, high and their combinations are marked. Performance at these regions is analyzed. For this purpose, the output images at these regions are enlarged and shown in Fig. 6.11, Fig. 6.12, Fig. 6.13 and Fig. 6.14. The error images of Lena (512×512) corresponding to various schemes are given in Fig. 6.17.

It may be observed from the Table 6.3 that the composite scheme, CS-I achieves a better PSNR and UQI gain over LAL for most of the images and video sequences. In case of images like Lena, Baboon and Crowd, CS-I gives a PSNR hike of 0.45 dB, 0.39dB and 0.37 dB over LAL at 4:1 compression ratio. Likewise, it shows noticeable PSNR gain of 0.507 dB and 0.565 dB over DCT in case of Lena and Barbara.

The composite scheme, CS-II is developed by combining ILLGP and CFWL. Experimental results show that, for most of the images and video sequences, CS-II gives better performance than ILLGP and CFWL schemes at 4:1 compression ratio as depicted in Table 6.3, Table 6.5 and Fig. 6.9. In case of sequences like Bus, Akiyo, Mobile and Coastguard, CS-II attains the considerable PSNR gain of 1.026 dB, 0.755 dB, 0.726 dB and 0.872 dB over DCT. Similarly, in case of images like Lena, Barbara, Cameraman and Living room, CS-II shows the PSNR improvement of 0.539 dB, 0.598 dB, 0.631 dB and 0.466 dB over DCT. In addition, it also shows better PSNR gain over CS-I algorithm because of improved nonlinearity due to incorporation of fuzzy rule base. Unlike CS-I, CS-II algorithm makes use of 5×5 kernel and 3×3 kernel at the pre-processing and post-processing phase, respectively. These global and local kernels enhance the HF information corresponding to 5×5 and 3×3 neighborhood resulting in improved performance over CS-I.

Computational complexity in terms of CPU execution time of the proposed and existing algorithms is evaluated with respect to increase in image dimension and is presented in Table 6.6. The execution time of the composite algorithms rises exponentially with respect to image dimension as it employs local, two-pass post-processing scheme. Hence, the composite schemes are more suitable for the images of small dimensions. The execution time of composite algorithms is found to be more than its constituent pre-processing and post-processing algorithms. The ILLGP pre-processing scheme is computationally less complex than CFWL post-processing scheme because it is a global processing scheme and operates on images of smaller dimension prior to image up-scaling. On the other hand, CFWL is a local fuzzy based two-pass post-processing scheme and operates on up-scaled images of larger dimension and hence consumes much more computational time than ILLGP. The composite scheme, CS-II which is developed by combining the ILLGP and CFWL schemes, has more computation time than either of the standalone schemes as illustrated in Table 6.6.

The execution time of CS-II is more than CS-I scheme because of employing fuzzy based mapping scheme in its post-processing phase. The rise in computation time of CS-II is also because of the employment of 5×5 convolution kernel in the pre-processing phase. CS-I, being a non-fuzzy composite scheme employs direct normalized mapping at the post-processing phase and 3×3 convolution kernel at the pre-processing phase and hence is computationally less complex than CS-II. So, in the perspective of real-time applications, pre-processing schemes like ILLGP and GIL are preferred because of their reduced computational complexity. In contrast, if quality is of the prime importance, then the composite scheme must be preferred because of its improved nonlinearity to reduce the non-uniform blurring effectively in the up-scaled images. Hence, the composite schemes are preferred for various off-line applications.

Lena image has various regions of different signal characteristics such as low, medium, high frequencies and their combinations. Four distinct regions corresponding to these signal patterns are enlarged and shown in Fig. 6.11, Fig. 6.12, Fig. 6.13 and Fig. 6.14. It may be observed from these figures that the composite schemes restore the HF details much effectively and give less blurring particularly at the fast changing and edge regions. In case of CS-I and CS-II, the edges are more pronounced and fine details are well restored than the standalone pre-processing and post-processing schemes. In addition, the low and medium frequency regions are well preserved in the up-sampled images without much deviation. Therefore, it is presumed that the proposed algorithms enhance the high variance regions much more than flat and slowly varying regions. Hence, the lost HF information is more enhanced which was degraded the most during the up-sampling process resulting in a better visual quality.

The absolute error image of various existing and proposed algorithms is shown in Fig. 6.17. The absolute error is an indicator for the degree of HF restoration performance of the algorithms. Less the error, better is the restoration performance and vice versa. It is apparent from the figure that the absolute error is much reduced in case of the proposed composite algorithms, CS-I and CS-II which indicate better HF restoration performance than the standalone schemes.

6.4 Conclusion

The proposed composite schemes exploit the advantages of pre-processing and post-processing operations for efficient restoration of HF and VHF information in the up-scaled images. The standalone pre-processing algorithm such as ILLGP and GIL are developed to meet the real-time requirements because of their reduced computational complexity and hence are suitable for online applications. The composite schemes on the other hand are developed for the quality enhancement in the up-scaled image at the expense of higher computational complexity and may be employed for offline applications.

As the nonlinearity of an algorithm increases, so does its adaptability to various image statistical characteristics. In the perspective of the HF and VHF restorations, CS-I and CS-II show better objective and subjective performance than most of the widely used interpolation techniques. However, the later shows better performance than the former because it makes use of fuzzy weighted Laplacian based post-processing scheme whose nonlinearity is enhanced by varying the various parameters such as slope, width and number of input and output membership functions. The combination of global pre-processing and local fuzzy based post-processing raises the nonlinearity of CS-II for better HF restoration performance.

The proposed composite schemes achieve considerable improvement in image quality over existing schemes for different types of images because of its adaptability to different local signal characteristics. The performance improvement is gained by enhancing the HF and VHF regions that have been degraded the most while preserving the medium and low frequency regions that have been degraded the least during the up-sampling process.

The use of filter kernel of different sizes, i.e. 5×5 for pre-processing and 3×3 for post-processing enables the composite scheme, CS-II to enhance the HF and VHF components corresponding to 3×3 and 5×5 neighbourhood resulting in an improved HF restoration performance than CS-I and other existing schemes.

The blocking pattern that arises in ILLGP pre-processing scheme is due to the use of a 5×5 Laplacian of Laplacian kernel. The pattern is more significant in edge and fast changing region and remains insignificant in flat and slowly varying regions. However, this artifact is reduced in composite scheme, CS-II due to the incorporation of fuzzy rule base. The inclusion of FIS provides a very small amount of blurring that removes the fine blocking artifacts in the up-scaled images resulting in improved image quality.

Both of the proposed composite schemes show better objective and subjective performance, irrespective of image types under varying conditions. In most of the up-sampled images, blurring is considerably reduced with a more pronounced edge and fine details as compared to standalone pre-processing, post-processing and existing schemes.

Chapter 7

Conclusion

Image up-scaling through interpolation is an area of great interest in recent years and is extensively used in many applications like video streaming, multimedia, internet technologies, HDTVs, display and printing industries. Various polynomial based interpolation schemes such as nearest-neighbor, bilinear, bicubic, cubic-spline and lanczos are used for such applications for their reduced computational complexity. It is apparent that the polynomial based interpolation schemes are computationally efficient but produce undesirable artifacts such as blurring at the edges. Though the edge-directed and transform domain interpolation schemes though preserve the edge information and fine details effectively than polynomial based interpolation schemes, they are computationally more complex. Hence, there are further scopes to develop efficient up-scaling schemes which are not only computationally efficient but also produce a better visual quality by preserving the fine details and edge information.

Hence, in this current research work, efforts are made to improve the performance of the existing 2-D polynomial based interpolation schemes by incorporating various spatial domain pre-processing, post-processing and composite techniques so as to obtain a better up-scaled image quality along with reduced computational complexity.

In order to overcome the challenges posed by the existing interpolation schemes, the following pre-processing, post-processing and composite algorithms have been proposed in this dissertation.

i. *Laplacian of Laplacian (**LLGP**) based Global Pre-processing Scheme [P1]*
ii. *Hybrid Global Pre-processing (**HGP**) Scheme [P2]*
iii. *Iterative Laplacian of Laplacian based Global Pre-processing (**ILLGP**) Scheme [P3]*
iv. *Unsharp Masking based Pre-processing (**UMP**) Scheme [P4]*
v. *Iterative Unsharp Masking (**IUM**) Scheme [P5]*

vi. *Error based Up-sampling (**EU**) Scheme [P6]*

vii. *Local Adaptive Laplacian (**LAL**) based Post-processing Algorithm [P7]*

viii. *Fuzzy Weighted Laplacian (**FWL**)based Post-processing Algorithm [P8]*

ix. *Legendre Functional Link Artificial Neural Network (**LFLANN**) based Post-processing Algorithm [P9]*

x. *Composite Scheme **(CS-I)** using Iterative Laplacian and Local Adaptive Laplacian [P10]*

xi. *Composite Scheme **(CS-II)** using Iterative Laplacian of Laplacian and Fuzzy Weighted Laplacian [P11]*

The comparative performance analysis of the best algorithms in their respective categories is presented in the next section.

7.1 Comparative Analysis

Comparative performance analysis of the algorithms developed in this thesis: LLGP [P1], ILLGP [P2], IUM [P5], LAL [P7], FWL [P8], LFLANN [P9], CS-II [P11] is presented here. The 20 publicly available (512×512) images and the performance metric, which have been used throughout the thesis, have also been employed here. The best performance value obtained for each of the metric is highlighted in bold.

The execution time comparison of various existing and proposed scheme is given in Table 7.3. In addition, peak-signal-to-noise-ratio (PSNR) in dB and universal quality index (UQI) are measured to determine the objective performance of the proposed algorithms. The PSNR and UQI comparisons of different schemes are given in Table 7.1 and Table 7.2, respectively.

The overall performances of the proposed algorithms are determined through the average of quantitative results experimented over 20 different images. So, average PSNR and UQI of the proposed algorithms over 20 different images are given in Table 7.4. The histogram of overall PSNR of existing and proposed algorithms is given in Fig. 7.1. The overall ranking of the proposed algorithms is done based on overall PSNR and UQI performances in Table 7.5. The subjective performance of the proposed algorithms is shown in Fig. 7.2.

Table 7.1 PSNR (dB) comparison of different schemes at 4:1 compression ratio for various (512×512) images

Image	**Bilinear [5]**	**Bicubic [4]**	**Lanczos3 [12]**	**DCT [74]**	**LLGP [P1]**	**ILLGP [P2]**	**IUM [P5]**	**LAL [P7]**	**FWL [P8]**	**LFLANN [P9]**	**CS-II [P11]**
	Existing Schemes				**Proposed Schemes**						
Lena	32.704	34.148	34.813	35.022	35.179	35.439	**35.569**	35.080	35.250	35.454	35.562
Boat	28.940	29.951	30.375	30.466	30.734	30.879	30.946	30.841	30.870	30.848	**30.983**
Livingroom	28.617	29.557	29.977	30.128	30.367	30.524	**30.595**	30.311	30.439	30.308	30.594
Fingerprint	28.045	30.632	31.722	32.133	32.407	32.798	**32.903**	31.138	31.345	31.966	32.458
Goldhill	30.574	31.405	31.725	31.716	31.935	32.080	32.166	31.993	32.092	32.027	**32.197**
Pirate	30.027	31.058	31.490	31.607	31.872	32.047	32.103	31.952	31.986	32.039	**32.161**
Baboon	33.588	35.014	35.662	35.889	35.897	36.142	**36.315**	35.850	36.042	35.737	36.271
Barbara	24.925	25.352	25.428	25.183	25.642	25.744	25.731	25.562	25.692	25.630	**25.781**
Bridge	25.728	26.504	26.826	26.918	27.205	27.333	27.348	27.160	27.255	27.102	**27.367**
Cat	30.949	31.982	32.427	32.562	32.889	33.046	**33.073**	32.584	32.826	31.842	33.037
Crowd	30.984	32.667	33.451	33.768	34.155	34.415	**34.467**	34.035	34.026	33.664	34.430
Cycle	21.208	21.895	22.154	22.129	22.498	22.633	22.631	22.452	22.543	**22.960**	22.661
F16	30.379	31.543	32.104	32.722	32.281	32.488	32.653	32.353	32.319	**32.790**	32.576
House	29.248	30.314	30.807	30.862	31.422	31.596	**31.628**	30.980	31.304	30.855	31.585
Lake	28.945	30.022	30.495	30.793	30.870	31.021	**31.069**	30.617	30.700	30.930	30.965
Cameraman	33.214	35.757	37.216	37.832	37.846	38.349	**38.700**	36.963	37.381	35.979	38.463
Elaine	32.534	33.117	33.309	33.284	33.255	33.371	33.478	33.315	33.413	33.244	**33.502**
Mandrill	23.045	23.630	23.859	23.925	24.184	24.288	**24.296**	24.066	24.178	24.180	24.282
Peppers	31.180	31.991	32.329	**32.747**	32.520	32.652	32.727	32.509	32.407	32.157	32.625
Ruler	12.335	12.613	12.673	12.600	12.843	12.890	**12.911**	12.584	12.823	13.415	12.765

Experimental results show that the proposed algorithms yield better objective and subjective performance than the state-of-art schemes. Based on the overall performance, presented in Table 7.4, IUM demonstrates the best performance among the proposed algorithms and state-of-art algorithms as well. The second and third best performances are given by CS-II and ILLGP, respectively. Likewise, the subjective performance of these algorithms is found to be better than other proposed and existing schemes with much pronounced edges and better preserved fine details as illustrated in Fig. 7.2.

The ILLGP scheme exhibits better HF restoration performance than LLGP because of employing higher order derivative operator than LLGP as depicted in Table 7.1. Since LLGP and ILLGP are pre-processing techniques and operate on LR images, they consume less processing time than the local adaptive post-processing schemes such as LAL and FWL as depicted in Table 7.3.

Table 7.2 UQI comparison of different schemes at 4:1 compression ratio for various (512×512) images

Image	**Bilinear [5]**	**Bicubic [4]**	**Lanczos3 [12]**	**DCT [74]**	**LLGP [P1]**	**ILLGP [P2]**	**IUM [P5]**	**LAL [P7]**	**FWL [P8]**	**LFLANN [P9]**	**CS-II [P11]**
	Existing Schemes				**Proposed Schemes**						
Lena	0.9922	0.9945	0.9953	0.9955	0.9957	0.9959	**0.9961**	0.9956	0.9958	0.9960	**0.9961**
Boat	0.9801	0.9845	0.9860	0.9863	0.9872	0.9876	0.9878	0.9876	0.9877	0.9878	**0.9880**
Livingroom	0.9761	0.9811	0.9829	0.9835	0.9846	0.9851	0.9853	0.9843	0.9850	0.9845	**0.9854**
Fingerprint	0.9785	0.9888	0.9915	0.9922	0.9929	0.9934	**0.9936**	0.9908	0.9913	0.9924	0.9931
Goldhill	0.9879	0.9901	0.9908	0.9908	0.9914	0.9916	0.9918	0.9915	0.9917	0.9914	**0.9919**
Pirate	0.9853	0.9885	0.9896	0.9899	0.9906	0.9910	0.9911	0.9908	0.9909	0.9909	**0.9912**
Baboon	0.9924	0.9946	0.9953	0.9956	0.9956	0.9959	**0.9960**	0.9956	0.9958	0.9955	**0.9960**
Barbara	0.9631	0.9669	0.9675	0.9657	0.9693	0.9701	0.9700	0.9687	0.9698	0.9692	**0.9705**
Bridge	0.9693	0.9747	0.9767	0.9772	0.9789	0.9795	0.9795	0.9787	0.9793	0.9782	**0.9798**
Cat	0.9919	0.9937	0.9943	0.9945	0.9949	**0.9951**	**0.9951**	0.9946	0.9949	0.9938	**0.9951**
Crowd	0.9894	0.9929	0.9941	0.9946	0.9951	0.9953	**0.9954**	0.9949	0.9950	0.9944	**0.9954**
Cycle	0.9323	0.9437	0.9475	0.9473	0.9526	0.9540	0.9539	0.9526	0.9538	**0.9567**	0.9551
F16	0.9847	0.9885	0.9900	0.9913	0.9905	0.9909	0.9912	0.9907	0.9907	**0.9916**	0.9912
House	0.9719	0.9785	0.9809	0.9812	0.9837	0.9843	**0.9844**	0.9820	0.9835	0.9817	**0.9844**
Lake	0.9895	0.9919	0.9928	0.9932	0.9934	0.9936	**0.9937**	0.9930	0.9932	0.9935	0.9936
Cameraman	0.9959	0.9977	0.9984	0.9986	0.9986	0.9988	**0.9989**	0.9983	0.9985	0.9982	0.9988
Elaine	0.9913	0.9924	0.9928	0.9927	0.9927	0.9929	**0.9931**	0.9928	0.9930	0.9927	**0.9931**
Mandrill	0.8957	0.9114	0.9169	0.9187	0.9250	0.9271	0.9271	0.9228	0.9260	0.9250	**0.9278**
Peppers	0.9923	0.9937	0.9942	**0.9947**	0.9944	0.9946	**0.9947**	0.9944	0.9943	0.9941	0.9946
Ruler	0.5188	0.5735	0.5897	0.5920	0.6293	0.6351	**0.6395**	0.6088	0.6371	0.6484	0.6337

Table 7.3 Execution time comparison of the proposed and existing algorithms at 4:1 CR

Images of different size	**Execution time in Seconds for different interpolation schemes**										
	Bilinear [5]	**Bicubic [4]**	**Lanczos3 [12]**	**DCT [74]**	**LLGP [P1]**	**ILLGP [P2]**	**IUM [P5]**	**LAL [P7]**	**FWL [P8]**	**LFLANN [P9]**	**CS-II [P11]**
Clock (200×200)	0.0141	0.0150	0.0151	0.0518	0.0946	0.1300	0.1445	3.1982	8.6076	0.8933	11.7605
Lena (256×256)	0.0147	0.0152	0.0157	0.0633	0.0987	0.1328	0.1471	7.7949	21.5439	1.4455	27.5542
Fruit (377×321)	0.0153	0.0163	0.0168	0.1729	0.1027	0.1395	0.1506	56.9265	82.9838	2.6542	97.2611
Lena (512×512)	0.0201	0.0215	0.0222	0.1673	0.1137	0.1420	0.1564	335.1430	384.076	5.6765	415.4207
Pentagon (1024×1024)	0.0354	0.0409	0.0442	0.6372	0.1281	0.1767	0.1905	6140.056	6399.084	22.7194	6462.9129

Table 7.4 Average PSNR and UQI comparison of different schemes over 20 different (512×512) images at 4:1 CR

Average Metric	Bilinear [5]	Bicubic [4]	Lanczos3 [12]	DCT [74]	LLGP [P1]	ILLGP [P2]	IUM [P5]	LAL [P7]	FWL [P8]	LFLANN [P9]	CS-II [P11]
	Existing Schemes				Proposed Schemes						
Average PSNR(dB)	28.358	29.457	29.942	30.114	30.300	30.486	**30.565**	30.117	30.244	30.156	30.513
Average UQI	0.9539	0.9610	0.9634	0.9638	0.9668	0.9676	**0.9679**	0.9654	0.9674	0.9678	0.9677

Table 7.5 Overall ranking of the proposed algorithms based on average PSNR and UQI images at 4:1 CR

Proposed schemes	LLGP [P1]	ILLGP [P2]	IUM [P5]	LAL [P7]	FWL [P8]	LFLANN [P9]	CS-II [P11]
Rank Score (based on PSNR)	4	3	**1**	7	5	6	2
Rank Score (based on UQI)	6	4	**1**	7	5	2	3

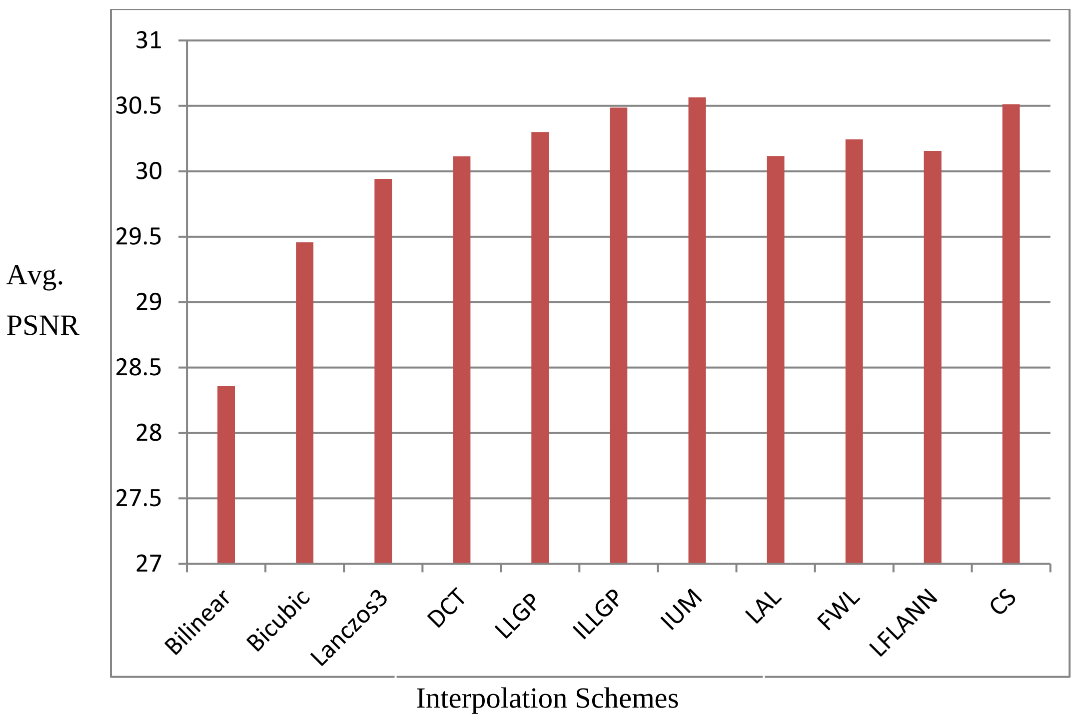


Fig 7.1 Average PSNR comparison of various existing and proposed schemes over 20 different (512×512) images at 4:1 compression ratio.

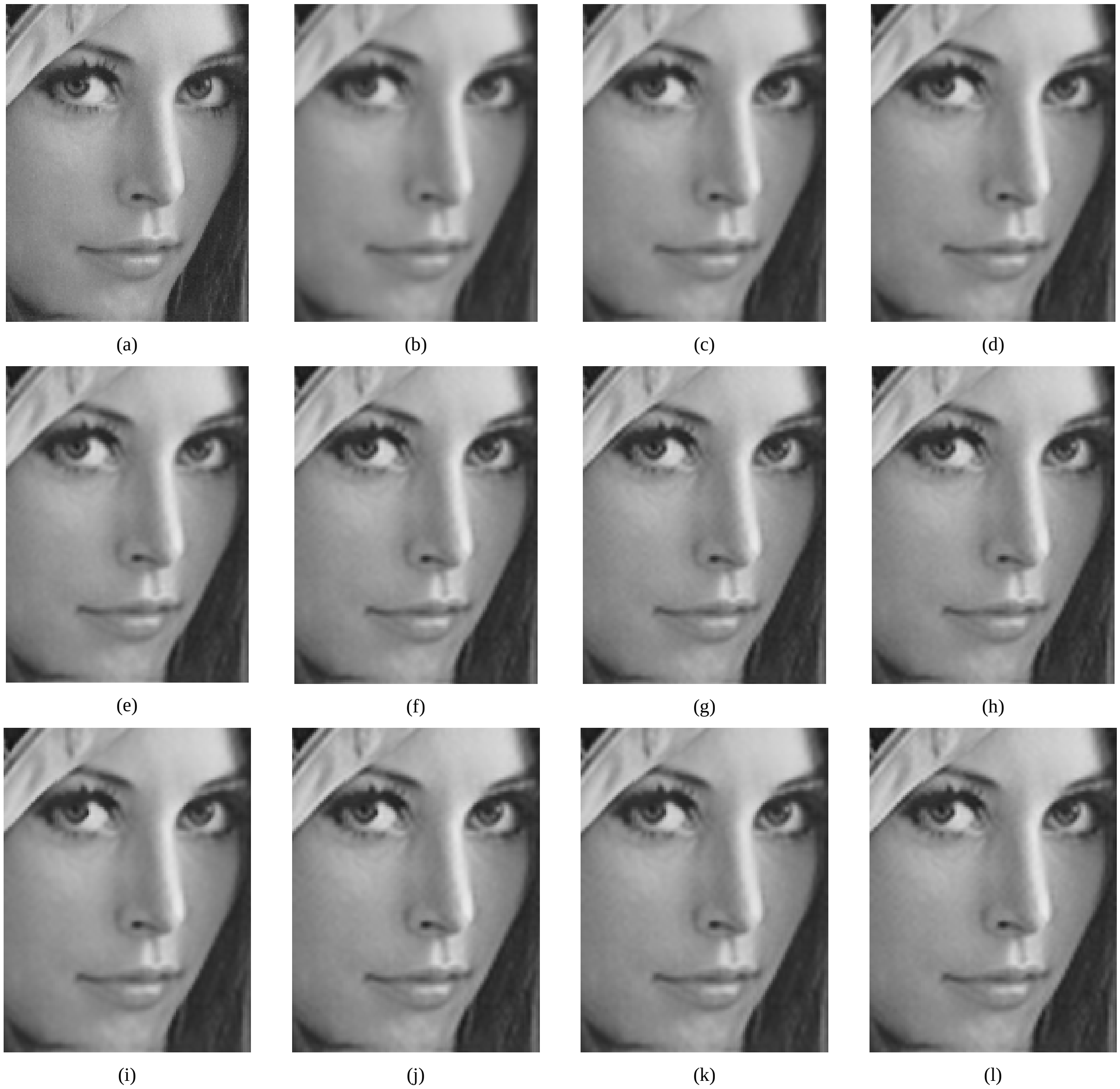


Fig. 7.2 Subjective evaluation of the selected rectangular region (164×125) of Lena (512×512) image using various up-sampling scheme at 4:1 compression ratio: (a) Original; (b) Bilinear; (c) Bicubic; (d) Lanczos-3; (e) DCT; (f) LLGP; (g) ILLGP; (h) IUM; (i) LAL; (j) FWL; (k) LFLANN; (l) CS-II

As per the experimental results shown in Table.7.1 and Table 7.2, The LAL post-processing scheme shows better objective performance in terms of PSNR and UQI than existing schemes because of locally enhancing the image regions as per the local variance. However, the performance of FWL is found to be better than LAL due to its fuzzy based mapping. The composite scheme (CS-II) developed by combining ILLGP and a fuzzy weighted Laplacian post-processing scheme gives better performance than the standalone schemes because of improved nonlinearity as shown in Fig. 7.1, Fig. 7.2, Table 7.4 and Table 7.5.

Moreover, the IUM scheme provides overall better performance amongst the proposed pre-processing, post-processing and composite algorithms in terms of PSNR gain and computational complexity. In addition, its performance is comparable to various sparse based up-scaling algorithms such as: Sparse representation based image interpolation with nonlocal autoregressive modeling (SR-NAM) [121] and Convolutional sparse coding for image super-resolution (CSC) [123]. The PSNR (dB) comparison of the proposed IUM scheme with different sparse based super-resolution schemes is given in Table 7.6 and Table 7.7.

Table 7.6 PSNR (dB) comparison of the proposed IUM scheme with sparse based SR-NAM scheme at 4:1 CR

Image 256×256	**PSNR (dB)**	
	SR-NAM [121]	**IUM**
House	33.520	33.502
Cameraman	25.940	27.163
Lena	35.010	32.441

Table 7.7 PSNR (dB) comparison of the proposed IUM scheme with sparse based CSC scheme at 4:1 CR

Image 512×512	**PSNR (dB)**	
	CSC [123]	**IUM**
Pirate	30.970	32.103
Bridge	27.840	27.348
Lena	36.660	35.570

From Table 7.3 and Table 7.8, it may be observed that the pre-processing algorithms take less processing time than the post-processing and composite algorithms as a result of employing global-based techniques. In contrast, the post-processing and composite techniques have more computational complexity because of the employment local-based algorithms. Hence, the pre-processing techniques such as LLGP, ILLGP and IUM are suitable candidates for online applications while LAL, FWL and CS-II are well suited for various off-line applications. The computational complexities of the proposed algorithms are discussed in the next section.

7.2 Computational Complexities of the Proposed Algorithms

Complexity of an algorithm is related to the number of multiplications and additions involved to obtain the final output. Since Lanczos-3 interpolation is employed as the basic interpolation paradigm for the various proposed pre-processing, post-processing and composite algorithms, its complexity is necessary for computing the complexity of all the proposed algorithms. Lanczos-3 interpolation kernel makes use of 6×6 mask for interpolating a pixel in a 6×6 neighborhood and therefore requires 36 multiplications and 35 additions per pixel. In case of LLGP algorithm, a 5×5 Laplacian of Laplacian (LOL) mask is used for HF extraction. However, out of 25 filter coefficients, 16 coefficients are zero. Hence, the convolution is performed using the remaining 9 nonzero coefficients. Therefore, there will be 9 multiplications and 8 additions per pixel. In addition, there will be one multiplication and one addition per pixel for intensity scaling and superimposition of HF extract with LR image. Furthermore, 36 multiplications and 35 additions are required for Lanczos-3 based up-scaling. So, there is a sum total of 46 multiplications and 44 additions required per pixel for LLGP algorithm.

In case of HGP algorithm, HF extract is obtained by performing LOL and Laplacian operations in series. LOL operation requires 9 multiplications and 8 additions per pixel whereas Laplacian operation requires 5 multiplications and 4 additions per pixel. Furthermore, there will be one multiplication and one addition per pixel required for intensity scaling and superimposition of HF extract with LR image. Finally, the up-scaling operation using Lanczos-3 interpolation requires 36 multiplications and 35 additions. Hence, there is a sum total of 51 multiplications and 48 additions required per pixel in case of HGP algorithm.

Similarly, in case of ILLGP algorithm, LLGP is operated twice for HF extraction and hence, requires 18 multiplications and 16 additions per pixel. The rest of the operation is same as LLGP algorithm. So, there is a sum total of 55 multiplications and 52 additions per pixel. Therefore, ILLGP is computationally more complex than LLGP and HGP as shown in Table 7.8.

Unsharp masking pre-processing (UMP) scheme makes use of an LPF of mask size 3×3 for blurring an input image and therefore requires 9 multiplications and 8 additions per pixel. The blurred image is subtracted from the original to generate the HF extract and requires one subtraction per pixel. Superimposition and scaling of HF extract with the input image needs one multiplication and one addition per pixel. Finally, the Lanczos-3 up-scaling process requires 36 multiplications and 35 additions per pixel. Hence, overall 46 multiplications and 45 additions are required per pixel for UMP pre-processing scheme.

Iterative unsharp masking (IUM) is an iterative scheme that makes use of an LPF filter mask of 3×3 dimension iteratively for HF extract. It requires 9 multiplications and 8 additions per pixel to obtain the blurred image. To obtain the unsharp mask, one subtraction is required per pixel. Hence, a sum total of 9 multiplications and 9 additions / subtractions are required for a single iteration. So, the total number of multiplications and additions / subtractions after 7 iterations would be 63 and 63 respectively. Subsequently, one addition per pixel is required for superimposing the HF extract to an LR image. Finally, Lanczos-3 interpolation requires 36 multiplications and 35 additions per pixel for up-scaling. Therefore, overall 99 multiplications and 99 additions are required per pixel for IUM scheme and so is computationally more complex than UMP scheme.

Error based up-sampling (EU) is a two-pass HF predictive scheme. During the first-pass a down-sampled image is up-scaled using Lanczos-3 interpolation which requires 36 multiplications and 35 additions per pixel. To obtain the HF extract, the Lanczos-3 interpolated image is subtracted from LR input image which requires one subtraction per pixel. Furthermore, one multiplication and one addition are required per pixel for scaling and superimposing the HF extract with the LR input image. Subsequently, the image is further up-sampled using Lanczos-3 interpolation which requires 36 multiplications and 35 additions per pixel. Hence a sum total of 73 multiplications and 72 additions per pixel are required for EU scheme. Hence, it is computationally less complex than IUM as illustrated in Table 7.8.

Local adaptive Laplacian (LAL) is a local based two-pass post-processing scheme. During the first pass, an up-scaled image using Lanczos-3 interpolation is generated which requires 36 multiplications and 35 additions per pixel. Furthermore, Local variance is calculated which requires 3 multiplications and 9 additions per pixel. During the second pass, 3×3 local variance is further computed followed by normalization and updation of the filter coefficients. The normalization and updation of the filter coefficients require 2 and 4 multiplications per pixel respectively. The extraction of HF contents using LAL operator requires 5 multiplications and 4 additions per pixel. Scaling and superimposition of HF extracts require one multiplication and one addition per pixel. Finally, a sum total of 54 multiplications and 58 additions per pixel are required for LAL algorithm. Since this algorithm is a post-processing scheme and operates on up-scaled images of dimension $2N \times 2N$, a sum total of $216N^2$ multiplications and $232N^2$ additions are required for LAL algorithm.

Fuzzy weighted Laplacian (FWL) algorithm is almost similar to LAL. However, it is computationally more complex than LAL because of fuzzy based mapping which consists of fuzzification, fuzzy if-then rules and defuzzification.

LFLANN is a post-processing scheme and is applied on the Lanczos-3 up-scaled image for restoration of HF contents. The up-scaling using Lanczos-3 interpolation requires 36 multiplications and 35 additions per pixel. The testing phase of LFLANN is taken for calculating the computational complexity. For a particular pixel, its 3×3 neighbourhood is taken for functional expansion using Legendre polynomial. The 9 dimensional pattern vector is expanded to 37 dimensional vector using Legendre expansion which requires 72 multiplications and 36 additions per pixel. Furthermore, the enhanced pattern vector is multiplied with the updated weights and the output is obtained by computing sum of products which requires 37 multiplications and 36 additions per pixel. Hence, a sum total of 145 multiplications and 107 additions per pixel are needed for LFLANN algorithm.

Composite scheme-I is developed by combining global iterative Laplacian (GIL) as pre-processing technique and LAL as post-processing technique. So, the composite scheme is computationally more complex than the standalone pre-processing and post-processing algorithms. The GIL pre-processing scheme makes use of Laplacian operator thrice for HF extraction and so requires 15 multiplications and 12 additions per pixel. Intensity scaling and superimposition of HF extracts require one multiplication and one addition per pixel. The pre-processing scheme is followed by up-scaling using Lanczos-3 interpolation and post-processing using LAL which requires 216 multiplications and 232 additions per pixel. Hence, a sum total of 232 multiplications and 245 additions per pixel are required for CS-I scheme. The detail comparison of GIL, LAL and CS-I is given in Table 7.8.

Table 7.8 Computational complexity comparison of various proposed pre-processing, post-processing and composite algorithms for an $N \times N$ image

Algorithm	Number of multiplications	Number of additions
Lanczos-3 [12]	$36N^2$	$35N^2$
DCT [74]	$5N^2 \log_2 N + 4N^2$	$15N^2 \log_2 N + 2N^2 + 6N$
DASR [116]	$106N^2$	$105N^2$
DSWD [82]	$124N^2$	$126N^2$
LLGP [P1]	$46N^2$	$44N^2$
HGP [P2]	$51N^2$	$48N^2$
ILLGP [P3]	$55N^2$	$52N^2$
UMP [P4]	$46N^2$	$45N^2$
IUM [P5]	$99N^2$	$99N^2$
EU [P6]	$73N^2$	$72N^2$
LFLANN [P9]	$145N^2$	$107N^2$
LAL [P7]	$216N^2$	$232N^2$
GIL	$52N^2$	$48N^2$
CS-I [P10]	$232N^2$	$245N^2$

Table 7.9 Operation counts of the existing and proposed algorithms

Image size $N \times N$	Number of Operations	Lanczos-3 [12]	DCT [74]	DASR [116]	DSWD [82]	LLGP [P1]	HGP-I [P2]	ILLGP [P3]	IUM [P5]
128×128	Multiplications	589824	638976	1736704	2031616	753664	835584	901120	1622016
	Additions	573440	1753856	1720320	2064384	720896	786432	851968	1622016
256×256	Multiplications	2359296	2883584	6946816	8126464	3014656	3342336	3604480	6488064
	Additions	2293760	7996928	6881280	8257536	2883584	3145728	3407872	6488064
512×512	Multiplications	9437184	12845056	27787264	32505856	12058624	13369344	14417920	25952256
	Additions	9175040	35916800	27525120	33030144	11534336	12582912	13631488	25952256

7.3 Conclusion

According to experimental results, shown in Table 7.4 and Table 7.5, the performance of the proposed pre-processing, post-processing and composite schemes is found to be superior than the existing interpolation techniques in terms of objective and subjective measures.

The proposed algorithms: LLGP and ILLGP perform better in terms of PSNR than various existing algorithms for different image types with a more pronounced edge and fine details preservation. Since the ILLGP algorithm is based on 8^{th} order derivative, it is capable of extracting much finer and subtler details of an image and hence yields better performance than LLGP. The LLGP algorithm is based on 4^{th} order derivatives and hence performs less as compared to ILLGP. Since these algorithms are global pre-processing techniques and operate on low-resolution images, have reduced computational complexity and are comparable with DCT as shown in Table 7.3, Table 7.8 and Table 7.9. So, they are suitable for real-time applications. IUM exhibits best performance amongst the pre-processing algorithms for most of the images as evident from Fig 7.2 and Fig. 7.3. It shows better performance because of its ability to restore the most degraded VHF component through signal decomposition using the filter bank which employs unsharp masking iteratively. In addition, IUM is computationally comparable with DCT in case of high resolution images and hence is a preferred candidate for real-time applications.

The proposed algorithms: LAL, FWL and LFLANN are local post-processing schemes which are employed for local HF enhancement of up-scaled images so as to lessen blurring at the edges and fast changing regions. Being local schemes, these algorithms tackle the local HF degradation more effectively than the existing schemes. LAL being a local post-processing scheme, it adaptively enhances high-variance regions more than the low-variance regions resulting in improved objective and subjective performance than DCT and other existing schemes as per the experimental results. However, the mapping technique introduced in this scheme is linear and therefore, the performance is further improved in FWL by employing nonlinear fuzzy based mapping. The incorporation of fuzzy rules into FWL makes the HF prediction more accurate than LAL resulting in better objective and subjective performance.

LFLANN, a soft-computing technique, is exploited here for HF restoration in Lanczos-3 interpolated, up-scaled images. The improvement in PSNR gain is due to the nonlinearity introduced into the system because of nonlinear function expansion using Legendre polynomials.

The proposed LAL and FWL post-processing algorithms are computationally more complex since they are local schemes and operate on high resolution, up-scaled images. Therefore, they are not suitable for real-time applications. In contrast, LFLANN takes much reduced execution time than LAL and FWL because of its single layer architecture. In addition, it gives much better objective performance because of its improved nonlinearity due to nonlinear function expansion and hence must be preferred amongst the various post-processing algorithms.

The proposed composite scheme, CS-II exploits the advantages of pre-processing and post-processing operations for efficient restoration of HF and VHF information in the up-scaled images and hence gives better performance than the standalone schemes. The stand-alone pre-processing algorithm such as ILLGP is developed to meet the real-time requirements because of their reduced computational complexity. The composite scheme on the other hand is developed for the quality enhancement in the up-scaled image at the expense of higher computational complexity and may be employed for offline applications. It shows better performance because it makes use of fuzzy weighted Laplacian based post-processing scheme whose nonlinearity is enhanced by varying the various parameters such as slope, width and number of input and output membership functions.

From the result analysis, IUM and CS-II show much better subjective and objective performance. However, IUM is the most preferred algorithm because of its reduced computational complexity and enhanced visual quality and hence proves its suitability for various real-time and offline applications.

7.4 Scope for Future Work

There is sufficient scope to carry further research work in developing better image up-scaling algorithms like:

1. Temporal-domain up-sampling
2. Up-sampling in intensity domain
3. Fusion of spatial and intensity-domain up-sampling
4. Sparse based image super-resolution
5. Deep-learning based resolution enhancement

References

[1] C. Chen, C. Kuo, T. Yao and S. Hsieh, “Anisotropic probabilistic neural network for image interpolation,” Journal of Math Imaging Vis., vol. 48, pp. 488-498, Feb. 2013.

[2] A. Maalouf and M. C. Larabi, “Image super-resolution a state-of-the-art review and evaluation,” Book Chapter Springer, Sc. and Business Media New York, pp. 181-218, 2013.

[3] T. M. Lehman, C. Gonner and K. Spitzer, “Survey: interpolation methods in medical image processing,” IEEE Trans. on Medical Imaging,” vol. 18, no. 11, pp. 1049-1075, Nov. 1999.

[4] R. G. Keys, “Cubic convolution interpolation for digital image processing,” IEEE Trans. on Acoust. Speech Signal Process., vol. 29, no.6, pp. 1153-1160, Dec. 1981.

[5] F. Arif and M. Akbar, “Re-sampling air borne sensed data using bilinear interpolation algorithm,” in Proc. of IEEE Int. Conf. on Mechatronics, pp. 62-65, Jul. 2005.

[6] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd edition, Prentice Hall, Aug. 2007

[7] M. Marta, S. Grgic, and M. Grgic, “Picture quality measure in image compression system,” in Proc. EUROCON 03, pp. 233-237, 2003.

[8] Z. Wang, A. C. Bovic, “A universal quality index”, IEEE Signal Processing Letters, vol. 9, no. 3, pp. 81-84, 2002.

[9] W. Siu and K. Hung, “Review of image interpolation and super-resolution,” in Proc. of the Asia Pacific Signal and Information Processing Association Annual Summit and Conference, pp. 1-10, 2012.

[10] W. Ye and A. Entezari, “A geometric construction of multivariate sinc functions,” IEEE Trans. on Image Process., vol. 21, no. 6, pp. 2969-2979, June 2012.

[11] Z. Ye, J. Suri, Y. Sun and R. Janer, “Four image interpolation techniques for ultrasound breast phantom data acquired using fischer’s full field digital mammography and ultrasound systems (FFDMUS): a comparative approach,” in proc. of IEEE Int. Conf. on Image Process. ICIP 2005, pp. 1238-1241, 2005.

[12] W. Burger, M. J. Burge, “Principle of Digital Image Processing: Core Algorithms,” Springer, pp. 231-232, 2009.

[13] K. Turkowski, “Filters for common re-sampling tasks,” Graphic Gems, Academic Press, 1990.

[14] T. Lehmann, C. Gonner and K. Spitzer, “Addendum: B-spline interpolation in medical Image processing,” IEEE Trans. on Med. Imag., vol. 20, no. 7, pp. 660-665, Jul. 2001.

[15] K. H. Chung, Y. H. Fung and Y. H. Chan, “Image enhancement using fractal,” in proc. of IEEE Int. Conf. (ICASSP), pp. 273-276, 2003.

[16] M. M. Hadhoud, M. I. Dessouky and F. E. A. El-samie, “Adaptive image interpolation based on local activity levels,” in Proc. of Twentieth National Radio Science Conf. 2003, March 2003.

[17] T. Blu, P. Thevenaz, and M. Unser, "Linear interpolation revitalized," IEEE Trans. on Image Process., vol. 13, no. 5, pp. 710-719, May 2004.

[18] H. S. Malvar, L. He and R. Cutler, "High-quality linear interpolation for demosaicing of bayer-patterned color images," in proc. of IEEE Int. Conf. ICASSP 2004, pp. 485-488, 2004.

[19] Aly. H.A and Dubois. E., "Image up-sampling using total-variation regularization with a new observation model," IEEE Trans, on Image Process., vol. 14, no. l0, pp. 1647-1659, Oct 2005.

[20] D. F. Shen, C. W. Chiu and P. J. Huang, "Modified Laplacian filter and intensity correction technique for image resolution enhancement," in Proc. IEEE Int. Conf. ICME 2006, pp. 457-460, 2006.

[21] X. He, W. Jia, N. Hur, Q. Wu and J. Kim, "Image translation and rotation on hexagonal structure," in Proc. of Sixth IEEE Int. Conf. on Computing and Information Technology, 2006.

[22] X. He, W. Jia, J. Li, Q. Wu and T. Hintz, "An approach to edge detection on a virtual hexagonal structure," in Proc. of IEEE Int. Conf. on Digital Image Computing techniques and Applications DICTA 2007, pp. 340-345, 2007.

[23] H. Gharavi and S. Gao, "Spatial interpolation algorithm for error concealment," in Proc. of IEEE Int. Conf. ICASSP 2008, pp. 1153-1156, 2008.

[24] D. Bera, L. Agarwal and S. Banerjee, "Multirate scan conversion of ultrasound images using wraped distance based adaptive bilinear interpolation," in Proc. of IEEE Int. Symposium on Computer-Based Medical System 2009, pp. 1-5, 2009.

[25] P. Shen, L. Zhang, Z. Wang, W. Shao, Z. Hu, X. Zhang, Y. Song, "A novel interpolation algorithm for nonlinear omni-catadioptric images," in Proc. IEEE Int. Conf. IC-NIDC 2012, pp. 508-511, 2012.

[26] Y. Jiang, T. Pervez, M. Pattich and B. Khan, "A configurable system for role-specific video imaging during Laparoscopic surgery," in Proc. of IEEE 12th Int. Conf. on Bioinformatics and Bioengineering BIBE 2012, pp. 738-743, Nov. 2012.

[27] W. Kang, J. Jeon, E. Lee, C. Cho, J. Jung, T. Kim, A. K. Katsaggelos and J. Paik, "Real-time super-resolution for digital zooming using finite kernel based edge orientation estimation and truncated image restoration," in Proc. of IEEE Int. Conf. ICIP 2013, pp. 1311-1315, 2013.

[28] Y. Xiao, Y. Pang and L. Zhao, "Image zooming method using hierarchical structure," in Proc. of Int. Conf. on Inform. And cloud computing 2013, pp. 787-792, 2013.

[29] Q. Dai, A. K. Katsaggelos, S. Yu, W. Kang, J. Jeon and J. Paik, " Directionally adaptive cubic spline interpolation using optimized interpolation kernel and edge orientation for mobile digital zooming system," In Proc. of IEEE Int. Conf. ISCE 2014, pp. 1-2, 2014.

[30] X. Zhong, Y. He, J. Wang and L. Mei, "An adaptive image amplification method with integer multipliers," in Proc. of IEEE Int. Conf. ICALIP 2014, pp. 414-418, 2014.

[31] N. V. Hung, N. T. T. Hien, P. T. Vinh, N. T. Thao and N. T. Dzung, "An utilization of edge detection in a modified bicubic interpolation used for frame enhancement in a camera based traffic monitoring," in Proc. of IEEE Int. Conf. ICIC 2017, pp. 316-319, 2017.

[32] S. W. Lee and J. K. Paik, “Image interpolation using fast B-spline filtering,” in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process. ICASSP 1993, vol. 5, pp. 177-180, 1993.

[33] F. G. B. De Natale, G. S. Desoli and D. D. Giusto, “Adaptive least-square bilinear interpolation (ALSBI): a new approach to image-data compression,” Electronics letters, vol. 29, no. 18, pp. 1638-1640, Sep. 1993.

[34] J. Han and S. Baek, “Parametric cubic convolution scaler for enlargement and reduction of image,” IEEE Trans. on Consumer Electronics, vol. 46, no. 2, pp. 247-256, May 2000.

[35] K. Jensen and D. Anastassiou, “Sub-pixel edge localization and interpolation of still images,” IEEE Trans. on Image Processing, vol. 4, no. 3, pp. 285-295, Mar. 1995.

[36] M. Sajjad, N. Ejaz, I. Mehmood and S. W. Baik, “Digital Image Super-resolution using Adaptive Interpolation based Gaussian Function,” Multimedia Tools and Applications, Springer, vol.74, no. 20, pp. 8961-8977, July 2013.

[37] X. Li, M. T. Orchard, “New edge-directed interpolation,” IEEE Trans. on Image Process., vol. 10, no. 10, pp. 1521-1527, 2001.

[38] H. Loung, A. Ledda and W. Philips, “An image interpolation scheme for repetitive structures,” in Proc. of ICIAR (LNCS 4141), pp. 104-115, 2006.

[39] W. Tam, C. Kok and W. Siu, “A modified edge directed interpolation for images,” in Proc. of EUSIPCO 2009, pp. 283-287, Aug. 2009.

[40] G. Shi, W. Dong, X. Wu and L. Zhang, “Context-based adaptive image resolution up-conversion,” Journal of Electronic imaging, vol. 19, no. 1, pp. 1-9, March 2010.

[41] T. Dung, J. Sole, P. Yin, C. Gomila and T. Q. Nguyen, “Selective data pruning-based compression using higher order edge-directed interpolation,” IEEE Trans. on Image Process., vol. 19, no. 2, pp. 399-409, Feb. 2010.

[42] K. Misiba, T. Suzuki and M. Ikehara, “Edge adaptive image interpolation using constrained least squares,” in Proc. of IEEE Int. Conf. ICIP 2010, pp. 2837-2840, Sep. 2010.

[43] S. Yu, R. Zhang, S. Wu. J. Hu and Y. Xie, “An edge-directed interpolation method for fetal spine MR images,” Biomedical Engineering online 2013, pp. 1-12, 2013.

[44] Q. Wang and R. K. Ward, “A new orientation-adaptive interpolation method,” IEEE Trans. on Image Processing, vol. l6, no. 4, pp. 889-900, April 2007.

[45] S. Yang, Y. Kim and J. Jeong, “Fine edge-preserving technique for display devices,” IEEE Trans. on Consumer Electronics, vo1. 54, no .4, pp. 1761 - 1769, Nov. 2008.

[46] K. P. Hong, J. K. Paik, H. J. Kim and C. H. Lee, “An edge-preserving image interpolation system for a digital camcorder,” IEEE Trans. on Consumer Electronics, vo1 .42, no. 3 , pp. 279-284, Aug. 1996.

[47] M. J. Chena, C. H. Huang and W. L. Lee, “A fast edge-oriented algorithm for image interpolation,” Image and Vision Computing, Vol. 23, pp. 791 -798, 2005.

[48] M. Li and T. Q. Nguyen, “Markov random field model based edge directed image interpolation," IEEE Trans. on Image Process., vol. l7, no. 7 , pp. 1121 - 1128 , July 2008.

[49] L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," IEEE Trans. on Image Process., vo1. 15, no. 8, pp.2226-2238, Aug. 2006.

[50] A. Behnad and X. Wu, "Image interpolation with hidden Markov model," in Proc. IEEE Int. Conf. Acoustics Speech Signal Processing, pp. 874-877, March 2010.

[51] W. Tam, C. Kok and W Siu, "A modified edge directed interpolation for images," Journal of Electronic Imaging, Vol.19, no. 1, pp. 1 -20, March 2010.

[52] X. Liu, D. Zhao, R. Xiong, S. Ma, W. Gao and H. Sun, "Image interpolation via regularized local linear regression," IEEE Trans. on Image Processing, vo1.20, no. 12, pp. 3455-3469, Dec. 2011.

[53] X. Zhang and X. Wu, "Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation," IEEE Trans. on Image Processing, vol. 17, no.6, pp. 887-896, June 2008.

[54] K. Hung and W. Siu, "Robust soft-decision interpolation using weighted least squares," IEEE Trans. on Image Processing, vol. 21, no. 3, pp. 1061-1069, March 2012.

[55] K. Hung and W. Siu, "Improved image interpolation using bilateral filter for weighted least square estimation, "in Proc. IEEE Int. Conf. Image Processing (ICIP 2010), pp. 3297-3300, Sep. 2010.

[56] C. Wong and W. Siu, "Further improved edge-directed interpolation and fast EDI for SDTV to HDTV Conversion," in Proc. 18th European Signal Processing Conference (EUSIPCO 2010), pp. 309-313, Aug. 2010.

[57] C. Wong and W. Siu, "Adaptive directional window selection for edge-directed interpolation," in Proc. ICCCN 2010, pp. 1-6, Aug. 2010.

[58] K. Tang, O. C. Au, L. Fang, Z. Yu and Y. Guo, "Image interpolation using autoregressive model and gauss-seidel optimization," in Proc. Sixth Int. Conf. Image and Graphics (ICIG 2011), pp. 66-69, Aug. 2011.

[59] J. Ren, J. Liu, W. Bai and Z. Guo, "Similarity modulated block estimation for image interpolation," in Proc. IEEE Int. Conf. Image Processing (ICIP 2011), pp. 1l77-ll80, Sept. 2011.

[60] K. Ni and T. Q. Nguyen, "An adaptive k-Nearest neighbor algorithm for MMSE image interpolation," IEEE Trans. on Image Processing, Vol.18, No. 9, pp 1976- 1987, Sep. 2009.

[61] J. Cao, M. Che, X. Wu and J. Liang, "GPU-aided directional image/video interpolation for real-time resolution up-conversion, " in Proc. IEEE Int. Workshop on Multimedia Signal Processing (MMSP), pp. 1-6, Oct. 2009.

[62] W. Lei, R. Xiong, S. Ma and L. Liang, "GPU based fast algorithm for tanner graph based image interpolation," IEEE Int. Workshop on Multimedia Signal Processing (MMSP), pp. 1-4, Oct. 2011.

[63] S. Mallat and G. Yu, "Super-resolution with sparse mixing estimators", IEEE Trans. on Image Processing, vol. 19, no. 2, pp. 2889-2900, 2010.

[64] K. Hung and W. Siu, "Fast video interpolation up-sampling using linear motion model," in Proc. IEEE Int. Conf. Image Processing (ICIP 2011), pp. 1341-1344, Sep. 2011.

[65] K. Hung and W Siu, "Depth-assisted nonlocal means hole filling for novel view synthesis," in Proc. Int, Conf. Image Processing (ICIP 2012), Oct. 2012.

[66] Z. Alkachouch and M. G. Bellanger, “Fast DCT based spatial-domain interpolation of blocks in images,” IEEE Trans. on Image Processing, vol. 9, no. 4, pp. 729-732, April 2000.

[67] R. Dugad and N. Ahuja, “A fast scheme for image size change in the compressed domain,” IEEE Trans. on Circuit and System for Video Technology, vol. 11, no. 4, pp. 461-474, April 2001.

[68] J. Mukharjee and S. K. Mitra, “Image resizing in compressed domain using sub-band DCT,” IEEE Trans. on Circuit and System for Video Technology, vol. 12, no. 7, pp. 620-627, July 2002.

[69] C. Wang, H. Yu and M. Zheng, “A fast scheme for arbitrary resizing of digital image in compressed domain,” IEEE Trans. on Consumer Electronics, vol. 49, no. 2, pp. 466-471, May 2003.

[70] H. Shu and L. Chau, “An efficient arbitrary downsizing algorithm for video transcending,” IEEE Trans. on Circuit and System for Video Technology, vol. 14, no. 6, pp. 887-891, June 2004.

[71] Y. S. Park and H. W. Park, “Arbitrary-ratio image resizing using fast DCT of composite length for DCT-based transcoding,” IEEE Trans. on Image Processing, vol. 15, no.2, pp. 494-500, Feb. 2006.

[72] C. Salazar and T. D. Tran, “A complexity scalable universal DCT domain image resizing algorithm,” IEEE Trans. on Circuit and System for Video Technology, vol. 17, no. 4, pp. 495-499, April 2007.

[73] H. Shuand and L. Chau, “A resizing algorithm with two stage realization for DCT-based transcoding,” IEEE Trans. on Circuit and System for Video Technology, vol. 17, no. 2, pp. 248-253, Feb. 2007.

[74] I. Shin and H. W. Park, “Adaptive up-sampling using DCT for spatial scalability of scalable video coding,” IEEE Trans. on Circuit and System for Video Technology, vol. 19, no. 2, pp. 206-214, Feb. 2009.

[75] Z. Wu, H. Yu, and C. W. Chen, “A new hybrid DCT-Wiener-based interpolation scheme for video intra-frame up-sampling,” IEEE signal processing letters, vol. 17. No.10, pp. 827-830, oct. 2010.

[76] K. Hung and W. Siu, “Hybrid DCT-Wiener-based interpolation via learnt Wiener filter,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., pp. 1419–1423, May 2013.

[77] J. Huang, K. Hung and W. Siu, “Hybrid DCT-wiener based interpolation using dual MMSE estimator scheme, in Proc. IEEE Int. Conf. on Digital Signal Processing, pp. 748-753, Aug. 2014.

[78] H. Lim and H. Park, “A ringing-artifact reduction method for block-DCT-based image resizing,” IEEE Trans. on Circuit and System for Video Technology, vol. 21, no. 7, pp. 879-889, July 2011.

[79] S. Zhao, H. Han and S. Peng, “Wavelet-domain HMT-based image super-resolution,” in Proc. Of IEEE Int. Conf. on Image Processing ICIP 2003, vol.2, pp. 953-956, 2003.

[80] A. Temizel and T. Vlachos, “Wavelet domain image resolution enhancement using cycle-spinning,” Electronic Letters, vol. 41, no. 3, Feb. 2005.

[81] X. Wu and B. Su, “A wavelet-based resolution enhancement technique,” in Proc. IEEE Int. Conf. on Electronics and Optoelectronics (ICEOE 2011), pp. 62-65, 2011.

[82] H. Damirel and G. Anbarjafari, “Image resolution enhancement by using discrete and stationary wavelet decomposition,” IEEE Tran. on Image Process., vol.20, no. 5, pp. 1458-1460, 2011.

[83] H. He and W. C. Siu, “Single image super-resolution using Gaussian process regression,” Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR 2 011), pp.449-456, 20-25 June 2011 , Colorado, USA

[84] S. Dai, M. Han, W. Xu, Y. Wu, Y. Gong and A. Katsaggelos, “Soft Cuts: A soft edge smoothness prior for color image super resolution,” IEEE Trans. on Image Processing, vol. l 8, no. 5, pp. 969-981, May 2009.

[85] J. Sun, J. Sun, Z. Xu and H. Y. Shum, “Gradient profile prior and its applications in image super-resolution and enhancement,” IEEE Trans. on Image Processing, vo1 .20, no .6, pp. 1529-1542, June 2011.

[86] X. Gao, K. Zhang, D. Tao and X. Li, “Joint learning for single-image super-resolution via a coupled constraint,” IEEE Trans. on Image Processing, vo1. 21, no.2, pp. 469-480, Feb. 2012.

[87] J. Yang, J. Wright, T. S. Huang and Y. Ma, “Image super-resolution via sparse representation,” IEEE Trans. on Image Processing, vol. l 9, no. 11, pp .2861-2873, Nov. 2010.

[88] K. I. Kim and Y. Kwon, “Single-image super-resolution using sparse regression and natural image prior,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vo1 .32, no .6, pp. 1127- 1133 , June 2010.

[89] W. Dong, L. Zhang, G. Shi and X. Wu, “Image de-blurring and super-resolution by adaptive sparse domain selection and adaptive regularization," IEEE Trans. on Image Processing, vol. 20, no. 7, pp. 1838-1857, July 2011.

[90] D. Glasner, S. Bagon and M. Irani, “Super-resolution from a single image”, Proc. Int. Conf. Computer Vision (ICCV), pp. 349-356, 2009.

[91] W. T. Freeman, T. R. Jones, and E. C. Pasztor, “Example-based super-resolution,” IEEE Computer Graphics and Applications, voI.22(2), pp. 56-65, March/April 2002.

[92] R. He and Z. Zhang, “Locally affine patch mapping and global refinement for image super-resolution,” Pattern Recognition, Volume 44, pp. 2210-2219, Issue 9, September 2011 .

[93] K. W. Hung and W. C. Siu, “Single-image super-resolution using iterative wiener filter,” Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), pp. 1 269- 1272, 25-30, March 2012.

[94] B. K. Gunturk, A. Batur, Y. Altunbasak, M. H. Hayes and R.M. Mersereau, “Eigenface-domain super-resolution for face recognition,” IEEE Trans. on Image Processing, vol. 12, no. 5 , pp. 597-606, May 2003 .

[95] H. Huang, H. He, X. Fan and 1. Zhang, “Super-resolution of human face image using canonical correlation analysis,” Pattern Recognition, Volume 43, Issue 7, pp. 2532-2543 , July 2010 .

[96] Y. Zhuang, J. Zhang and F. Wu, “Hallucinating faces: LPH super-resolution and neighbour reconstruction for residue compensation”, Pattern Recognition, Volume 40, Issue 11 , pp. 3178-3194, November 2007.

[97] W. Zhang and W. K. Cham, “Hallucinating face in the DCT domain”, IEEE Trans. on Image Processing, vo1. 20, no. 10, pp.2769-2779, Oct. 2011.

[98] K. Guo, X. Yang, R. Zhang, G. Zhai and S. Yu, "Face super-resolution using 8-connected Markov Random Fields with embedded prior" ICPR, pp. 1-4, 2008.

[99] H. Zhuo and K. M. Lam, "Eigen transformation-based face super-resolution in the wavelet domain," Pattern Recognition Letters, vol. 33 (6), pp. 718-727, 2012.

[100] Y. Hu, K. M. Lam, G. Qiu and T. Shen, "From Local Pixel Structure to Global Image Super-Resolution: A New Face Hallucination Framework," IEEE Trans. on Image Processing, vol. 20(2), pp. 433-445, 2011.

[101] S. Farsiu, M. Elad, and P. Milanfar, "Video-to-video dynamic super-resolution for gray-scale and color sequences," EURASIP Journal on Applied Signal Processing, No. Article ID 6 1 859, pp. 1-15, 2006.

[102] C. Bishop, A. Blake, and B. Marthi, "Super-resolution enhancement of video", Proc. of the Ninth International Workshop on Artificial Intelligence and Statistics, January 2003.

[103] K. Simonyan, S. Grishin, D. Vatolin and D. Popov, "Fast video super-resolution via classification," Proc. IEEE Int. Conf. Image Processing (ICIP 2008) pp. 349-352, 12-15 Oct. 2008, San Diego, USA.

[104] K. W. Hung and W. C. Siu, "New motion compensation model via frequency classification for fast video super-resolution," Proc. IEEE Int. Conf. Image Processing (ICIP 2009), pp. 1193 - 1196, 7- 11 November, 2009, Cairo, Egypt.

[105] M. Irani and S. Peleg, "Improving resolution by image registration," CVGIP: Graph. Models and Image Proc., vol. 53, pp. 231 -9, May 1991.

[106] A. Marquina and S. J. Osher, "Image super-resolution by TV regularization and Bregman iteration," Sci. Comput., vol. 37, pp. 367-382, 2008.

[107] M. Protter and M. Elad, "Super resolution with probabilistic motion estimation," IEEE Trans. on Image Processing, vol. 18, no. 8 , pp. 1899-1904, Aug. 2009.

[108] M. Protter, M. Elad, H. Takeda, and P. Milanfar, "Generalizing the nonlocal- means to super-resolution reconstruction," IEEE Trans. On Image Processing, vol. 18, no. 1, pp. 36-51, Jan. 2009.

[109] H. Takeda, P. Mi Ianfar, M. Protter, and M. Elad, "Super-resolution without explicit sub-pixel motion estimation," IEEE Trans. on Image Processing, vo1. 18, no.9, pp. 1958 - 1975, Sept. 2009.

[110] Y. Zhuo, J. Liu, J. Ren and Z. Guo. "Nonlocal based super resolution with rotation invariance and Search window relocation", Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP 2012), pp. 853-856, March, 2012, Kyoto, Japan.

[111] R. C. Hardie, "A fast image super-resolution algorithm using an adaptive wiener filter," IEEE Trans. on Image Processing, vol. 16 , no. 12, pp.2953 -2964, Dec. 2007.

[112] B. Narayanan, R. C. Hardie, K. E. Barner and M. Shao, "A computationally efficient super-resolution algorithm for video processing using partition filters," IEEE Trans. on Circuits and Systems for Video Technology, vo1. 17, no. 5, pp. 621 -634, May 2007.

[113] R. C. Hardie, K. J. Barnard, E. E. Armstrong, "Joint MAP registration and high-resolution image estimation using a sequence of under sampled images ," IEEE Trans. on Image Processing, vol. 6, no . 12, pp. 1621-1633, Dec 1997.

[114] Y. He, K. H. Yap, L. Chen, and L. P. Chau, “A nonlinear least square technique for simultaneous image registration and super-resolution,” IEEE Trans. on Image Processing, vol. 16, no. 11, pp. 2830-2841, Nov. 2007.

[115] N. A. Woods, N. P. Galatsanos, and A. K. Katsaggelos, “Stochastic methods for joint registration, restoration, and interpolation of multiple under-sampled images,” IEEE Trans. on Image Processing, vol. 15, no. 1, pp. 201-213, Jan. 2006.

[116] G. Anbarjafari and H. Demirel, “Image super-resolution based on interpolation of wavelet-domain high frequency sub-bands and spatial-domain input image,” ETRI Journal, Wiley, vol. 32, no. 3, pp. 390-394, June 2010.

[117] X. Zhang, J. Jiang and S. Peng, “Commutability of blur and affine warping in super-resolution with application to joint estimation of triple-coupled variables,” IEEE Trans. on Image Processing, vo1. 21, no .4, pp. 1796- 1808, April 2012.

[118] S. K. Mishra, G. Panda and S. Meher, “Chebyshev functional link artificial neural networks for de-noising of images corrupted by salt and pepper noise,” International Journal of Recent Trends in Engineering, Academy Publisher, vol. 1, pp. 413-417, 2009.

[119] S. K. Nanda and D. P. Tripathy, “Application of functional link artificial neural network for prediction of machinery noise in open cast mines,” Advances in Fuzzy Systems, Hindawi Publishing Corporation, 2011.

[120] X. Lu, Y. Yuan and P. Yan, “Alternatively constrained dictionary learning for image super-resolution,” IEEE Trans. on Cybernetics, vol. 44, no. 3, pp. 366-377, 2014.

[121] W. Dong, L. Zhang, R. Lukac and G. Shi, “Sparse representation based image interpolation with nonlocal autoregressive modelling,” IEEE Trans. on Image Processing, vol.22, no. 4, pp. 1382-1394, 2013.

[122] M. Nazzal and H. Ozkaramanli, “Improved single image super-resolution using sparsity and structured dictionary learning in wavelet domain,” Proc. of IEEE Int. Conf. on Signal Processing and Communication Applications, pp. 1-4, 2013.

[123] S. Gu, W. Zuo, Q. Xie, D. Meng, X. Feng and L. Zhang, “Convolutional sparse coding for image super-resolution,” Proc. of IEEE Int. Conf. on Computer Vision (ICCV), pp. 1823-1831, 2015.

[124] J. Ahmed, G. L. Baloch, A. Bhatti and R. Klette, “Coupled directionally structured dictionaries for single image super-resolution,” Proc. of IEEE Int. Conf. on Imaging Systems and Techniques (IST), pp. 1-6, 2017.

[125] X. Han, J. Wang, B. Shi, Y. Zheng and Y. Chen, “Hyper-spectral image super-resolution using non-negative spectral representation with data-guided sparsity,” Proc. of IEEE Int. Symposium on Multimedia (ISM), pp. 500-506, 2017.

[126] X. Pei, Y. Guan, P. Cai and T. Dong, “Face hallucination via gradient constrained sparse representation,” IEEE Access, vol. 6, pp. 4577-4586, 2018.

[127] T. Guo, H. S. Mousavi and V Monga, "Deep learning based image super-resolution with coupled back-propagation," Proc. of IEEE Global Conf. on Signal and Information Processing (Global SIP), pp. 237-241, 2016.

[128] M. Sharma, S. Chaudhuri and B. Lall, "Deep learning based frameworks for image super-resolution and noise-resilient super-resolution," Proc. of IEEE Int. Joint Conf. on Neural Networks (IJCNN), pp. 744-751, 2017.

[129] A. Amanatiadis and I. Andreadis, "A survey on evaluation methods for image interpolation," Measurement Science and Technology, IOP Publishing, vol. 20, pp. 1-9, 2009.

[130] M. Vetterli, "Fast 2-D discrete cosine transform," Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), pp. 1538-1541, 1985.

[131] S. Mallat, A wavelet tour of signal processing, Academic Press, 9th October, 2008.

Dissemination

Journals

1. **A. Acharya** and S.Meher, "Efficient fuzzy composite predictive scheme for effectual 2-D up-sampling of images for multimedia applications," Journal of Visual Communication and Image Representation, **Elsevier**, vol. 44, pp. 156-186, April 2017.

2. **A. Acharya** and S.Meher, "Composite High Frequency Predictive Scheme for Efficient 2-D Up-scaling Performance," Multimedia Tools and Applications, **Springer**, vol. 76, pp. 1-37, January 2017.

3 **A. Acharya** and S.Meher, "Region adaptive unsharp masking based DCT interpolation for efficient video intra frame up-sampling." IJCA Special Issue on Electronic Design and Signal Processing ICEDSP (3), pp. 29-33, February 2013.

Book Chapters

1. **A. Acharya** and S. Meher, "Region adaptive unsharp masking based Lanczos-3 interpolation for 2-D upsampling: Crisp-rule versus fuzzy-rule based approach," Sensing Technology: Current Status and Future Trends II Smart Sensor, Measurement and Instrumentation, **Springer** International Publishing, vol. 8, pp. 47-73, 2014.

Conferences

1. **A. Acharya** and S.Meher, "An efficient, adaptive unsharp masking based interpolation for video intra frame up-sampling," in Proc. IEEE Asia Pacific Conf. on Postgraduate Research in Microelectronics and Electronics, Dec. 2012.

2. **A. Acharya** and S.Meher, "Region adaptive unsharp masking based Lanczos3 interpolation for video intra frame up-sampling," in Proc. IEEE Int. Conf. on Sensing Technology, Dec. 2012.

3. **A. Acharya** and S.Meher, "No reference fuzzy unsharp masking based DCT interpolation for better 2-D up-sampling," in Proc. IEEE Int. Conf. on Fuzzy Systems, July 2013.

4. **A. Acharya** and S.Meher, "Region based Laplacian post-processing for better 2-D up-sampling," in Proc. of Annual IEEE Int. Conf. INDICON 2013, IIT Bombay, Dec. 2013.

5. **A. Acharya** and S.Meher, " Local adaptive Laplacian for better 2-D up-sampling," in Proc. 2nd Int. Symposium on Computer, Communication, Control and Automation (3CA 2013), Singapore, Dec. 2013.

6. **A. Acharya** and S. Meher, "An Improved Image Super-resolution using Local Adaptive Unsharp Masking," in Proc. IEEE Int. Conf. on Electrical, Electronics, Signals, Communication and Optimization (EESCO), Vishakapatnam, January 2015.

7. **A. Acharya** and S. Meher, " Error based Sharpening for Efficient 2-D Up-scaling Performance ," in Proc. of annual IEEE Int. Conf. INDICON, New Delhi, Dec. 2015.